OTOMATISASI VALIDASI *SERIAL NUMBER* DAN *SCAN CODE* PADA ASET *IT* MENGGUNAKAN *YOLO* DAN *OCR* GUNA EFISIENSI PENDATAAN DAN KEAMANAN ASET *IT* DI PT XYZ INDONESIA

PROYEK AKHIR

Disusun Oleh:

Clinton Alfaro

3312301080

Disusun untuk memenuhi salah satu syarat memperoleh gelar Ahli Madya Teknik Informatika

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

POLITEKNIK NEGERI BATAM

2026

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yesus Kristus yang penuh kasih dan anugerah-Nya yang selalu menyertai setiap langkah kehidupan hingga akhirnya Penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul "OTOMATISASI VALIDASI SERIAL NUMBER DAN SCAN CODE PADA ASET IT MENGGUNAKAN YOLO DAN OCR GUNA EFISIENSI PENDATAAN DAN KEAMANAN ASET IT DI PT XYZ INDONESIA" dengan baik sebagai bagian dari pemenuhan syarat akademik di Jurusan Informatika, Program Studi Teknik Informatika, Politeknik Negeri Batam.

Laporan ini disusun sebagai salah satu syarat akademik dan menjadi bukti perjalanan penuh tantangan, pembelajaran, serta doa. Melalui Tugas Akhir ini, penulis memperoleh kesempatan untuk mengasah kemampuan, menambah wawasan, dan menerapkan ilmu yang telah dipelajari. Dengan rendah hati, Penulis menyampaikan terima kasih kepada semua pihak yang telah memberikan dukungan, bimbingan, dan doa, di antaranya:

1. Ir. Bambang Hendrawan, S.T., MSM., CIPMP., CISCP.selaku Rektor Politeknik Negeri Batam yang telah memberikan dukungan, kebijakan, serta sarana dan prasarana yang memadai sehingga penulis dapat menyelesaikan pendidikan serta penyusunan laporan Tugas Akhir dengan baik.
2. Ahmad Hamim Thohari, S.S.T., M.T. selaku Kepala Jurusan yang telah memberikan bimbingan, dukungan serta fasilitas akademik yang baik sehingga penulisan dapat menempuh pendidikan dan menyelesaikan penyusunan laporan Tugas Akhir ini dengan baik.
3. Ari Wibowo, S.T., M.T., selaku dosen pembimbing, yang telah memberikan arahan, nasihat, dan bimbingan yang sangat berharga sehingga penulis dapat menyelesaikan laporan ini dengan baik.
4. Yeni Rokhayati, S.Si., M.Sc. selaku Ketua Program Studi Teknik Informatika yang telah memberikan dukungan serta saran akademik sehingga proses perkuliahan dan penyusunan laporan ini berjalan dengan baik.

5. Wahyu Hidayat selaku IT Manager PT XYZ Indonesia, atas kesempatan, arahan strategis, serta dukungan yang diberikan sehingga kegiatan pengerjaan proyek ini berjalan dengan baik.
6. Ikhsan Kurniawan, selaku Asst. IT Manager Infrastructure, atas arahan teknis terkait infrastrukur, bantuan, fasilitas dan masukan yang konstruktif selama proses pengerjaan proyek.
7. Robie Ramady, selaku Lead IT System Operation, atas arahan dan bimbingan teknis mengenai sistem dan operasional, dukungan, serta pengalaman praktis yang sangat bermanfaat selama pengerjaan proyek.
8. PT. XYZ Indonesia yang telah memberikan kesempatan dan dukungan serta fasilitas yang memadai untuk melaksanakan pengembangan sistem.
9. Thomson Siagian, Sonti Meldawati Siambaton, dan Lusiana Ria Siagian, selaku keluarga tercinta yang menjadi sumber kekuatan dan tempat penulis untuk kembali dalam setiap keadaan. Terima kasih atas cinta yang tulus dan doa yang tak pernah putus, serta dukungan baik secara moral, materi, dan spiritual. Kehadiran keluarga adalah alasan bagi penulis untuk hidup lebih lama lagi dan terus berjuang untuk menyelesaikan setiap proses dengan tuntas.

Penulis menyadari sepenuhnya bahwa laporan ini masih jauh dari kata sempurna. Oleh karena itu, dengan segala kerendahan hati, penulis membuka diri untuk menerima kritik dan saran yang membangun demi perbaikan di masa mendatang. Akhir kata, penulis berharap laporan ini bermanfaat bagi pembaca, menambah wawasan, dan menjadi referensi di bidang teknologi informasi serta manajemen aset.

Batam, 17 April 2026

Clinton Alfaro Siagian

DAFTAR ISI

KATA PENGANTAR ii
DAFTAR ISI iv
DAFTAR TABEL vii
DAFTAR GAMBAR ix
ABSTRAK xii

BAB I PENDAHULUAN 1
1.1. Latar Belakang 1
1.2. Rumusan Masalah 2
1.3. Tujuan 3
1.4. Batasan Masalah 3
1.5. Manfaat 3

BAB II TINJAUAN PUSTAKA 5
2.1. Penelitian Terkait 5
2.2. Landasan Teori 9
2.2.1 Kajian Pustaka 9
2.2.1.1 Sistem Inventaris Aset *IT* 9
2.2.1.2 PT XYZ Indonesia 9
2.2.2 Teknologi yang digunakan 10
2.2.2.1 Artificial Intelligence (AI) 10
2.2.2.2 You Only Look Once (YOLO) 11
2.2.2.3 Optical Character Recognition (OCR) 11
2.2.3 Bahasa Pemograman 12
2.2.3.1 *Javascript* 12
2.2.3.2 *Python* 12
2.2.4 *Framework* 12
2.2.4.1 Next.js (React.js) 13
2.2.4.2 Flask (Python) 13
2.2.4.3 Application Programming Interface (API) 14
2.2.5 *Database* 15
2.2.5.1 *PostgreSQL* 15

2.2.6 *Software* yang digunakan 15
2.2.6.1 *Visual Studio Code* 15
2.2.6.2 *GitHub* 16
2.2.6.3 *Roboflow* 16
2.2.6.4 *Google Colab* 17
2.2.6.5 *Postman* 17
2.3. Metode Pengembangan Produk 18

BAB III ANALISIS DAN PERANCANGAN 22
3.1. Analisis Kebutuhan 22
3.1.1 Proses Bisnis Berjalan 22
3.1.2 Gambaran Umum Sistem 24
3.2. Perancangan 25
3.2.1 *Flowchart* sistem yang dirancang 26
3.2.2 Kebutuhan Fungsional 28
3.2.3 Kebutuhan *Non-Fungsional* 31
3.2.4 *Use Case Diagram* 32
3.2.5 *Scenario Use Case* 33
3.2.6 *Activity Diagram* 50
3.2.6.1 *Activity Diagram Register* 50
3.2.6.2 *Activity Diagram Login* 51
3.2.6.3 *Activity Diagram Dashboard Admin* dan *Superadmin* 52
3.2.6.4 *Activity Diagram Scanning Preparation* 54
3.2.6.5 *Activity Diagram Scanning* Perangkat atau Material 55
3.2.6.6 *Activity Diagram Scanning Serial Number* atau *Scan code* 56
3.2.6.7 *Activity Diagram* Validasi dan Verifikasi Data hasil *Scan* 57
3.2.6.8 *Activity Diagram* Melihat Data *Assets* 58
3.2.6.9 *Activity Diagram* Laporan Pengecekan Inventaris 59
3.2.6.10 *Activity Diagram Monitoring real-time* 60
3.2.6.11 *Activity Diagram* Verifikasi Laporan Pengecekan 61
3.2.6.12 *Activity Diagram* Manajemen Hak Akses Pengguna 62
3.2.6.13 *Activity Diagram Logout* 63
3.2.7 *ER Diagram* 64
3.2.8 Perancangan Antarmuka (*Wireframe*) 66

BAB IV IMPLEMENTASI DAN PENGUJIAN 68
4.1 Hasil Implementasi 68
4.1.1 Implementasi Basis Data 69
4.1.2 Implementasi *API Backend* 70
4.1.3 Implementasi Model *AI* 87
4.1.3.1 Pengumpulan Data 88
4.1.3.2 Implementasi Pra-pemrosesan Data 92
4.1.3.3 Implementasi Pelabelan Data *(Labeling)* 95
4.1.3.4 Implementasi Pelatihan Model *(Training)* 100
4.1.3.5 Implementasi *Deployment* Model 103
4.1.4 Proses Deteksi Objek dan Ekstraksi Data *YOLO* dan *OCR* 104
4.1.5 Implementasi *UI* 112
4.2 Pengujian *Users* Acceptance Testing (UAT) 130
4.2.1 Pengujian Perangkat Lunak *(Functional Testing)* 130
4.2.2 Pengujian Perangkat Lunak *(Non-Functional Testing)* 171

BAB V KESIMPULAN 187
5.1 Kesimpulan 187
5.2 Saran 188

DAFTAR PUSTAKA 189
DAFTAR LAMPIRAN 192

DAFTAR TABEL

Tabel 2. 1 Perbandingan Penelitian Sebelumnya 7

Tabel 3. 1 Kebutuhan Fungsional 28
Tabel 3. 2 Kebutuhan *Non-Fungsional* 31
Tabel 3. 3 *Use Case Scenario Register* 33
Tabel 3. 4 *Use Case Scenario Login* 34
Tabel 3. 5 *Use Case Scenario Dashboard* 35
Tabel 3. 6 *Use Case Scenario Scanning Preparation* 37
Tabel 3. 7 *Use Case Scenario* Pengaturan Metode Penyimpanan 38
Tabel 3. 8 *Use Case Scenario Scanning* Perangkat atau Material 39
Tabel 3. 9 *Use Case Scenario Scanning &* Ekstraksi *Serial Number/Scan Code*. 41
Tabel 3. 10 *Use Case Scenario* Validasi & Verifikasi Data Hasil *Scan* 42
Tabel 3. 11 *Use Case Scenario* Melihat Data *Assets* 43
Tabel 3. 12 *Use Case Scenario* Laporan Pengecekan Inventaris 44
Tabel 3. 13 *Use Case Scenario Monitoring Real-Time* 45
Tabel 3. 14 *Use Case Scenario* Verifikasi Laporan Pengecekan 47
Tabel 3. 15 *Use Case Scenario* Manajemen Hak Akses Pengguna 48
Tabel 3. 16 *Use Case Scenario Logout* 49

Tabel 4. 1 Implementasi *API* 70
Tabel 4. 2 *Blackbox Testing Register* 130
Tabel 4. 3 *Blackbox Testing Login* 132
Tabel 4. 4 *Blackbox Testing Dashboard Admin & Superadmin* 135
Tabel 4. 5 *Blackbox Testing Create Scanning Preparation* 137
Tabel 4. 6 *Blackbox Testing* Pengaturan Metode Penyimpanan 140
Tabel 4. 7 *Blackbox Testing Scanning* Perangkat atau Material 142
Tabel 4. 8 *Blackbox Testing Scanning Serial Number* atau *Scan Code* 146
Tabel 4. 9 *Blackbox Testing* Validasi & Verifikasi Hasil *Scan* 151
Tabel 4. 10 *Blackbox Testing* Melihat Data *Assets* 154

Tabel 4. 11 *Blackbox Testing* Laporan Pengecekan Inventaris..... 157
Tabel 4. 12 *Blackbox Testing Monitoring Real-time* 161
Tabel 4. 13 *Blackbox Testing* Verifikasi Laporan Pengecekan..... 163
Tabel 4. 14 *Blackbox Testing* Manajemen Hak Akses Pengguna..... 166
Tabel 4. 15 *Blackbox Testing Logout*..... 169
Tabel 4. 16 *Performance Testing*..... 172
Tabel 4. 17 *Security Testing*..... 183

DAFTAR GAMBAR

Gambar 2. 1 Metode *Waterfall* 18

Gambar 3. 1 *Activity Diagram* Proses Bisnis yang Sedang Berjalan..... 23
Gambar 3. 2 Gambaran Umum Sistem 24
Gambar 3. 3 *Flowchart* sistem yang dirancang 26
Gambar 3. 4 *Use Case Diagram* 32
Gambar 3. 5 *Activity Diagram Register* 50
Gambar 3. 6 *Activity Diagram Login* 51
Gambar 3. 7 *Activity Diagram Dashboard Admin* 52
Gambar 3. 8 *Activity Diagram Dashboard Superadmin* 53
Gambar 3. 9 *Activity Diagram Scanning Preparation* 54
Gambar 3. 10 *Activity Diagram Scanning* Perangkat atau Material 55
Gambar 3. 11 *Activity Diagram Scanning Serial Number* atau *Scan Code* 56
Gambar 3. 12 *Activity Diagram* Validasi dan Verifikasi Data hasil *Scan* 57
Gambar 3. 13 *Activity Diagram* Melihat Data *Assets* 58
Gambar 3. 14 *Activity Diagram* Laporan Pengecekan Inventaris 59
Gambar 3. 15 *Activity Diagram Monitoring Real-time* 60
Gambar 3. 16 *Activity Diagram* Verifikasi Laporan Pengecekan 61
Gambar 3. 17 *Activity Diagram* Manajemen Hak Akses Pengguna 62
Gambar 3. 18 *Activity Diagram Logout* 63
Gambar 3. 19 *ER Diagram* Sistem Inventaris *IT* 64
Gambar 3. 20 *Wireframe Web* 66
Gambar 3. 21 *Wireframe Mobile* 67

Gambar 4. 1 Implementasi Basis Data 69
Gambar 4. 2 Struktur folder *Dataset Inova* 88
Gambar 4. 3 Pengumpulan Data Perangkat 89
Gambar 4. 4 Pengumpulan Data *Serial Number* 90
Gambar 4. 5 Pengumpulan Data Material 91
Gambar 4. 6 Pengumpulan Data *Scan Code* 91

Gambar 4. 7 Struktur folder pengolahan *Dataset Inova* 92
Gambar 4. 8 Ekstraksi *frames* Perangkat 93
Gambar 4. 9 Ekstraksi *frames Serial Number* 93
Gambar 4. 10 Ekstraksi *frames* Material 94
Gambar 4. 11 Ekstraksi *frames Scan Code* 94
Gambar 4. 12 *Workspace labeling* Data *Inova* 95
Gambar 4. 13 *Labeling* Data Perangkat 96
Gambar 4. 14 *Labeling* data *Serial Number* 97
Gambar 4. 15 *Labeling* Data Material 98
Gambar 4. 16 *Labeling* Data *Scan Code* 99
Gambar 4. 17 *Training* Model *Dataset* Perangkat 100
Gambar 4. 18 *Training* Model *Dataset Serial Number* 101
Gambar 4. 19 *Training* Model *Dataset* Material 102
Gambar 4. 20 *Training* Model *Dataset Scan Code* 102
Gambar 4. 21 *Deployment* Model 103
Gambar 4. 22 Tampilan fitur *scanning* dengan kamera aktif 105
Gambar 4. 23 Pemrosesan deteksi objek 106
Gambar 4. 24 Hasil deteksi dan *Bounding Box* pada Perangkat 106
Gambar 4. 25 Hasil deteksi dan *Bounding Box* pada Material 107
Gambar 4. 26 Deteksi *Serial Number* 107
Gambar 4. 27 Pemrosesan Deteksi *Serial Number* 108
Gambar 4. 28 Deteksi *Scan Code* 108
Gambar 4. 29 Pemrosesan deteksi *Scan Code* 109
Gambar 4. 30 Hasil *Bounding Box* dan Ekstraksi *Serial Number* dan *Scan Code* 110
Gambar 4. 31 *Alert* duplikasi dan format data hasil ekstraksi tidak sesuai 111
Gambar 4. 32 Implementasi *Landing Page* 112
Gambar 4. 33 Implementasi *Register* 113
Gambar 4. 34 Implementasi *Login* 113
Gambar 4. 35 Implementasi *Dashboard* 114
Gambar 4. 36 Implementasi *Create Preparation* 116

Gambar 4. 37 Implementasi *Scanning Preparation List* 117
Gambar 4. 38 Implementasi Edit *Scanning Preparation* 119
Gambar 4. 39 Implementasi *Scanning Assets* 121
Gambar 4. 40 Implementasi *Validation & Verification* 123
Gambar 4. 41 Implementasi *Assets Inventory* 124
Gambar 4. 42 Implementasi *Assets Details* 124
Gambar 4. 43 Implementasi *History & Activity Logs* 125
Gambar 4. 44 Implementasi *Report & Analytics* 126
Gambar 4. 45 Implementasi *Verify Reports & Analytics* 127
Gambar 4. 46 Implementasi *Management Users* 128
Gambar 4. 47 Implementasi *Profile* 129

ABSTRAK

Pengelolaan inventaris aset berupa perangkat teknologi dan material merupakan aspek penting dalam menjaga keberlanjutan operasional perusahaan. PT XYZ Indonesia sebagai perusahaan fabrikasi dan rekayasa berskala internasional di sektor minyak dan gas memiliki ribuan aset untuk mendukung aktivitas operasional harian. Proses pengelolaan inventaris yang masih dilakukan secara manual menggunakan *spreadsheet* menimbulkan kendala berupa pencatatan yang lambat, risiko kesalahan input, kesulitan verifikasi, serta potensi manipulasi data akibat tidak adanya bukti autentik pengecekan, sehingga berdampak pada menurunnya akurasi dan efisiensi pengelolaan aset. Untuk mengatasi permasalahan tersebut, proyek akhir ini bertujuan merancang dan membangun sistem inventaris aset *IT* yang mampu melakukan validasi *serial number* dan *scan code* secara otomatis melalui teknologi *computer vision*. Sistem mengintegrasikan algoritma *YOLOv8n* untuk mendeteksi objek serta area label *serial number* atau *scan code*, dan Optical Character Recognition (OCR) untuk mengekstraksi teks serta menampilkan informasi perangkat atau material secara otomatis. Proses pemindaian tidak hanya digunakan untuk identifikasi data, tetapi juga menyimpan hasil *scanning* sebagai bukti autentik bahwa pengecekan aset telah dilakukan oleh petugas. Pengembangan sistem menggunakan metode *Waterfall* meliputi teknologi *Next.js (React.js)* sebagai antarmuka pengguna, *Flask (Python)* sebagai proses *Artificial Intelligence*, serta *PostgreSQL* sebagai basis data utama. Sistem dilengkapi dengan fitur pemindaian otomatis, penyimpanan hasil *scan* sebagai bukti pengecekan, kode identitas unik aset, *monitoring* inventaris secara *real-time* melalui *dashboard*, verifikasi digital oleh *superadmin*, serta laporan inventaris secara sistematis. Hasil dari proyek ini berupa sistem inventaris berbasis *AI* yang mampu meningkatkan efisiensi proses pendataan, meminimalkan kesalahan input, serta meningkatkan akurasi dan transparansi pengelolaan aset *IT* di PT XYZ Indonesia.

Kata kunci: *Artificial Intelligence,* Sistem Inventaris *IT, You Only Look Once, Optical Character Recognition, Validasi Serial Number, Scan code,* Validasi data *real-time*

BAB I
PENDAHULUAN

1.1. Latar Belakang

PT XYZ Indonesia, berlokasi di Batam, merupakan bagian dari XYZ *Limited*, sebuah perusahaan multinasional yang fokus pada rekayasa untuk sektor kelautan, energi, dan industri lepas pantai. Perusahaan ini menawarkan solusi lengkap yang mencakup desain, konstruksi, perawatan, perbaikan, serta konversi berbagai tipe kapal dan infrastruktur lepas pantai. Dengan lebih dari 6 dekade pengalaman di tingkat grup, PT XYZ Indonesia berkontribusi dalam pembangunan serta pemeliharaan *rig*, platform lepas pantai, *jacket*, floating production storage and offloading (FPSO), kapal khusus, dan energi baru serta terbarukan. Selain itu perusahaan ini menjadi salah satu fasilitas kunci dikawasan yang berperan dalam memenuhi permintaan industri energi dan kelautan secara regional.

Seiring dengan kompleksitas bisnis dan besarnya aset perusahaan, pengelolaan inventaris Teknologi Informasi (IT) di departemen *IT Infrastructure* menghadapi tantangan besar. Pencatatan aset tidak hanya mencakup perangkat *IT* seperti *pc*, *laptop, keyboard, mouse, monitor, power supply, Jabra speaker, webcam, ipphone*, dan perangkat lainnya, tetapi juga material-material seperti *trunking*, kabel Local Area Network (LAN), *RJ45*, *Modular Jack*, pipa, *elbow*, *flexible* dan komponen instalasi jaringan. Seluruh proses pencatatan hingga kini masih dilakukan secara manual melalui *spreadsheet* atau formulir, yang rentan menimbulkan kesalahan, keterlambatan, dan manipulasi data. Permasalahan juga sering terjadi ketika *supervisor* tidak memiliki bukti valid, sehingga pengecekan bisa dilaporkan seolah telah dilakukan oleh teknisi. Kondisi ini menunjukkan lemahnya akurasi dan akuntabilitas sistem manual sehingga metode inventaris konvensional dinilai tidak mampu menjamin transparansi serta integritas data yang dibutuhkan oleh organisasi modern (Prasetyo, 2017).

Untuk mengatasi masalah tersebut, perusahaan pernah menerapkan sistem berbasis *barcode* dengan memberikan label pada setiap barang. Namun, metode ini

memiliki kelemahan serius karena *barcode* mudah rusak bila terkena air, hilang, tergores, atau sengaja dilepas sehingga validitas data inventaris terganggu. Sistem inventaris berbasis *labeling* konvensional sendiri memiliki kerentanan terhadap manipulasi serta penyalahgunaan aset, yang dapat berujung pada kesalahan pencatatan, kehilangan aset, hingga potensi kecurangan dalam lingkungan kerja (Utami, 2025).

Oleh karena itu, dibutuhkan solusi yang lebih efisien, akurat, dan aman melalui penerapan INOVA: Otomatisasi Validasi *Serial Number* dan *Scan Code* pada Aset *IT* menggunakan *YOLO dan OCR* guna efisiensi pendataan Aset di PT XYZ Indonesia. Sistem ini dikembangkan menggunakan *React.js (Next.js)* untuk antarmuka, *Python (Flask)* untuk integrasi *YOLO* dan *OCR*, serta *PostgreSQL* sebagai basis data utama. Nantinya *YOLO* mendeteksi objek berupa perangkat atau material. Untuk perangkat, sistem melakukan *scanning* dan validasi *serial number*, sedangkan untuk material, sistem melakukan *scanning* dan verifikasi *scan code*. Kemudian *OCR* mengekstrak teks secara otomatis dan menampilkan informasi perangkat atau material dan dikirim ke *database*. Setelah dilakukan *scanning* akan diberikan kode unik permanen pada tiap perangkat atau material sehingga identitas barang tetap konsisten meski dipindai berulang. Dengan ini, manipulasi penggunaan foto *serial number* barang luar dapat dicegah. Konsep ini mirip validasi *biometrik* pada *face recognition* yang membedakan pengecekan asli dari foto tiruan (Agustian, 2025). Dengan fitur tersebut, *INOVA* diharapkan mampu mempercepat pendataan aset, meningkatkan akurasi, dan memperkuat keamanan aset di PT XYZ Indonesia.

1.2. Rumusan Masalah

Rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana merancang sistem inventaris aset *IT* yang dapat menggantikan metode manual *spreadsheet* yang rawan kesalahan dan juga manipulasi?
2. Bagaimana mengatasi kelemahan pencatatan manual?
3. Bagaimana menerapkan *AI* yang dapat mendeteksi perangkat atau material serta memvalidasi *serial number* dan *scan code* secara otomatis?

4. Bagaimana memastikan proses pengecekan perangkat atau material benar-benar dilakukan oleh *teknisi* yang bertugas?

1.3. Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

1. Merancang sistem inventaris *IT* yang menggantikan metode manual *spreadsheet* dengan lebih andal dan minim manipulasi.
2. Mengurangi risiko kesalahan pencatatan dan meningkatkan akurasi.
3. Menggunakan *AI* dengan integrasi *YOLO* dan *OCR* untuk mendeteksi, memindai, dan menyimpan data otomatis ke *database*.
4. Menyediakan verifikasi inventaris melalui pemindaian yang menghasilkan bukti autentik dan kode unik setiap perangkat atau material.

1.4. Batasan Masalah

Untuk menjaga fokus penelitian, batasan masalah penelitian ini adalah:

1. Sistem inventaris difokuskan pada dua jenis aset, yaitu perangkat yang menggunakan *serial number* dan material yang menggunakan *scan code*.
2. Teknologi yang digunakan terbatas pada integrasi *YOLO* dan *OCR*, tanpa mencakup metode identifikasi lain seperti *RFID*.
3. Verifikasi hanya melalui pemindaian yang menghasilkan bukti autentik dan kode unik, tanpa *biometrik* seperti *face recognition*.
4. Proyek hanya mencakup perancangan, pembangunan, dan pengujian sistem inventaris berbasis *web responsif* tanpa integrasi *ERP*.

1.5. Manfaat

Manfaat dari peneliti mengembangkan proyek akhir ini sebagai berikut:

1. Menerapkan teknologi *AI* berbasis *YOLO* dan *OCR* untuk mendeteksi serta memvalidasi perangkat dan material secara otomatis sehingga meningkatkan efisiensi proses inventaris aset.
2. Menyediakan sistem inventaris yang lebih andal sebagai pengganti metode manual *spreadsheet* yang rawan kesalahan serta manipulasi data.
3. Meminimalkan risiko pencatatan yang salah, meningkatkan akurasi data, dan mempercepat proses inventaris aset *IT* secara *real-time*.

4. Menyediakan mekanisme verifikasi inventaris yang lebih transparan *scan* perangkat atau material dan pemberian kode identitas unik sehingga meningkatkan keamanan dan akuntabilitas aset perusahaan.
5. Mendukung Sustainable Development Goals (SDGs), yaitu:
   - *SDG* 9 *(Industry, Innovation, and Infrastructure),* yang menekankan pentingnya pengembangan teknologi dan inovasi untuk mendukung pembangunan industri yang berkelanjutan. Sistem yang dikembangkan merupakan bentuk penerapan inovasi teknologi berbasis *AI* untuk meningkatkan efisiensi dan transformasi digital dalam pengelolaan inventaris aset *IT*.
   - *SDG* 12 *(Responsible, Consumption and Production),* yang berfokus pada pengelolaan sumber daya secara efisien dan berkelanjutan dimana sistem inventaris ini membantu perusahaan khususnya PT XYZ Indonesia dalam mengelola aset secara lebih terstruktur, akurat, dan transparan sehingga dapat mengurangi risiko kehilangan, penyalahgunaan, serta pemborosan aset.

BAB II
TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Adapun penelitian terkait dengan proyek ini yang menjadi acuan dalam pengembangan proyek sebagai berikut:

1. Penelitian berjudul Aplikasi Pengelolaan *Smart Parking* membahas perancangan sistem parkir pintar berbasis otomatisasi untuk memudahkan pengguna, meningkatkan keamanan, dan membantu pengelola memantau ketersediaan lahan (Ramadhan, Putri, & Alwidan, 2023). Sistem menggunakan *RFID* untuk akses masuk/keluar, ultrasonik untuk deteksi kendaraan, *YOLOv4* dan *Tesseract OCR* untuk pengolahan citra plat nomor. Kemudian menggunakan *Arduino Uno, NodeMCU ESP8266,* motor *DC*, *relay*, dan *RFID reader*, dan perangkat lunak dibangun dengan *Visual Studio Code, XAMPP, database SQL,* dan *Python*. Alur kerja dimulai dari pengguna melakukan tap kartu *RFID*, sensor mendeteksi kendaraan, kamera mengenali plat nomor, dan data tersimpan di sistem, dan informasi ketersediaan *slot* parkir ditampilkan secara *real-time*. Sistem menyediakan fitur pendaftaran pengguna, info parkir, *history*, serta rekap data *admin*. Hasil pengujian menunjukkan sistem sudah berjalan baik, namun akurasi deteksi sangat dipengaruhi oleh kualitas pencahayaan dan kondisi kamera, sehingga performa menurun pada malam hari atau cahaya rendah.
2. Penelitian berjudul Deteksi Plat Nomor Kendaraan Menggunakan Algoritma *YOLO* membahas pengembangan sistem manajemen parkir modern berbasis *computer vision* untuk menggantikan metode manual pencatatan data kendaraan (Nugraha, 2023). Sistem ini mengintegrasikan *YOLO* untuk mendeteksi plat nomor dan *EasyOCR* untuk pengenalan karakter. Implementasi dilakukan menggunakan *Python* dengan antarmuka grafis *Tkinter* dan *database SQLite*. Cara kerja sistem yaitu pengguna mengunggah gambar kendaraan, lalu *YOLO* mendeteksi area plat nomor,

EasyOCR mengenali karakter, dan ditampilkan sebagai detail plat, lokasi, dan waktu, sekaligus disimpan ke *database*. Sistem menyediakan opsi *input* manual jika ada karakter yang kurang terbaca, sehingga meningkatkan akurasi. Hasil pengujian menunjukkan tingkat akurasi 88,8% pada gambar resolusi *HD* dan 86,3% pada resolusi *sub-HD*, dengan semua fungsi (*upload* gambar, deteksi, penyimpanan, dan *database*) berjalan baik berdasarkan uji *blackbox*. Kelemahannya akurasi sistem masih menurun pada gambar resolusi rendah dan harus intervensi manual untuk hasil yang optimal.

3. Penelitian berjudul Sistem Identifikasi Informasi *Expired Date* Produk Kemasan Menggunakan Kolaborasi Metode *YOLOv11m* dan *PaddleOCR*. Penelitian ini mengombinasikan *YOLOv11m* untuk mendeteksi tanggal kedaluwarsa pada kemasan dan *PaddleOCR* untuk membaca teks secara otomatis (Richo, 2025). Proses dimulai dari pengumpulan *dataset* 350 citra kemasan, anotasi label, *preprocessing*, pelatihan model dengan parameter optimal. Hasil pengujian menunjukkan akurasi tinggi, dengan *precision* 98,56%, mAP50-95 sebesar 93,08%, dan rata-rata sistem 97,42%. Inovasi utamanya adalah penerapan dua teknologi *AI* secara bersamaan yang mempercepat proses verifikasi produk dan meminimalkan *human error* pada pengecekan manual. Namun, kelemahan penelitian ini adalah sistem masih terbatas pada dataset kemasan tertentu, sehingga kurang fleksibel jika digunakan pada variasi produk dengan desain berbeda.
4. Skripsi yang berjudul Rancang Bangun Sistem *Monitoring* Area Parkir dan Pelacakan Posisi Kendaraan Berbasis *Mikrokontroler* dan *Computer Vision* membangun sistem parkir cerdas dengan menggunakan *OCR* untuk mengenali plat nomor kendaraan yang kemudian disimpan dalam *database* (Alfisyahri, 2024). Sistem ini juga dilengkapi *mikrokontroler* untuk melacak posisi kendaraan sehingga memudahkan pengelolaan area parkir. Penelitian ini memberikan solusi praktis dalam otomatisasi parkir serta meningkatkan keamanan kendaraan di area tertentu. Namun, kelemahannya adalah sistem ini hanya berfokus pada area parkir dan kendaraan, sementara

database yang digunakan masih bersifat lokal sehingga skalabilitas sistem terbatas.

Tabel 2. 1 Perbandingan Penelitian Sebelumnya

Peneliti/Tahun	**Judul/Topik Penelitian**	**Metode/Teknologi**	**Kelebihan**	**Kekurangan / Celah**
(Ramadhan, Putri, & Alwidan, 2023)	Aplikasi Pengelolaan *Smart Parking.*	*RFID*, sensor *ultrasonik*, *YOLOv4*, *Tesseract OCR*, *Arduino*, *ESP8266*, *Python*, *SQL*, *XAMPP*.	Sistem parkir *real-time*, mendukung *monitoring slot*, ada fitur pendaftaran, *history*, dan rekap data *admin*.	Akurasi deteksi sangat dipengaruhi oleh pencahayaan dan kondisi kamera, sehingga menurun pada malam hari atau area dengan cahaya rendah.
(Nugraha, 2023)	Deteksi Plat Nomor Kendaraan Menggunakan Algoritma *YOLO.*	*YOLO (real-time detection)*, *EasyOCR*, *Python*, *Tkinter GUI*, *SQLite.*	Deteksi plat otomatis, akurasi cukup tinggi (≥88%), ada *input* manual untuk koreksi hasil *OCR.*	Akurasi sistem masih menurun pada gambar resolusi rendah dan membutuhkan intervensi manual untuk hasil optimal.
(Richo, 2025)	Sistem Identifikasi *Expired Date* Produk Kemasan dengan *YOLOv11m* dan *PaddleOCR.*	*YOLOv11m* (deteksi posisi), *PaddleOCR* (ekstraksi teks), *dataset* 350 citra, *preprocessing*, *Python.*	Akurasi sangat tinggi (*precision* 98,56%, mAP50-95 93,08%), mengurangi *human error*, mempercepat verifikasi produk.	Sistem terbatas pada *dataset* kemasan tertentu, sehingga kurang fleksibel jika diterapkan pada variasi produk dengan

				desain berbeda.
(Alfisyahri, 2024)	Rancang Bangun Sistem *Monitoring* Area Parkir dan Pelacakan Posisi Kendaraan Berbasis *Mikrokontroler*.	*Mikrokontroler*, *OCR*, *database* lokal, sistem *monitoring* posisi kendaraan.	Otomatisasi parkir, membantu pelacakan posisi kendaraan, meningkatkan keamanan di area parkir.	Cakupan hanya pada area parkir tertentu dan *database* masih bersifat lokal sehingga skalabilitas terbatas.
Penelitian ini	OTOMATISASI VALIDASI *SERIAL NUMBER* DAN *SCAN CODE* PADA ASET *IT* MENGGUNAKAN *YOLO* DAN *OCR* GUNA EFISIENSI PENDATAAN DAN KEAMANAN ASET *IT* DI PT XYZ INDONESIA.	*YOLO* (deteksi perangkat/ material serta *serial number* atau *scan code*), *OCR* (ekstraksi teks), *React.js (Next.js)*, *Python (Flask)*, *PostgreSQL*.	Validasi perangkat atau material melalui pemindaian *serial number* atau *scan code* secara otomatis, disertai pemberian kode unik permanen untuk meningkatkan akurasi serta keamanan data inventaris.	Lingkup hanya pada aset *IT* dan belum mencakup integrasi dengan sistem *ERP* atau metode identifikasi lain seperti *RFID* lanjutan.

2.2. Landasan Teori

2.2.1 Kajian Pustaka

2.2.1.1 Sistem Inventaris Aset *IT*

Inventaris merupakan proses pencatatan, pengelolaan, dan pengawasan terhadap aset yang dimiliki organisasi untuk mendukung kegiatan operasional. Manajemen inventaris bertujuan menjaga ketersediaan barang dengan jumlah tepat, mengurangi risiko kehilangan, serta meningkatkan efisiensi operasional. Dalam lingkup *IT*, inventaris aset mencakup perangkat keras seperti *pc, laptop, keyboard, mouse, monitor, ipphone, webcam, Jabra speaker,* serta perangkat-perangkat yang memiliki identitas berupa *serial number*, dan juga material seperti *Trunking,* LAN, *RJ45, Modular Jack,* dan material-material lain yang diidentifikasi menggunakan *scan code* untuk memudahkan proses pengelolaan. Penelitian oleh (Nursikuwagus, 2016) menunjukkan bahwa pengelolaan inventaris aset *IT* manual sering menimbulkan masalah akurasi data, keterlambatan pencatatan, dan lemahnya fungsi kontrol pemeliharaan. Oleh karena itu, sistem inventaris berbasis digital diperlukan untuk meningkatkan akurasi dan transparansi dalam manajemen aset.

Inventaris aset *IT* tidak hanya berfungsi sebagai daftar barang, tetapi juga mendukung pemeliharaan, keamanan, dan optimalisasi penggunaan aset. Penerapan sistem informasi inventaris berbasis teknologi digital, seperti pemindaian *serial number,* atau *scan code, QR code,* memungkinkan pencarian data dilakukan lebih cepat serta meminimalkan terjadinya kesalahan. Studi lain oleh (Rakhma, 2022) pada PT Pan Brothers juga menegaskan bahwa sistem *monitoring* inventaris berbasis *web* mempermudah karyawan dalam mengakses data aset dan mengurangi risiko manipulasi pencatatan. Dengan demikian, dalam PT XYZ Indonesia, inventaris aset *IT* berbasis sistem informasi digital menjadi kebutuhan strategis untuk efisiensi, keandalan data, dan keamanan aset perusahaan.

2.2.1.2 PT XYZ Indonesia

PT XYZ Indonesia merupakan anak perusahaan dari XYZ Pte Ltd yang berlokasi di Batam, dan menjadi salah satu pusat produksi utama untuk proyek minyak dan gas skala internasional. Perusahaan ini bergerak di bidang fabrikasi,

rekayasa, dan konstruksi dengan fokus pada pembangunan anjungan lepas pantai, modul, *jacket*, serta *Floating Production Storage and Offloading* (FPSO) yang banyak digunakan dalam industri energi (Putri et al., 2023). Dalam mendukung aktivitas bisnis yang kompleks, PT XYZ Indonesia memiliki Departemen *IT Infrastructure* yang berperan penting dalam menjaga keberlangsungan operasional digital perusahaan. Departemen ini terbagi menjadi dua bagian, yaitu *System Operation* yang merupakan penggabungan fungsi *System*, *Operation* & *End Users Service*, dan Facilities dimana meliputi pengelolaan seluruh *server* dan sistem perusahaan, pelaksanaan pekerjaan lapangan terkait fasilitas serta *infrastruktur IT* seperti instalasi perangkat keamanan, *CCTV*, dan jaringan, serta layanan *End Users Service* yang mencakup penanganan kebutuhan pengguna, *remote troubleshooting*, pengelolaan inventaris perangkat, hingga implementasi Device as a Service (DaaS), serta *Digitalization & Innovation* yang berfokus pada pengembangan aplikasi dan sistem berbasis teknologi modern, termasuk solusi *AI* dengan metode *Computer Vision* untuk mendukung kebutuhan *IT Infrastructure*, mendorong otomatisasi proses, dan menghadirkan inovasi digital guna meningkatkan efisiensi serta efektivitas operasional perusahaan. Saat ini, proses inventaris di departemen tersebut masih dilakukan secara manual melalui *spreadsheet* maupun *barcode*, yang rawan mengalami kesalahan, kerusakan *barcode*, keterlambatan, hingga risiko manipulasi data (Putri et al., 2023). Oleh karena itu, pengembangan sistem otomatisasi validasi *serial number* dan *scan code* pada aset *IT* menggunakan *Yolo* dan *OCR* diharapkan mampu meningkatkan efisiensi, akurasi, serta keamanan manajemen aset *IT* di PT XYZ Indonesia.

2.2.2 Teknologi yang digunakan

2.2.2.1 Artificial Intelligence (AI)

Artificial Intelligence adalah bidang ilmu komputer yang berfokus pada pengembangan sistem yang mampu meniru kecerdasan manusia, seperti pengenalan pola, pengambilan keputusan, dan pemrosesan bahasa. Menurut (Zein, 2021) *AI* memungkinkan komputer melakukan tugas yang sebelumnya membutuhkan kecerdasan manusia. Dalam sistem inventaris, *AI* akan otomatisasi proses validasi *serial number* pada perangkat dan *scan code* pada material dengan

metode *computer vision* dan *optical character recognition (OCR)*, sehingga mampu mengurangi *human error* serta mempercepat proses verifikasi data.

2.2.2.2 You Only Look Once (YOLO)

YOLO adalah algoritma *object detection* berbasis *deep learning* yang dikembangkan oleh Joseph Redmon dan tim Pada tahun 2016. *YOLO* mampu mendeteksi objek hanya dalam satu kali pemrosesan citra dengan lebih cepat dan efisien. Algoritma ini bekerja dengan membagi gambar ke dalam *grid*, lalu memprediksi *bounding box* dan probabilitas kelas pada setiap *grid*. Pada penelitian menggunakan *YOLOv8n*, yang dirilis *Ultralytics* Januari 2023. *YOLOv8n* memiliki ukuran *input* 640 *piksel*, parameter 3,2 juta, serta kinerja deteksi dengan *mAP50-95* sebesar 37,3 pada *dataset COCO* dengan kecepatan inferensi hingga *0,99 ms di GPU A100 TensorRT* (Agustian, 2025). Dalam sistem *INOVA*, *YOLOv8n* digunakan untuk mendeteksi objek apakah berupa perangkat atau material, kemudian melakukan *bounding box* pada area *serial number* atau *scan code* secara *real-time* dengan cepat dan akurat, dan kemudian diproses oleh *OCR* untuk ekstraksi teks secara otomatis.

2.2.2.3 Optical Character Recognition (OCR)

OCR adalah teknologi untuk mengubah teks dari citra atau dokumen gambar menjadi teks digital yang dapat diolah komputer. Menurut (Banu, 2023), proses *OCR* bekerja melalui tahap segmentasi citra, ekstraksi fitur karakter, klasifikasi, dan rekonstruksi huruf atau angka sehingga teks hasil pemindaian dapat dibaca mesin. Salah satu *engine OCR* yang digunakan adalah *Tesseract OCR*, yang dikembangkan oleh Hewlett-Packard. *Tesseract* mendukung lebih dari 100 bahasa, termasuk bahasa Indonesia, dan mampu melakukan deteksi layout halaman secara otomatis Selain itu, terdapat juga *EasyOCR*, yaitu *framework OCR* yang mendukung lebih dari 80 bahasa, memiliki akurasi yang baik, serta mudah diintegrasikan dalam berbagai aplikasi berbasis *Python*.

Dalam sistem inventaris *INOVA*, *OCR* menggunakan *easyocr* untuk mengekstrak teks *serial number* atau *scan code* dari area yang telah terdeteksi oleh *YOLO*. Kemudian divalidasi dan mengirimkan data ke *database* perusahaan untuk memastikan keaslian perangkat maupun material yang dicek. Dengan *OCR*, sistem

dapat mengenali teks secara otomatis meskipun terdapat variasi font, pencahayaan, atau rotasi objek, sehingga meminimalkan risiko *human error* dalam pengecekan.

2.2.3 Bahasa Pemograman

2.2.3.1 *Javascript*

JavaScript adalah bahasa pemrograman tingkat tinggi yang dikembangkan oleh *Netscape* pada tahun 1995 dan menjadi bahasa utama dalam pengembangan aplikasi web interaktif. *JavaScript* memungkinkan manipulasi elemen antarmuka secara dinamis, komunikasi *asynchronous* melalui Application Programming Interface (API), serta pengembangan aplikasi *full-stack* seperti *Node.js* dan berbagai *framework* modern. Berdasarkan data (Mozilla Developer Network, 2024) *JavaScript* mendukung *event-driven programming*, modularitas, serta kompatibilitas lintas *platform* sehingga banyak digunakan dalam pengembangan sistem berbasis *web* responsif dan *real-time*. Dalam proyek ini, *JavaScript* digunakan untuk membangun antarmuka sistem inventaris yang interaktif, responsif di berbagai perangkat, serta data validasi aset secara *real-time.*

2.2.3.2 *Python*

Python merupakan bahasa pemrograman tingkat tinggi yang dikembangkan oleh (Rossum, 1991) dengan fokus pada keterbacaan kode, fleksibilitas, serta produktivitas pengembangan perangkat lunak. Menurut *Python Software Foundation*, *Python* mendukung berbagai paradigma pemrograman dan memiliki *library* yang luas, khususnya pada bidang *AI*, *Machine learning*, serta *Computer vision* seperti *TensorFlow*, *PyTorch*, dan *OpenCV*. Keunggulan tersebut menjadikan *Python* banyak digunakan sebagai *backend* pemrosesan data dan pengembangan model *AI* yang membutuhkan integrasi dengan sistem *web* melalui *REST API*. Pada proyek ini, Python digunakan untuk deteksi objek, ekstraksi teks serta komunikasi antara model *AI* dan antarmuka *web* inventaris.

2.2.4 *Framework*

Framework merupakan kerangka kerja perangkat lunak yang menyediakan struktur, aturan, serta kumpulan fungsi atau *library* yang dirancang untuk mempermudah proses pengembangan aplikasi berbasis *web* maupun *desktop*.

Framework mengintegrasikan berbagai komponen pemograman dalam satu sistem terorganisir sehingga pengembang tidak perlu membangun fitur dasar dari awal (Ii et al., 2015). Menurut (Pressman et al., 2019), penggunaan *framework* membantu meningkatkan efisiensi pengembangan, konsistensi kode, serta mempercepat implementasi karena telah menyediakan arsitektur dan pola desain yang terstandarisasi. Dengan *framework*, proses pembangunan aplikasi menjadi lebih terstruktur, hemat waktu, dan minim kesalahan dalam perancangan sistem.

2.2.4.1 *Next.js (React.js)*

Next.js adalah *framework open source* berbasis *React* yang dikembangkan oleh *Vercel* sejak 2016. *Framework* ini mendukung server-side rendering (SSR) dan static site generation (SSG) sehingga aplikasi *web* lebih cepat, interaktif, dan ramah mesin pencari (SEO) (Hanafi, 2024). Fitur bawaan seperti *automatic routing*, integrasi Application Programming Interface (API), *hot reload*, dan komponen antarmuka modular untuk aplikasi berskala besar. Dalam sistem *INOVA*, *Next.js* akan menjadi tampilan utama yang menghubungkan pengguna dengan sistem inventaris aset *IT*. Seluruh menu seperti *login*, pemindaian perangkat atau material dan *scanning serial number* melalui kamera, validasi hasil *scan*, penampilan data perangkat dan kode unik, pemantauan status inventaris *real-time*, hingga unduhan laporan harian/mingguan akan dibangun menggunakan *Next.js* agar *responsif* digunakan baik di perangkat *mobile* maupun *desktop*. Selain itu, *dashboard analytics* menyajikan data validasi dan statistik inventaris secara *real-time* dalam grafik atau tabel untuk keputusan yang lebih akurat.

2.2.4.2 *Flask (Python)*

Flask adalah *micro web framework* berbasis *Python* yang dikembangkan oleh Armin Ronacher pada tahun 2010 yang dirancang dengan filosofi sederhana, ringan, namun fleksibel. *Framework* ini menyediakan fitur dasar seperti *routing*, template *rendering*, dan request *handling* tanpa sehingga memungkinkan pengembang menambahkan *library* eksternal sesuai kebutuhan. Keunggulan utama *flask* yaitu memudahkan integrasi dengan berbagai *library machine learning* maupun *computer vision* berbasis *Python*, termasuk *TensorFlow*, *PyTorch*, dan *OpenCV*. Penggunaan *flask* dalam pengembangan aplikasi berbasis *AI* sangat

efektif karena mendukung implementasi *REST API* untuk komunikasi data r*eal-time*. Hal ini juga dibuktikan oleh (Agustian, 2025) dimana *flask* sebagai penghubung antara model *deep learning* dan antarmuka *web* dalam sistem deteksi wajah. Dalam proyek ini, *flask* akan menjadi penghubung antara proses deteksi perangkat/material serta *serial number/scan code (YOLO)* dan ekstraksi teks *(OCR)* dengan antarmuka *Next.js*, sekaligus mengelola komunikasi dengan basis data *PostgreSQL* untuk penyimpanan hasil validasi inventaris.

2.2.4.3 Application Programming Interface (API)

Application Programming Interface (API) adalah sekumpulan aturan, protokol, dan antarmuka yang memungkinkan suatu aplikasi atau sistem perangkat lunak berkomunikasi serta bertukar data dengan aplikasi lainnya. *API* telah menjadi bagian penting dalam ekosistem perangkat lunak modern karena mendukung integrasi antar sistem, penggunaan kembali fungsi yang sudah tersedia, serta mempercepat proses pengembangan perangkat lunak. Menurut (Ofoeda et al., 2019), *API* berperan sebagai media yang memungkinkan perangkat lunak saling terhubung dan bertukar informasi, sehingga mampu menciptakan nilai tambah dalam proses bisnis maupun pengembangan layanan digital. Selain itu, *API* juga membantu pengembang dalam memanfaatkan fungsi yang telah diimplementasikan sebelumnya sehingga dapat mengurangi waktu dan usaha dalam membangun aplikasi. Dalam pengembangan sistem informasi, *API* banyak digunakan untuk menghubungkan aplikasi dengan *database*, layanan pihak ketiga, maupun modul internal sistem agar pertukaran data dapat berlangsung secara otomatis, cepat, dan konsisten. Pada proyek *INOVA*, *API* berfungsi sebagai penghubung antara antarmuka sistem, proses validasi data hasil pemindaian, model *YOLO* dan *OCR*, serta basis data *PostgreSQL* sehingga seluruh data inventaris, *serial number*, *scan code*, dan bukti foto pengecekan dapat diproses dan disinkronkan secara terintegrasi.

2.2.5 *Database*

2.2.5.1 *PostgreSQL*

PostgreSQL adalah sistem manajemen basis data relasional (*Relational Database Management System* / RDBMS) *open-source* yang dikembangkan pada tahun 1986 di Universitas California, *Berkeley*, sebagai kelanjutan dari proyek *POSTGRES* yang dipimpin oleh Michael Stonebraker tahun 2010. *PostgreSQL* mendukung *ACID compliance* (Atomicity, Consistency, Isolation, Durability) yang memastikan integritas data tetap terjaga meskipun terjadi kegagalan sistem. *PostgreSQL* memiliki fleksibilitas tinggi dengan dukungan *structured query language* (SQL) standar, *object-relational features*, serta kemampuan menangani data dalam jumlah besar secara efisien. Menurut (Hafsah, 2025), *PostgreSQL* banyak digunakan pada sistem informasi modern karena kestabilannya dalam menangani transaksi kompleks dan skalabilitas yang baik pada lingkungan perusahaan. Dalam proyek *INOVA*, *PostgreSQL* berfungsi sebagai basis data utama untuk menyimpan data inventaris, hasil validasi *serial number* dan *scan code* dari *YOLO* dan *OCR*, serta bukti foto pengecekan. Setiap perangkat atau material diberi *unique identifier* untuk menjaga konsistensi data meski dipindai berulang, dan sistem mendukung *export* laporan otomatis dalam format *excel*.

2.2.6 *Software* yang digunakan

2.2.6.1 *Visual Studio Code*

Visual Studio Code (VS Code) adalah *source-code editor* yang dikembangkan oleh *Microsoft* dengan dukungan untuk berbagai bahasa pemrograman, integrasi ekstensi, serta sistem kontrol versi. *VS Code* banyak digunakan karena performanya ringan, mendukung *debugging*, *syntax highlighting*, *intellisense*, serta integrasi dengan *Git* untuk proses kolaborasi pengembangan perangkat lunak. Dalam sistem *INOVA*, *VS Code* digunakan sebagai lingkungan pengembangan utama *(Integrated Development Environment)* untuk membangun tampilan sistem dengan *Next.js*, dan integasi *YOLO* dan *OCR* menggunakan *Flask* berbasis *Python*, serta integrasi ke *database PostgreSQL*. Dengan ekstensi seperti

Python Tools, ES7 React Snippets, dan *Prettier* membantu mempercepat *coding, debugging,* dan konsistensi kode.

2.2.6.2 *GitHub*

GitHub merupakan *platform* berbasis *cloud* untuk manajemen kode sumber yang menggunakan sistem kontrol versi *Git*. Platform ini memfasilitasi kolaborasi tim dalam pengembangan perangkat lunak melalui penyimpanan repositori secara terpusat, pelacakan perubahan kode, serta pengelolaan versi secara sistematis. *GitHub* juga menyediakan berbagai fitur pendukung seperti *issue tracking, pull request,* serta integrasi dengan layanan *continuous integration* yang mendukung siklus pengembangan perangkat lunak modern. Selain itu, *GitHub* dikenal sebagai *platform social coding* yang memungkinkan interaksi antar pengembang serta berbagi pengetahuan melalui repositori *open source*.

Menurut (Cosentino et al., 2016), *GitHub* menjadi salah satu platform paling populer dalam pengembangan perangkat lunak karena menyediakan fitur kolaboratif dan *metadata* proyek yang dapat dimanfaatkan untuk analisis, pengembangan, serta peningkatan kualitas perangkat lunak. Hal ini menjadikan *GitHub* sebagai media yang efektif dalam mendukung praktik pengembangan perangkat lunak modern berbasis kolaborasi. Dalam sistem *INOVA*, *GitHub* digunakan sebagai repositori utama untuk menyimpan kode *frontend (Next.js), backend (Flask, YOLO, dan OCR),* serta skema *database PostgreSQL*. Selain itu, *GitHub* berperan penting dalam memfasilitasi kolaborasi antara teknisi dan tim *IT* PT XYZ Indonesia, tidak hanya melalui dokumentasi proyek dan manajemen versi, tetapi juga menyediakan integrasi yang mudah ke dalam *pipeline deployment* sehingga proses pengembangan berjalan secara terstruktur dan transparan.

2.2.6.3 *Roboflow*

Roboflow merupakan platform berbasis *cloud* yang digunakan untuk mengelola *dataset* pada bidang *computer vision* secara terstruktur, meliputi proses pengumpulan data, anotasi, augmentasi, hingga ekspor *dataset* ke berbagai format yang kompatibel dengan model *deep learning*. Penggunaan platform seperti *Roboflow* dapat membantu meningkatkan efisiensi dalam tahap persiapan data melalui fitur pelabelan *(labeling), preprocessing,* serta manajemen *dataset* yang

sistematis sehingga proses pengembangan model menjadi lebih konsisten dan mudah direproduksi. Secara umum, pendekatan berbasis visual dalam pengelolaan data dan pemrograman memungkinkan pengguna untuk berinteraksi dengan data secara lebih intuitif serta meminimalkan kesalahan dalam proses pengolahan (Alexandrova et al., 2016). Dalam sistem *INOVA*, *Roboflow* dimanfaatkan sebagai media pelabelan *(labeling)* data inventaris, khususnya untuk menandai objek aset pada citra sehingga *dataset* yang dihasilkan memiliki kualitas anotasi yang baik dan siap digunakan dalam proses pelatihan model deteksi objek.

2.2.6.4 *Google Colab*

Google Colab merupakan layanan *notebook* berbasis *cloud* yang dikembangkan oleh *Google* untuk menjalankan kode *Python* secara langsung melalui *browser* tanpa memerlukan instalasi lokal. *Google Colab* menyediakan dukungan sumber daya komputasi seperti Graphics Processing Unit (GPU) dan Tensor Processing Unit (TPU) yang sangat membantu dalam proses pengembangan serta pelatihan model *machine learning* dan *deep learning*. Selain itu, *Google Colab* memungkinkan pengguna untuk mengakses berbagai *runtime environment* serta menjalankan komputasi dengan kecepatan tinggi tanpa bergantung pada perangkat keras lokal. Integrasi dengan *Google Drive* juga memberikan kemudahan dalam penyimpanan *dataset*, pengelolaan file, serta kolaborasi eksperimen secara fleksibel. Menurut (Kanani et al., 2019), *Google Colab* dapat digunakan sebagai solusi alternatif ketika keterbatasan perangkat keras seperti *CPU* dan *GPU* menjadi kendala dalam proses komputasi *deep learning*. Dalam sistem *INOVA*, *Google Colab* digunakan untuk melakukan *training* model deteksi objek dengan memanfaatkan *GPU NVIDIA T4* sehingga proses pelatihan model menjadi lebih cepat dan mampu menangani dataset inventaris yang memiliki kompleksitas visual cukup tinggi.

2.2.6.5 *Postman*

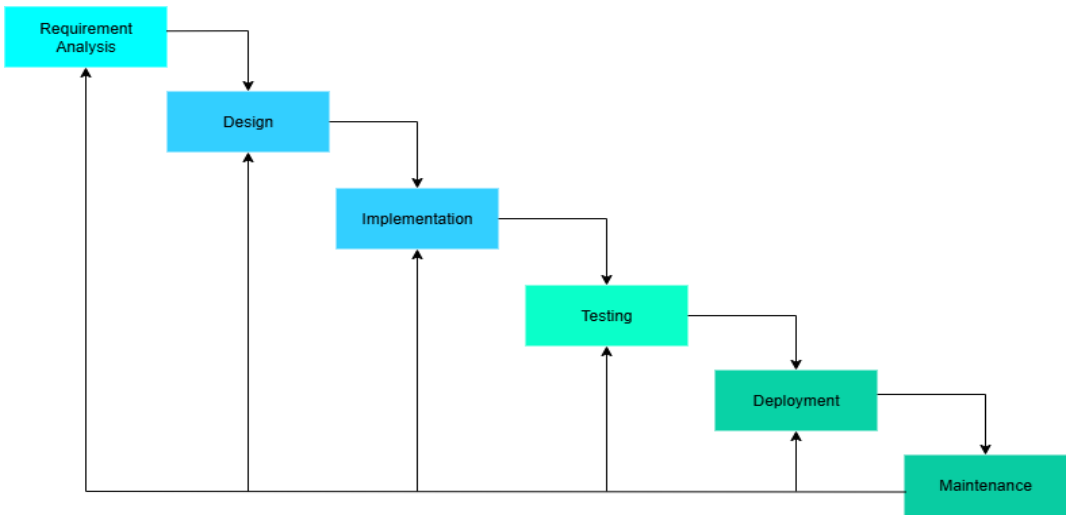
Postman merupakan sebuah *platform* yang digunakan untuk pengujian, pengelolaan, dan integrasi *API* yang mendukung proses komunikasi antar sistem secara efisien. Dalam pengembangan sistem berbasis data dan *machine learning*, Postman berperan penting dalam melakukan pengujian *endpoint API*, validasi

respons data, serta memastikan integrasi antara model dengan aplikasi berjalan dengan baik. Selain itu, *Postman* menyediakan antarmuka yang intuitif sehingga memudahkan pengguna dalam mengirim *request*, mengelola parameter, serta menganalisis hasil respons secara *real-time*.

Menurut (Jin et al., 2025), sistem berbasis *API* modern dirancang untuk mendukung integrasi yang fleksibel, skalabilitas tinggi, serta efisiensi dalam pengolahan dan pertukaran data pada berbagai level aplikasi. Hal ini menunjukkan bahwa penggunaan *tools* seperti *Postman* sangat relevan dalam mendukung pengembangan sistem yang terintegrasi dan berbasis layanan *(service-oriented).* Dalam sistem *INOVA*, *Postman* digunakan untuk melakukan pengujian *API* pada proses integrasi model deteksi objek dengan aplikasi, sehingga memastikan bahwa data hasil prediksi dapat dikirim dan diterima dengan akurat serta mendukung proses *deployment* sistem secara optimal.

2.3. Metode Pengembangan Produk

Metode pengembangan perangkat lunak yang digunakan dalam proyek ini adalah *Waterfall* Model sebagaimana yang ditunjukkan pada Gambar 2.1, yang merupakan salah satu model dalam Software Development Life Cycle (SDLC). Model ini diperkenalkan oleh Royce pada tahun 1970 ini cocok untuk sistem dengan kebutuhan jelas sejak awal. Menurut (Bassil, 2012), *Waterfall* memiliki tahapan berurutan mulai dari *requirement analysis, design, implementation, testing, deployment,* hingga *maintenance.*



Gambar 2. 1 Metode *Waterfall*

Berikut penjelasan tiap tahap pengembangan sistem proyek:

1. *Requirement Analysis*

   Pada tahap ini dilakukan pengumpulan data mengenai permasalahan dan kebutuhan sistem. (Bassil, 2012), menjelaskan bahwa *requirement analysis* kebutuhan bertujuan menghasilkan spesifikasi kebutuhan *fungsional* dan *non-fungsional.* Pada proyek *INOVA*, analisis dilakukan dengan wawancara bersama *Asst Manager* dan Teknisi *IT Infrastructue* PT XYZ Indonesia untuk mengetahui masalah inventaris manual dan *barcode* seperti kesalahan pencatatan atau *human error,* keterlambatan pelaporan, kurangnya bukti otentik pengecekan, kesulitan pelacakan aset dan material, serta menentukan kebutuhan sistem seperti pemindaian perangkat/material menggunakan kamera *device,* pendeteksian *serial number* dan *scan code* secara otomatis, validasi data ke *database* perusahaan secara *real-time,* pembuatan laporan inventaris, serta pemberian kode identitas unik permanen pada setiap aset untuk keaslian data.

2. *Design*

   Tahap *design* berfokus pada perancangan arsitektur sistem, rancangan basis data, serta antarmuka pengguna. (Bassil, 2012) menegaskan bahwa desain perangkat lunak berfungsi sebagai cetak biru *(blueprint)* yang menjabarkan kebutuhan menjadi rancangan teknis yang terstruktur. Pada proyek *INOVA*, desain mencakup perancangan gambaran umum sistem, *use case diagram, activity diagram, entity relationship diagram (ERD),* serta rancangan antarmuka berbasis *Next.js* akan disusun dengan menu utama seperti *login, dashboard inventaris, scanning perangkat/material* beserta *serial number* dan *scan code,* validasi hasil *scan, monitoring* inventaris *real-time,* serta laporan (harian/mingguan dalam format *excel*) yang dapat diakses sesuai peran pengguna.

3. *Implementation*

   Implementation adalah tahap penerjemahan desain menjadi kode program yang siap dijalankan. Menurut (Bassil, 2012) menegaskan bahwa implementasi bertujuan menghasilkan produk nyata dari rancangan

konseptual. Dalam proyek *INOVA*, implementasi dilakukan dengan membangun tampilan antarmuka berbasis *Next.js*, yang disusun melalui menu utama seperti *Login*, *Dashboard* Inventaris, *scanning perangkat/material* beserta *serial number* dan *scan code*, validasi hasil *scan*, *Monitoring* Inventaris *Real-Time*, serta Laporan Harian/Mingguan dalam format *excel*, di mana setiap menu dapat diakses sesuai peran pengguna (*Superadmin* atau *Admin*).

Selanjutnya, melakukan proses pelatihan *YOLO* untuk mendeteksi area *serial number* atau *scan code* dari perangkat atau material, kemudian *OCR* untuk mengekstraksi teks, yang diintegrasikan melalui *Flask* dan *REST API* agar hasil deteksi langsung terhubung ke antarmuka sistem. Implementasi juga mencakup perancangan basis data *PostgreSQL* untuk menyimpan informasi perangkat atau material dalam satu tabel *assets*, hasil validasi, riwayat dan bukti pengecekan, serta laporan otomatis. Integrasi ini membuat *INOVA* mampu mengotomatiskan alur inventaris, dari pemindaian hingga validasi dan pelaporan data yang akurat.

4. *Testing*

Tahap *testing* dilakukan untuk memastikan sistem berjalan sesuai kebutuhan. Menurut (Munthe, 2017), pengujian bertujuan menemukan kesalahan dan menjamin kualitas sebelum sistem digunakan. Pada proyek *INOVA*, digunakan metode *Black Box Testing* dengan menguji fitur berdasarkan *output* tanpa melihat kode program. Pengujian mencakup seluruh menu utama, disertai *Users* Acceptance Testing (UAT) bersama pembimbing, *Asst Manager*, *IT Officer*, dan Teknisi *IT* untuk menilai kepraktisan sistem dalam kegiatan inventaris. Pada tahap ini, pengguna mencoba berbagai skenario, seperti pemindaian perangkat atau material, validasi *serial number* atau *scan code*, dan pengunduhan laporan, untuk sistem benar-benar meningkatkan efisiensi, akurasi dan keamanan aset.

5. *Deployment*

Tahap *deployment* dilakukan setelah sistem lulus pengujian (Munthe, 2017), dimana sistem di-*deploy* ke *server* internal perusahaan agar

dapat digunakan Seluruh *internal* Departemen *IT Infrastructure*. Kegiatan ini meliputi konfigurasi jaringan, pengaturan *database PostgreSQL*, integrasi aplikasi dengan intranet perusahaan, serta manajemen hak akses pengguna sesuai perannya. Selain itu, instalasi dependensi seperti *Flask* dan pustaka *computer vision*, autentikasi dan enkripsi agar akses inventaris lebih aman.

6. *Maintenance*

Tahap *maintennace* berfokus agar sistem tetap stabil, aman, dan akurat digunakan dalam jangka panjang (Munthe, 2017). Kegiatan meliputi perbaikan fitur lama dan bug, penambahan fitur baru, serta pembaruan *database PostgreSQL*. Model *AI* juga diperbarui secara berkala dengan *dataset* baru agar akurasi deteksi dan validasi *serial number* atau *scan code* tetap tinggi. Selain itu, *maintenance server* di *monitoring* agar tidak mengalami *downtime* sehingga sistem selalu siap digunakan kapan saja.

BAB III
ANALISIS DAN PERANCANGAN

3.1. Analisis Kebutuhan

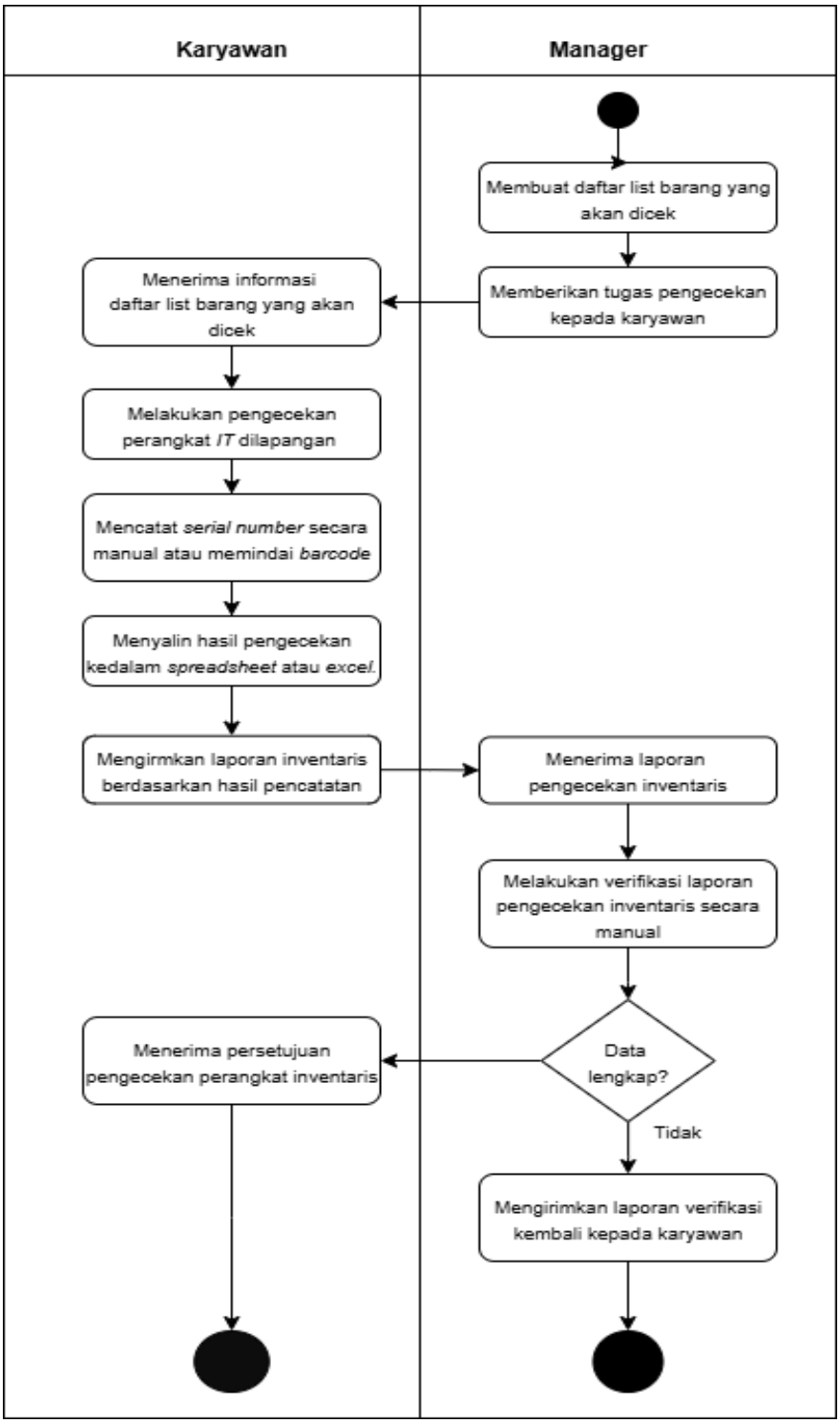
Analisis kebutuhan sistem merupakan tahap awal pengembangan yang mencakup identifikasi kebutuhan fungsional dan *non*-fungsional, serta pemetaan proses bisnis inventaris PT XYZ Indonesia. Tahap ini menyoroti kelemahan pencatatan manual dan *barcode*, kemudian solusi berbasis *AI* untuk otomatisasi deteksi dan validasi *serial number* atau *scan code*. Hasil analisis menjadi dasar perancangan *use case*, skenario interaksi, *activity diagram, ERD*, serta antarmuka *Next.js* dengan fitur *login, dashboard,* pemindaian, dan validasi *serial number* atau *scan code, monitoring real-time,* dan laporan inventaris perangkat atau material.

3.1.1 Proses Bisnis Berjalan

Proses bisnis sistem inventaris aset *IT* di PT XYZ Indonesia sebagaimana dijelaskan pada Gambar 3.1 dimana sistem inventaris masih dilakukan secara manual menggunakan *spreadsheet* dan *scan code* sederhana. Proses dimulai dari *Manager* mengirimkan daftar barang untuk dicek. Kemudian karyawan melakukan pengecekan perangkat di lapangan, memeriksa dan mencatat *serial number*, lalu menyalin data ke *spreadsheet* untuk disusun menjadi laporan inventaris. Laporan tersebut diverifikasi secara manual oleh *Manager*, dan akan disetujui jika data lengkap atau mengembalikannya bila terdapat kesalahan.

Namun, metode manual ini memiliki berbagai kendala *barcode* perangkat sering kali rusak atau hilang sehingga sulit dipindai, pencatatan *serial number* atau *barcode* rentan terjadi kesalahan *input* dan memakan banyak waktu, serta tidak adanya bukti autentik (seperti foto atau *log* waktu) membuat data mudah dimanipulasi. Selain itu, proses verifikasi laporan yang masih manual membuat alur kerja menjadi lambat, tidak efisien, dan meningkatkan risiko kehilangan atau ketidaksesuaian data aset. Oleh karena itu, dibutuhkan sistem inventaris perangkat berbasis *AI* yang mampu melakukan deteksi perangkat atau material secara otomatis sehingga dapat mempercepat pencatatan, meningkatkan akurasi data,

menyediakan bukti pengecekan secara *real-time*, serta mempermudah verifikasi dan persetujuan laporan secara digital dan terstruktur.



Gambar 3. 1 *Activity Diagram* Proses Bisnis yang Sedang Berjalan

3.1.2 Gambaran Umum Sistem

Gambar 3. 2 Gambaran Umum Sistem

Sistem inventaris perangkat *IT* (*INOVA*) pada Gambar 3.2 dirancang untuk meningkatkan efisiensi, akurasi, dan keamanan dalam pengelolaan aset *IT* di PT XYZ Indonesia. Sistem ini mengintegrasikan teknologi deteksi objek menggunakan *YOLOv8* untuk mengenali perangkat atau material serta membaca *serial number* atau *scan code*, dan teknologi *OCR* untuk mengekstrak teks, yang dihubungkan melalui *framework Next.js* sebagai antarmuka pengguna dan *framework Flask* sebagai pengelola logika sistem. Setiap permintaan dari pengguna akan dikirim melalui antarmuka sistem ke *server* melalui *API*, kemudian diproses sesuai alur kerja yang telah dirancang. Data hasil *scanning*, validasi, dan inventaris disimpan dan dikelola dalam *database PostgreSQL*. Proses inventaris dilakukan secara terstruktur mulai dari pembuatan *scanning preparation*, proses pemindaian melalui

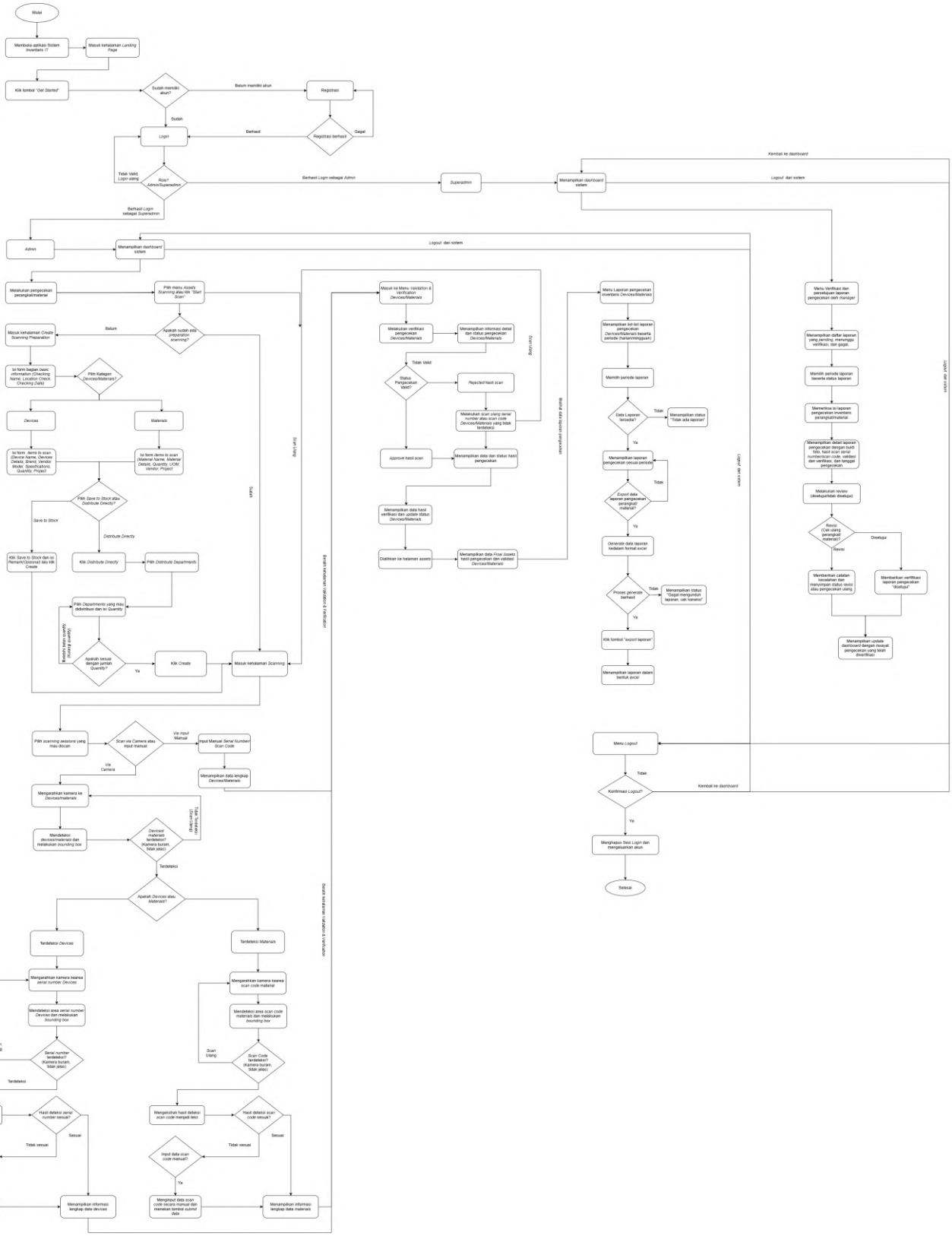
kamera atau input manual, validasi data, data aset final, hingga laporan inventaris dalam bentuk *excel*, dimana seluruh data ditampilkan melalui dashboard secara *real-time* untuk menjaga transparansi dan konsistensi data.

Sistem ini melibatkan dua aktor utama, yaitu *Admin* dan *Superadmin*. *Admin* melakukan proses inventaris seperti mengisi *scanning preparation*, melakukan *scanning* perangkat atau material, memilih metode penyimpanan (*save to stock* atau *distribute directly*), serta melakukan validasi hasil *scan* sebelum data disimpan sebagai aset. Sementara itu, *Superadmin* berperan dalam memantau aktivitas inventaris, meninjau laporan secara berkala, melakukan verifikasi dan persetujuan laporan, serta mengelola hak akses pengguna termasuk reset *password*. Interaksi antara pengguna dan sistem didukung oleh teknologi *computer vision* dan *OCR* yang memproses gambar untuk mendeteksi dan mengekstrak informasi, kemudian hasilnya dikembalikan ke sistem untuk ditampilkan dan divalidasi. Seluruh data pengguna, aset, hasil *scan*, laporan, dan riwayat aktivitas tersimpan dalam *database* dan diakses secara *real-time*, sehingga sistem tidak hanya sebagai alat pencatatan inventaris aset, tetapi juga *monitoring* dan pengambilan keputusan yang lebih efektif.

3.2. Perancangan

Perancangan sistem inventaris perangkat *IT* berbasis *AI* di PT XYZ Indonesia dilakukan untuk memenuhi kebutuhan pengguna sekaligus mengatasi permasalahan pada proses inventaris manual yang masih berjalan. Perancangan ini tidak hanya berfokus pada pengembangan fitur didalam sistem, tetapi juga mempertimbangkan bagaimana sistem dapat meningkatkan efisiensi, akurasi, serta kemudahan dalam pengelolaan inventaris aset. Oleh karena itu, perancangan sistem diawali dengan penyusunan alur sistem yang digambarkan melalui *flowchart*, kemudian dilanjutkan dengan identifikasi kebutuhan fungsional dan *non*-fungsional sebagai dasar dalam pengembangan sistem. Kebutuhan fungsional menggambarkan fitur dan layanan yang harus tersedia dalam sistem agar dapat digunakan sesuai tujuan dan alur kerja yang telah dirancang. Sedangkan kebutuhan *non*-fungsional berfokus pada kualitas sistem, seperti performa, keamanan, dan kehandalan sehingga sistem dapat digunakan secara optimal oleh *admin* maupun *superadmin*.

3.2.1 *Flowchart* sistem yang dirancang



Gambar 3. 3 *Flowchart* sistem yang dirancang

Proses *flowchart* sistem pada Gambar 3.3 yang dirancang di atas menunjukkan alur proses sistem inventaris aset *IT* berbasis *web* yang melibatkan dua aktor utama, yaitu *Superadmin* dan *Admin*, serta Sistem. Proses dimulai ketika pengguna membuka aplikasi dan masuk ke halaman *Landing Page*. Jika belum memiliki akun, pengguna melakukan registrasi; jika sudah, pengguna *login* dan sistem memverifikasi kredensial, lalu diarahkan ke *dashboard* sesuai *role*.

Pengguna dengan *role Admin* memilih menu *Assets Scanning*; jika *session* tersedia langsung *scanning*, jika belum maka membuat *Scanning Preparation* dengan mengisi *checking name*, *location*, *checking date*, serta memilih kategori *devices* atau *materials*. Data yang diisi meliputi detail item sesuai kategori, kemudian memilih metode *save to stock* atau *distribute directly* dengan pengisian jumlah sesuai batas *quantity*, lalu *create* untuk masuk ke halaman *scanning*. Pada proses *scanning*, pengguna memilih *session* dan melakukan *scan* via kamera atau input manual; *devices* membaca *serial number* dan *materials* membaca *scan code*, dengan opsi input manual jika tidak sesuai. Data hasil *scan* dikirim ke *validation*, jika valid di-*approve*, jika tidak valid di-*reject* dan kembali ke *scanning*. Data tervalidasi ditampilkan dan tersimpan sebagai aset pada halaman *Assets*.

Pada menu Laporan Pengecekan Inventaris, pengguna memilih periode laporan sistem akan menampilkan data inventaris beserta status validasi, dan pengguna dapat mengekspor laporan ke dalam bentuk *excel*. Jika proses *generate* berhasil, sistem menampilkan notifikasi sukses dan menyimpan laporan.

Menu terakhir menu khusus untuk *Superadmin* yaitu verifikasi dan persetujuan laporan inventaris untuk meninjau laporan akhir serta pengelolaan hak akses pengguna seperti reset password. Jika data lengkap, *Superadmin* menyetujui laporan dengan *approval*, namun jika terdapat kesalahan, laporan dikembalikan untuk revisi atau pengecekan ulang. Terakhir, melalui menu *Logout*, pengguna memilih opsi keluar, sistem mengonfirmasi dan menghapus sesi *login*, dan kembali ke halaman *login*. Proses ini memastikan seluruh alur mulai dari autentikasi, pengecekan aset dengan *AI*, validasi, unggah bukti, pembuatan dan persetujuan laporan, hingga pengelolaan akses dan *logout* berjalan secara terstruktur dan *real-time* sesuai perancangan sistem.

3.2.2 Kebutuhan Fungsional

Kebutuhan fungsional atau *Functional requirements* merupakan kebutuhan yang berisi rangkaian persyaratan yang harus dipenuhi oleh sistem aplikasi untuk melakukan fungsi dan prosesnya secara efektif dan efisien. Pada sistem inventaris *IT* berbasis *AI* di PT XYZ Indonesia, kebutuhan ini meliputi autentikasi pengguna, *scanning preparation*, pemindaian perangkat atau material melalui kamera atau input manual, deteksi dan ekstraksi *serial number* atau *scan code* secara otomatis, validasi data hasil *scanning (approve/reject)*, pengelolaan data aset final, pembuatan laporan inventaris, serta *monitoring* data aset secara *real-time*, sehingga *Admin* dan *Superadmin* dapat menjalankan perannya dengan sesuai dengan alur sistem yang telah dirancang. Rincian kebutuhan fungsional terdapat pada Tabel 3.1.

Tabel 3. 1 Kebutuhan Fungsional

Aktor	No	Fitur	Deskripsi
Admin	FR-01	Registrasi	Pengguna baru dapat mendaftar dengan memasukan *usersname*, email, No. *badge*, *Department*, *Position* dan *password* serta konfirmasi *password*.
	FR-02	*Login*	Pengguna dapat masuk ke dalam sistem dengan memasukkan *email* dan *password*.
	FR-03	Melihat halaman *dashboard*	Pengguna dapat melihat halaman *dashboard* berisi ringkasan data aset, status validasi, aktivitas *scanning*, laporan pengecekan terbaru, dan *monitoring* secara *real-time*.
	FR-04	*Scanning Preparation*	Pengguna dapat membuat *scanning preparation* dengan mengisi *checking name, location, checking date,* memilih kategori *(devices/materials)*, serta mengisi detail item sesuai kategori.

	FR-05	Pengaturan Metode Penyimpanan	Pengguna dapat memilih metode *save to stock* atau *distribute directly* dengan pengisian jumlah sesuai batas *quantity*.
	FR-06	*Scanning* Perangkat atau Material	Pengguna melakukan proses *scanning devices/materials* berdasarkan *session* menggunakan kamera atau input manual.
	FR-07	*Scanning* dan ekstraksi teks *Serial Number/Scan Code*.	Pengguna melakukan proses *scanning* pada *serial number/scan code*, dengan opsi input manual jika hasil tidak sesuai.
	FR-08	Validasi & Verifikasi Data Hasil S*can*	Pengguna melakukan validasi hasil *scanning* dengan status *approve* atau *reject*.
	FR-09	Melihat Data *Assets*	Pengguna dapat melihat data aset (perangkat atau material) hasil *scan* dan validasi yang tersimpan pada sistem.
	FR-10	Mengelola	Pengguna dapat melihat dan mengunduh laporan inventaris (harian/mingguan) dalam format *excel*.
	FR-11	*Logout*	Pengguna melakukan *logout*.
Sistem	FR-12	Deteksi otomatis	Sistem mendeteksi objek perangkat atau material dari input gambar.
	FR-13	Deteksi dan Ekstrak Teks Otomatis	Sistem mendeteksi dan mengestrak teks *serial number* atau *scan code* dan hasil deteksi.
	FR-14	Pengiriman dan	Sistem mengirim dan menyimpan data hasil *scanning*, validasi, dan data asek ke *database*.

		Penyimpanan Data	
	FR-15	Menampilkan Hasil *Scan* dan Validasi	Sistem menampilkan data hasil *scanning*, beserta status validasi secara *real-time*.
	FR-16	Menampilkan Data *Assets*	Sistem menampilkan data aset (perangkat atau material) hasil *scanning* dan validasi data.
	FR-17	Pengelolaan Laporan Inventaris	Sistem menampilkan data hasil pengecekan berupa laporan inventaris perangkat atau material *IT* (harian/mingguan) dalam format *excel.*
Superadmin	FR-18	Registrasi	Pengguna baru dapat mendaftar dengan memasukan *usersname*, email, No. *badge*, *Department*, *Position* dan *password* serta konfirmasi *password*.
	FR-19	*Login*	Pengguna dapat masuk ke dalam sistem dengan memasukkan *email* dan *password*.
	FR-20	Melihat halaman *dashboard*	Pengguna dapat melihat halaman *dashboard* berisi ringkasan jumlah aset, status validasi hasil *scanning*, laporan pengecekan, serta grafik monitoring inventaris secara *real-time*.
	FR-21	*Monitoring* Inventaris Perangkat atau material *IT*	Pengguna dapat memantau kondisi pengecekan inventaris perangkat atau material *IT* secara *real-time*.

	FR-22	Verifikasi Laporan Pengecekan	Pengguna dapat memeriksa laporan serta memberikan persetujuan atau masukan.
	FR-23	Manajemen Hak Akses Pengguna	Pengguna dapat mengelola hak akses seluruh pengguna sistem termasuk reset *password*.
	FR-24	*Logout*	Pengguna dapat keluar dari sistem.

3.2.3 Kebutuhan *Non-Fungsional*

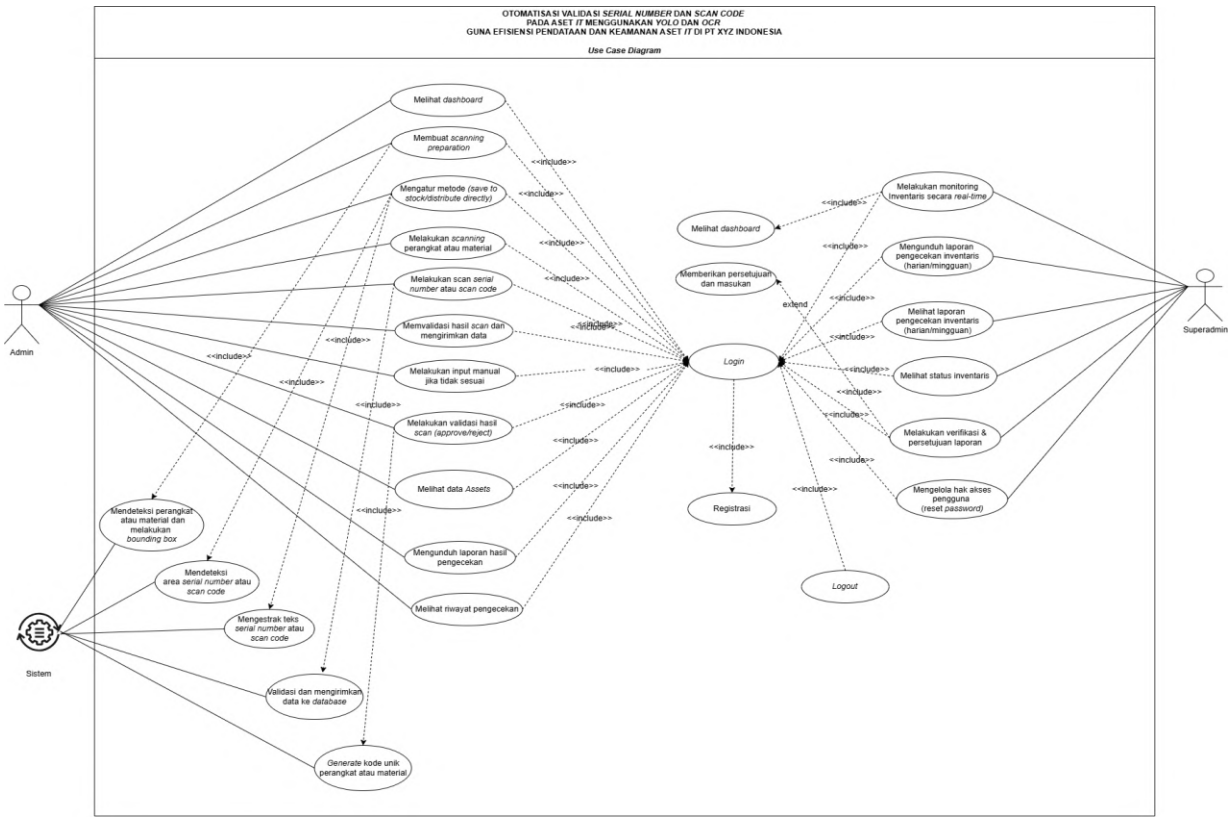
Kebutuhan *Non*-Fungsional atau *Non-functional requirements* adalah persyaratan yang tidak berkaitan langsung dengan fungsionalitas system atau aplikasi, namun berfokus pada aspek teknis, kinerja dan pengalaman pengguna. Pada sistem inventaris perangkat atau material *IT* berbasis *AI*, sistem harus mampu memberikan pengalaman pengguna yang aman dan responsif serta mampu menangani data besar, memberi respon cepat, menjaga keamanan dari manipulasi, serta memiliki mekanisme *backup* untuk keandalan data. Rincian kebutuhan non-fungsional terdapat pada Tabel 3.2.

Tabel 3. 2 Kebutuhan *Non-Fungsional*

No	Kebutuhan *Non-Fungsional*
NF-01	Sistem harus memiliki performa yang baik, mampu menangani banyak pengguna secara bersamaan, serta mampu memberikan waktu respons yang cepat dan konsisten pada setiap permintaan pengguna tanpa mengalami penurunan performa yang signifikan.
NF-02	Sistem harus memiliki mekanisme keamanan yang memadai untuk melindungi data sensitif melalui autentikasi pengguna, enkripsi data, serta pencegahan terhadap akses tidak sah dan serangan umum seperti *SQL Injection*, Cross-Site Scripting (XSS), dan *brute force attack.*

3.2.4 *Use Case Diagram*

Use case diagram pada Gambar 3.4 menunjukkan interaksi antara aktor dengan sistem inventaris barang *IT*. Terdapat tiga aktor utama, yaitu *Admin*, *Superadmin*, dan Sistem.



Gambar 3. 4 *Use Case Diagram*

Admin memiliki kemampuan melakukan registrasi dan *login* ke dalam sistem, kemudian mengakses *dashboard* untuk melihat ringkasan data aset, status validasi, serta aktivitas *scanning*. *Admin* selanjutnya membuat *scanning preparation* dengan mengisi *checking name*, *location*, *checking date*, memilih kategori *devices* atau *materials*, serta menginput detail item, kemudian mengatur metode penyimpanan yaitu *save to stock* atau *distribute directly*. Pada proses berikutnya, *Admin* melakukan pemindaian perangkat atau material melalui kamera atau input manual, dimana sistem akan membaca *serial number* jika objek berupa perangkat dan *scan code* jika objek berupa material, serta memungkinkan input manual apabila hasil tidak sesuai. Data hasil *scanning* kemudian divalidasi dengan status *approve* atau *reject* sebelum dikirim dan disimpan sebagai data aset. *Admin*

juga dapat melihat data aset beserta kode unik, memantau status validasi, mengakses riwayat pengecekan, serta mengunduh laporan inventaris harian atau mingguan.

Di sisi lain, *Superadmin* berperan dalam mengawasi proses inventaris dengan melakukan *login* ke sistem, memantau data secara *real-time* melalui *dashboard*, melihat laporan pengecekan inventaris, serta memberikan persetujuan atau masukan terhadap laporan tersebut. Selain itu, Superadmin juga dapat mengelola hak akses pengguna termasuk melakukan reset *password* untuk menjaga keamanan sistem. Sistem sendiri berperan dalam mendeteksi objek perangkat atau material, mendeteksi area *serial number* atau *scan code*, mengekstrak teks hasil *scanning*, serta mengirimkan dan menyimpan data ke *database*, termasuk menghasilkan kode unik untuk setiap perangkat atau material. Seluruh interaksi ini mendukung otomatisasi inventarisasi serta meningkatkan efisiensi, akurasi, dan keamanan pengelolaan aset *IT* di PT XYZ Indonesia.

3.2.5 *Scenario Use Case*

1) *Use Case* Registrasi *Admin* dan *Superadmin*

Tabel 3. 3 *Use Case Scenario Register*

Use Case Scenario 1: Register	
Nomor	UC-01
Nama	*Register*
Deskripsi	Pengguna yang ingin mengakses Sistem Inventaris *IT INOVA* harus mendaftarkan akun terlebih dahulu dengan mengisi formulir registrasi.
Primary Aktor	*Admin* dan *Superadmin*
Preconditions	Pengguna belum memiliki akun dalam sistem.
Postconditions	Pengguna memiliki akun yang terdaftar dalam aplikasi dan dapat melakukan *login* ke dalam sistem.
Main Succes Scenario	1. Pengguna membuka halaman aplikasi *INOVA*. 2. Pengunjung memilih opsi “Daftar”.

	3. Pengguna mengisi form registrasi dengan data yang diperlukan (*Name*, Email, No. *Badge*, *Departement*, *Position*, *Password*, Konfirmasi *Password*). 4. Sistem memvalidasi data yang dimasukkan pengguna. 5. Jika valid, sistem menyimpan data pengguna ke dalam *database*. 6. Pengguna menerima notifikasi bahwa registrasi berhasil. 7. Sistem mengarahkan pengguna ke halaman *login*.
Alternative Flow (Extention)	3a. Pengguna tidak mengisi *form* registrasi dengan lengkap. • 3b. Sistem memberi peringatan kepada pengguna "Data wajib diisi dengan lengkap". 4a. Data Email yang di *input* pengguna sudah terdaftar. • 4b. Sistem menampilkan pesan "Email sudah digunakan". 5a. Pengguna tidak memiliki koneksi internet saat proses registrasi. • 5b. Sistem menampilkan pesan kesalahan "Registrasi gagal, periksa koneksi" dan meminta mencoba kembali.

2) *Use Case Scenario Login Admin* dan *Superadmin*

Tabel 3. 4 *Use Case Scenario Login*

Use Case Scenario 2: Login	
Nomor	UC-02
Nama	*Login*
Deskripsi	Pengguna yang sudah terdaftar dan memilki akun melakukan *login* kedalam aplikasi dengan mengisi form yang telah disediakan.
Primary Aktor	*Admin* dan *Superadmin*
Preconditions	Pengguna telah terdaftar dalam sistem.

Postconditions	Pengguna berhasil *login* dan diarahkan kehalaman *dashboard* sesuai *role*.
Main Succes Scenario	1. Pengguna membuka halaman aplikasi *INOVA*. 2. Pengunjung memilih opsi "*Login*". 3. Pengguna mengisi *form login* dengan data yang diperlukan (Email, *Password*). 4. Sistem memvalidasi data yang dimasukkan pengguna. 5. Jika valid, sistem mengarahkan pengguna ke *dashboard* aplikasi.
Alternative Flow (Extention)	3a. Pengguna lupa *password*. • 3b. Pengguna memilih opsi "Lupa *Password*", dan sistem menampilkan halaman pemulihan akun untuk mengirimkan tautan *reset* ke email. 4a. Pengguna memasukkan email/*password* salah. • 4b. Sistem menampilkan pesan "Email atau *Password salah",* dan meminta pengguna untuk mengisi ulang data yang benar. 5a. Pengguna tidak memiliki koneksi internet saat proses *login*. • 5b. Sistem menampilkan pesan kesalahan "*Login* gagal, periksa koneksi" dan meminta mencoba kembali.

3) *Use Case Scenario Dashboard Admin* dan *Superadmin*

Tabel 3. 5 *Use Case Scenario Dashboard*

Use Case Scenario 3: Dashboard	
Nomor	UC-03
Nama	*Dashboard*
Deskripsi	Pengguna yang *login*, akan dapat melihat *dashboard* berisi ringkasan data aset, status validasi hasil *scanning*, aktivitas pengecekan, laporan pengecekan terbaru, serta grafik monitoring inventaris secara *real-time*.

Primary Aktor	*Admin* dan *Superadmin*
Preconditions	Pengguna telah berhasil *login* kedalam sistem.
Postconditions	*Dashboard* menampilkan informasi *real-time* inventaris pada *dashboard* sesuai *role* pengguna.
Main Succes Scenario	1. Pengguna berhasil *login* kedalam sistem. 2. Sistem menampilkan halaman *dashboard* sesuai *role*. 3. Sistem menampilkan ringkasan jumlah seluruh aset. 4. Sistem menampilkan aktivitas pengecekan atau *scanning* terbaru. 5. Sistem menampilkan status validasi hasil *scanning (approve/reject)*. 6. Sistem menampilkan data laporan pengecekan terbaru. 7. Sistem menampilkan grafik monitoring inventaris secara *real-time*
Alternative Flow (Extention)	3a. Data aset belum tersedia. • 3b. Sistem menampilkan pesan “Data aset belum tersedia, lakukan pengecekan terlebih dahulu”. 4a. Aktivitas *scanning* belum tersedia. • 4b. Sistem menampilkan pesan “Belum ada aktivitas pengecekan”. 5a. Data validasi tidak tersedia • 5b. Sistem menampilkan *placeholder* dengan pesan “Belum ada yang divalidasi”. 6a. Data laporan pengecekan terbaru tidak tersedia. • 6b. Sistem menampilkan pesan “Belum ada pengecekan terbaru”. 7a. Grafik gagal dimuat karena gangguan sistem/koneksi • 7b. Sistem menampilkan pesan ”Data grafik tidak tersedia”.

4) *Use Case Scenario Scanning Preparation*

Tabel 3. 6 *Use Case Scenario Scanning Preparation*

Use Case Scenario 4: Scanning Preparation	
Nomor	UC-04
Nama	*Scanning Preparation*
Deskripsi	Pengguna membuat *scanning preparation* dengan mengisi data pengecekan seperti *checking name, location, checking date,* memilih kategori perangkat atau material, serta mengisi detail item yang akan dilakukan pengecekan.
Primary Aktor	*Admin*
Preconditions	Pengguna telah berhasil *login* kedalam sistem.
Postconditions	Data *scanning preparation* berhasil disimpan dan siap dilanjutkan ke pengaturan metode penyimpanan.
Main Succes Scenario	1. Pengguna berada dihalaman *dashboard*. 2. Pengguna memilih menu "Assets Scanning" dan mengklik "Create Scanning Preparation". 3. Sistem menampilkan form scanning preparation. 4. Pengguna mengisi data basic information yaitu checking name, location, dan checking date. 5. Pengguna memilih kategori *(devices* atau *materials)*. 6. Pengguna mengisi *items detail* sesuai kategori. 7. Sistem menyimpan data sementara. 8. Sistem mengarahkan untuk pemilihan metode penyimpanan.
Alternative Flow (Extention)	4a. Data tidak lengkap • 4b. Sistem menampilkan pesan "Data harus diisi lengkap". 5a. Kategori tidak dipilih • 5b. Sistem menampilkan pesan "Pilih kategori terlebih dahulu". 6a. Detail item tidak valid atau kosong.

	• 6b. Sistem meminta pengguna untuk mengisi seluruh detail item.

5) *Use Case Scenario* Pengaturan Metode Penyimpanan

Tabel 3. 7 *Use Case Scenario* Pengaturan Metode Penyimpanan

Use Case Scenario 5: Pengaturan Metode Penyimpanan	
Nomor	UC-05
Nama	Pengaturan Metode Penyimpanan
Deskripsi	Pengguna menentukan metode penyimpanan untuk *scanning* yaitu *save to stock* atau *distribute directly* serta mengisi jumlah item, kemudian menyimpan seluruh data *scanning preparation* untuk memulai proses *scanning*.
Primary Aktor	*Admin*
Preconditions	Pengguna telah mengisi data *scanning preparation* bagian *basic information* dan *items detail*.
Postconditions	Data *scanning preparation* beserta metode penyimpanan tersimpan dan *session scanning* berhasil dibuat.
Main Succes Scenario	1. Pengguna berada pada tahap pengaturan metode penyimpanan. 2. Sistem menampilkan opsi metode penyimpanan. 3. Pengguna memilih metode (save to stock atau distribute directly). 4. Pengguna mengisi jumlah item sesuai *quantity*. 5. Pengguna menekan tombol *create*. 6. Pengguna mengisi detail item sesuai kategori. 7. Sistem memvalidasi seluruh data *(preparation* dan *method)*. 8. Sistem menyimpan data ke *database* sebagai *session scanning*. 9. Sistem menampilkan notifikasi bahwa data berhasil dibuat.

	10. Sistem mengarahkan pengguna ke halaman *scanning*.
Alternative Flow (Extention)	3a. Metode tidak dipilih • 3b. Sistem menampilkan pesan "Pilih metode penyimpanan". 4a. Jumlah melebih batas *quantity* • 4b. Sistem menampilkan pesan "Jumlah melebihi batas *quantity*". 4c. Jumlah tidak diinput • 4d. Sistem menampilkan pesan "Jumlah harus diisi". 6a. Validasi data gagal. • 6b. Sistem menampilkan pesan "Data tidak valid, periksa kembali". 6c. Detail item kosong • 6d. Sistem menampilkan pesan "Detail item wajib diisi". 7a. Gagal menyimpan data • 7b. Sistem menampilkan pesan "Gagal menyimpan data, coba kembali".

6) *Use Case Scenario Scanning* Perangkat atau Material

Tabel 3. 8 *Use Case Scenario Scanning* Perangkat atau Material

Use Case Scenario 6: Scanning Perangkat atau Material	
Nomor	UC-06
Nama	*Scanning* Perangkat atau Material
Deskripsi	Pengguna melakukan pemindaian perangkat atau material menggunakan kamera atau input manual berdasarakan *session* yang telah dibuat.
Primary Aktor	*Admin*
Preconditions	Pengguna telah berhasil *login* dan memiliki *session scanning* serta perangkat kamera aktif.

Postconditions	Perangkat atau material berhasil dipindai dan lanjut untuk proses *scan serial number* atau *scan code*.
Main Succes Scenario	1. Pengguna berada pada halaman *scanning*. 2. Sistem menampilkan *session scanning* yang aktif. 3. Pengguna memilih *session scanning*. 4. Sistem mengaktifkan kamera *(device/webcam)*. 5. Pengguna mengarahkan kamera ke objek (perangkat atau material). 6. Sistem mengenali jenis objek (perangkat atau material). 7. Sistem menampilkan hasil deteksi objek pada layar. 8. Pengguna melanjutkan ke proses *scanning serial number* atau *scan code*.
Alternative Flow (Extention)	4a. Kamera tidak aktif atau tidak tersedia • 4b. Sistem menampilkan pesan ”Kamera tidak tersedia, silakan periksa perangkat”. 5a. Objek tidak terdeteksi dengan jelas • 5b. Sistem menampilkan pesan “*Objek* tidak terbaca, coba ulangi lagi” atau sistem menampilkan pesan jika kondisi kamera terlalu buram “*Objek* tidak terdeteksi, kamera terlalu buram, silahkan ulangi lagi”. 6a. Sistem gagal mendeteksi objek. • 6b. Sistem menampilkan pesan ”Gagal mendeteksi objek, coba kembali”. 6c. Objek terdeteksi tidak sesuai dengan *session scanning*. • 6d. Sistem menampilkan pesan ”Objek tidak sesuai dengan *session*, silakan ulangi lagi”.

7) *Use Case Scenario Scanning* & Ekstraksi *Serial Number* atau *Scan Code*

Tabel 3. 9 *Use Case Scenario Scanning* & Ekstraksi *Serial Number/Scan Code*

Use Case Scenario 7: Scanning & Ekstraksi *Serial Number* atau *Scan Code*	
Nomor	UC-07
Nama	*Scanning* & Ekstraksi *Serial Number* atau *Scan Code*
Deskripsi	Setelah proses *scanning* perangkat atau material, Pengguna melakukan *scanning* pada *serial number* atau *scan code*, kemudian sistem mengekstrak teks dan menampilkan hasilnya, dengan opsi input manual jika tidak sesuai.
Primary Aktor	*Admin*
Preconditions	Objek perangkat atau material telah berhasil dideteksi.
Postconditions	Data *serial number* atau *scan code* berhasil diekstrak dan ditampilkan untuk proses validasi.
Main Succes Scenario	1. Pengguna berada dihalaman *scanning* dan berada pada proses *scanning serial number* atau *scan code*. 2. Sistem mendeteksi *area serial number* atau *scan code* pada objek (perangkat atau material). 3. Sistem melakukan proses ekstraksi teks menggunakan *ocr*. 4. Sistem menampilkan hasil ekstraksi teks kepada pengguna. 5. Pengguna memeriksa hasil teks yang ditampilkan. 6. Jika sesuai, pengguna melanjutkan ke proses validasi.
Alternative Flow (Extention)	2a. Sistem gagal membaca *serial number/scan code* atau teks tidak terdeteksi. • 2b. Sistem menampilkan pesan “Data tidak terbaca, silakan ulangi *scan*”. 3a. Hasil ekstraksi teks tidak jelas atau tidak akurat. • 3b. Sistem meminta pengguna melakukan *scan* ulang atau menggunakan input manual. 4a. Data terdeteksi duplikat.

	• 4b. Sistem menampilkan status "Data sudah terdaftar (duplikat)", dan memberikan opsi untuk *scan* ulang atau menggunakan input manual. 5a. Format *serial number* atau *scan code* tidak sesuai standar. • 5b. Sistem menampilkan pesan "Format *serial number/scan code* tidak valid", dan meminta pengguna *scan* ulang atau input manual..

8) *Use Case Scenario* Validasi & Verifikasi Data hasil *Scan*

Tabel 3. 10 *Use Case Scenario* Validasi & Verifikasi Data Hasil *Scan*

Use Case Scenario 8: Validasi & Verifikasi Data hasil *Scan*	
Nomor	UC-08
Nama	Validasi & Verifikasi Data hasil *Scan*
Deskripsi	Setelah proses *scanning*, pengguna melakukan validasi terhadap data hasil *scan devices/materials* dengan memberikan status *approve* atau *reject* sebelum data disimpan sebagai final *Assets*.
Primary Aktor	*Admin*
Preconditions	Pengguna telah melakukan proses *scanning* dan ekstraksi *serial number* atau *scan code*, serta data hasil *scan* sudah tersedia.
Postconditions	Data hasil *scan* tervalidasi dengan status *approve* atau *reject*, dan data yang disetujui disimpan sebagai data aset dalam sistem.
Main Succes Scenario	1. Pengguna berada dihalaman validasi dan verifikasi. 2. Sistem menampilkan hasil data *serial number* atau *scan code* beserta informasi perangkat atau material. 3. Sistem menampilkan status hasil deteksi untuk dilakukan validasi. 4. Pengguna memeriksa kesesuaian data hasil *scan*. 5. Pengguna memilih aksi validasi *(approve* atau *reject)*. 6. Jika *approve*, sistem menghasilkan kode unik untuk perangkat atau material.

	7. Sistem menyimpan data hasil validasi ke *database* sebagai data aset. 8. Sistem menampilkan notifikasi bahwa data berhasil divalidasi dan disimpan. 9. Jika *reject*, sistem mengembalikan proses ke tahap *scanning* untuk dilakukan pengecekan ulang.
Alternative Flow (Extention)	2a. Data hasil *scan* tidak lengkap atau tidak sesuai. • 2b. Sistem menampilkan pesan "Data tidak lengkap atau tidak valid" dan meminta pengguna melakukan pengecekan ulang. 5a. Data di *approve*. • 5b. Sistem menampilkan pesan "Data berhasil di *approve* dan mengirim data ke halaman *assets*" 5c. Data di *reject*. • Sistem mengarahkan pengguna kembali ke proses *scanning* untuk *scan* ulang.

9) *Use Case Scenario* Melihat Data *Assets*

Tabel 3. 11 *Use Case Scenario* Melihat Data *Assets*

Use Case Scenario 9: Melihat Data *Assets*	
Nomor	UC-09
Nama	Melihat Data *Assets*
Deskripsi	Pengguna dapat melihat data final *Assets* (perangkat atau material) hasil *scanning* dan validasi yang telah tersimpan dalam sistem.
Primary Aktor	*Admin*
Preconditions	Pengguna telah melakukan proses *scanning* dan validasi data hasil *scan*.
Postconditions	Data *assets* muncul dan ditampilkan dalam sistem secara lengkap.

Main Succes Scenario	1. Pengguna berada di *dashboard* dan memilih menu ”*Assets Inventory”*. 2. Pengguna berada dihalaman *Assets*. 3. Sistem mengambil data aset dari *database.* 4. Sistem menampilkan daftar aset perangkat atau material. 5. Sistem menampilkan detail data (nama aset, kategori, *serial number/scan code*, lokasi, status, tanggal). 6. Pengguna dapat melakukan pencarian atau filter data *assets* dan detail masing-masing *assets* berdasarkan *session preparation* yang dibuat sebelumnya.
Alternative Flow (Extention)	3a. Data aset belum tersedia. • 3b. Sistem menampilkan pesan ”Data aset belum tersedia”. 3c. Sistem gagal mengambil data dari *database*. • 3d. Sistem menampilkan status *“*Gagal memuat data aset periksa koneksi”. 6a. Hasil pencarian atau filter tidak ditemukan. • 6b. Sistem menampilkan pesan *“*Data tidak ditemukan sesuai kriteria”.

10) *Use Case Scenario* Laporan Pengecekan Inventaris

Tabel 3. 12 *Use Case Scenario* Laporan Pengecekan Inventaris

Use Case Scenario 10: Laporan Pengecekan Inventaris Perangkat atau Material	
Nomor	UC-10
Nama	Laporan Pengecekan Inventaris Perangkat atau Material
Deskripsi	Pengguna dapat melihat dan mengunduh laporan hasil list pengecekan inventaris perangkat atau material berdasarkan periode tertentu dalam format *excel*.
Primary Aktor	*Admin* dan *Superadmin*

Preconditions	Pengguna telah *login* dan data hasil *scanning* dan validasi serta data aset telah tersimpan dalam sistem.
Postconditions	Laporan hasil pengecekan inventaris perangkat berhasil diunduh dan ditampilkan dalam format *excel*.
Main Succes Scenario	1. Pengguna berada dihalaman *dashboard*. 2. Pengguna memilih menu "*Report & Analytics*". 3. Sistem menampilkan opsi periode pengecekan (harian/mingguan). 4. Pengguna memilih periode laporan pengecekan. 5. Pengguna melihat data detail item-item pengecekan perangkat/material sesuai periode dan memunculkan list nya dalam laporan. 6. Sistem mengambil data pengecekan perangkat/material sesuai periode dan menghasilkan file laporan. 7. Pengguna mengunduh laporan dalam format *excel*.
Alternative Flow (Extention)	2a. Data laporan pengecekan kosong • 2b. Sistem menampilkan pesan "Tidak ada data laporan untuk periode ini". 4a. Koneksi Gagal. • 4b. Sistem menampilkan status "Gagal mengunduh laporan, cek koneksi".

11) *Use Case Scenario Monitoring Real-Time*

Tabel 3. 13 *Use Case Scenario Monitoring Real-Time*

Use Case Scenario 11: Monitoring Real-Time	
Nomor	UC-11
Nama	*Monitoring Real-Time*
Deskripsi	Pengguna dapat memantau aktivitas pengecekan inventaris perangkat atau material secara *real-time* melalui *dashboard*.

	Sistem menampilkan status pengecekan, daftar aset, progres validasi, serta riwayat pengecekan berdasarkan periode tertentu.
Primary Aktor	*Superadmin*
Preconditions	Pengguna telah *login* ke dalam aplikasi dan memiliki hak akses untuk melihat data *monitoring*. Data hasil pengecekan perangkat atau material sudah tersimpan dalam *database*.
Postconditions	Sistem menampilkan informasi pengecekan perangkat dan progres validasi serta data *assets* dalam bentuk data numerik, grafik, dan tabel yang diperbarui secara *real-time*.
Main Succes Scenario	1. Pengguna berada di *dashboard* 2. Sistem mengambil seluruh data inventaris dari *database*. 3. Sistem menampilkan ringkasan jumlah aset. 4. Sistem menampilkan status validasi *(approve/reject).* 5. Sistem menampilkan grafik monitoring inventaris. 6. Sistem menampilkan aktivitas *scanning* dan validasi terbaru. 7. Pengguna dapat melakukan filter berdasarkan periode, status, atau kategori. 8. Sistem memperbarui data secara *real-time*.
Alternative Flow (Extention)	2a. Sistem gagal mengambil data dari *database* karena koneksi jaringan tidak stabil. • 2b. Sistem menampilkan pesan *"*Gagal memuat data, periksa koneksi jaringan Anda*"*. 3a. Tidak ada perangkat atau material yang terdaftar. • 3b. Sistem menampilkan pesan *"*Belum ada data perangkat atau material pada sistem*"*. 5a. Grafik gagal dimuat. • 5b. Sistem menampilkan pesan *"*Data grafik tidak tersedia". 7a. Filter tidak menghasilkan hasil apapun.

	• 7b. Sistem menampilkan pesan "Tidak ditemukan data sesuai kriteria pencarian".

12) *Use Case Scenario* Verifikasi Laporan Pengecekan

Tabel 3. 14 *Use Case Scenario* Verifikasi Laporan Pengecekan

Use Case Scenario 12: Verifikasi Laporan Pengecekan	
Nomor	UC-12
Nama	Verifikasi Laporan Pengecekan
Deskripsi	Pengguna dapat memverifikasi laporan hasil pengecekan inventaris dan memberikan persetujuan atau revisi.
Primary Aktor	*Superadmin*
Preconditions	Data laporan pengecekan sudah tersedia dalam sistem.
Postconditions	Laporan diverifikasi dengan status disetujui atau dikembalikan untuk revisi atau meminta untuk melakukan pengecekan ulang.
Main Succes Scenario	1. Pengguna telah *login* dan berada di *dashboard* sistem. 2. Pengguna membuka menu *"Report & Analytics"*. 3. Sistem menampilkan daftar list laporan inventaris. 4. Pengguna memilih salah satu laporan. 5. Sistem menampilkan detail laporan. 6. Pengguna memeriksa isi laporan. 7. Pengguna memberikan persetujuan atau revisi serta meminta untuk melakukan pengecekan ulang jika diperlukan. 8. Sistem menyimpan status verifikasi laporan. 9. Sistem menampilkan notifikasi bahwa verifikasi berhasil.
Alternative Flow (Extention)	5a. Data pengecekan salah satu perangkat atau material pada laporan belum lengkap. • 5b. Sistem menampilkan pesan "Data perangkat atau material belum lengkap".

	7a. Pengguna memilih revisi. • 7b. Sistem mengembalikan laporan ke *admin* untuk pengecekan ulang. 7c. Pengguna membatalkan proses. • 7d. Sistem kembali kehalaman *dashboard*

13) *Use Case Scenario* Manajemen Hak Akses Pengguna

Tabel 3. 15 *Use Case Scenario* Manajemen Hak Akses Pengguna

Use Case Scenario 13: Manajemen Hak Akses Pengguna	
Nomor	UC-13
Nama	Manajemen Hak Akses Pengguna
Deskripsi	Pengguna dapat mengelola hak akses seluruh pengguna seperti melihat data pengguna, mengubah *role*, dan melakukan reset *password*.
Primary Aktor	*Superadmin*
Preconditions	Pengguna telah berhasil *login* kedalam sistem.
Postconditions	Data hak akses pengguna sistem berhasil diperbaharui.
Main Succes Scenario	1. Pengguna berada didashboard. 2. Pengguna memilih menu "*Users Management*". 3. Sistem menampilkan daftar pengguna sistem. 4. Pengguna memilih salah satu pengguna. 5. Sistem menampilkan detail data pengguna. 6. Pengguna mengubah *role* atau reset *password*. 7. Sistem menyimpan perubahan data pengguna. 8. Sistem menampilkan notifikasi perubahan berhasil.
Alternative Flow (Extention)	3a. Data pengguna tidak tersedia • 3b. Sistem menampilkan pesan "Tidak ada data pengguna". 6a. Data input tidak valid. • 6b. Sistem menampilkan pesan "Data tidak valid".

	7a. Sistem gagal menyimpan perubahan. • 7b. Sistem menampilkan pesan ”Gagal menyimpan perubahan data”.

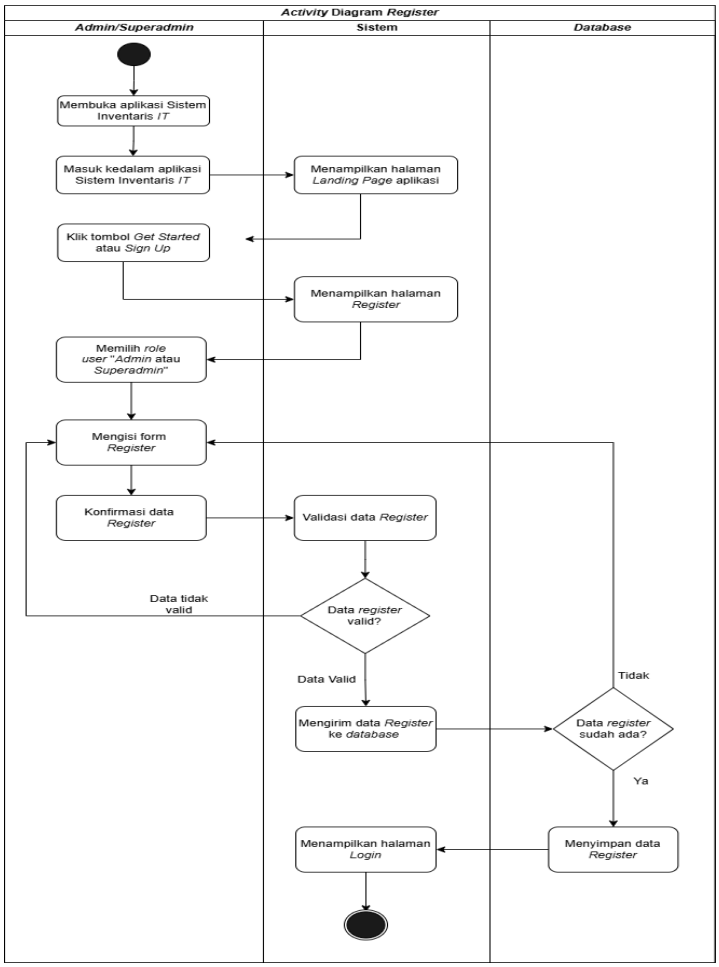
14) *Use Case Scenario Logout*

Tabel 3. 16 *Use Case Scenario Logout*

Use Case Scenario 14: Logout	
Nomor	UC-14
Nama	*Logout*
Deskripsi	Pengguna dapat keluar dari sistem setelah menyelesaikan aktivitas.
Primary Aktor	*Admin* dan *Superadmin*
Preconditions	Pengguna telah berhasil *login* kedalam aplikasi.
Postconditions	Pengguna keluar dari aplikasi dan harus *login* kembali untuk mengakses fitur-fitur yang membutuhkan autentikasi.
Main Succes Scenario	1. Pengguna membuka aplikasi dan masuk ke halaman “*profile”*. 2. Pengguna memilih opsi” *Logout”*. 3. Sistem menampilkan konfirmasi untuk memastikan apakah pengguna ingin keluar. 4. Pengguna menekan “Ýa” untuk mengkonfirmasi pilihan *logout*. 5. Sistem mengakhiri sesi pengguna dan menampilkan halaman *login*.
Alternative Flow (Extention)	3a. Pengguna membatalkan *logout*. • 3b. Pengguna memilih opsi "Batal" saat diminta konfirmasi *logout*. • 3b. Sistem tetap mempertahankan sesi aktif dan tetap berada diaplikasi.

3.2.6 *Activity Diagram*

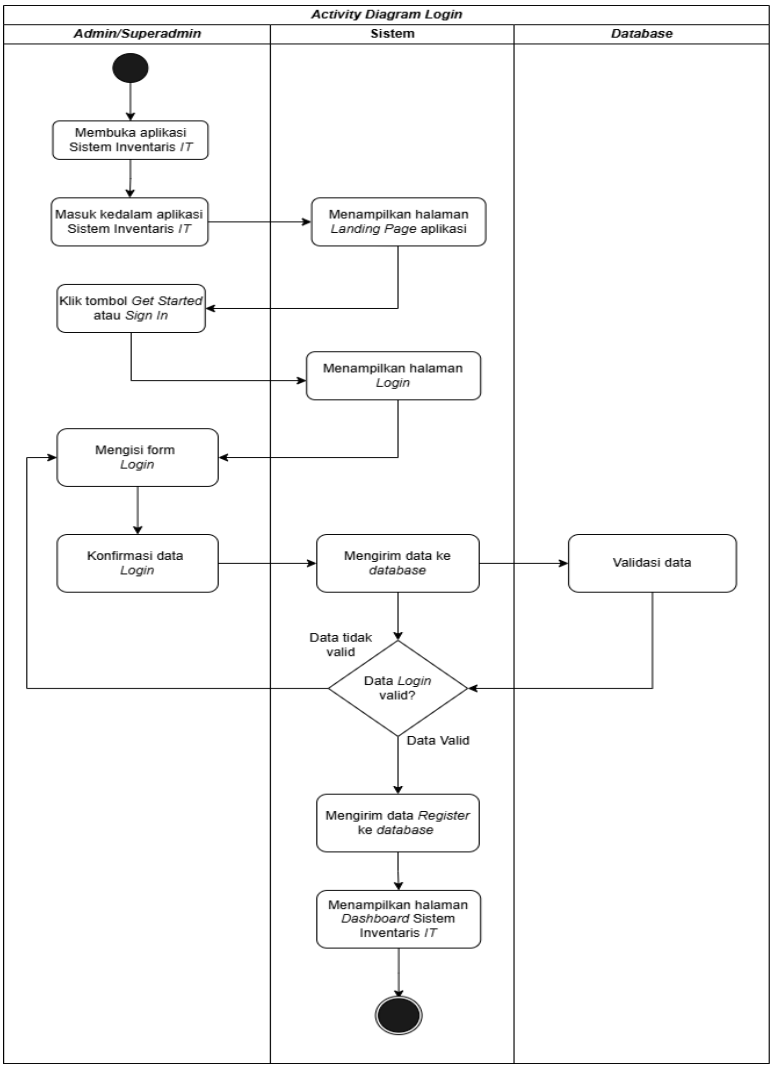
3.2.6.1 *Activity Diagram Register*



Gambar 3. 5 *Activity Diagram Register*

Proses *Activity Diagram Register* pada Gambar 3.5 menjelaskan alur registrasi pengguna (*Admin* atau *Superadmin*) pada Sistem Inventaris *IT*. Proses dimulai dengan masuk ke halaman utama, lalu menekan tombol *Get Started/Sign Up* untuk menuju halaman *Register*. Pengguna memilih role (*Admin* atau *Superadmin*) dan mengisi form registrasi, kemudian mengonfirmasi data yang dimasukkan. Sistem menerima data dan melakukan validasi. Jika data tidak valid pengguna diminta memperbaiki data. Jika valid, sistem memeriksa apakah data registrasi sudah ada di *database*. Jika sudah ada registrasi dibatalkan. Namun jika belum ada, sistem menyimpan data baru ke *database*. Setelah berhasil disimpan, sistem menampilkan halaman *Login*, dan proses registrasi selesai.

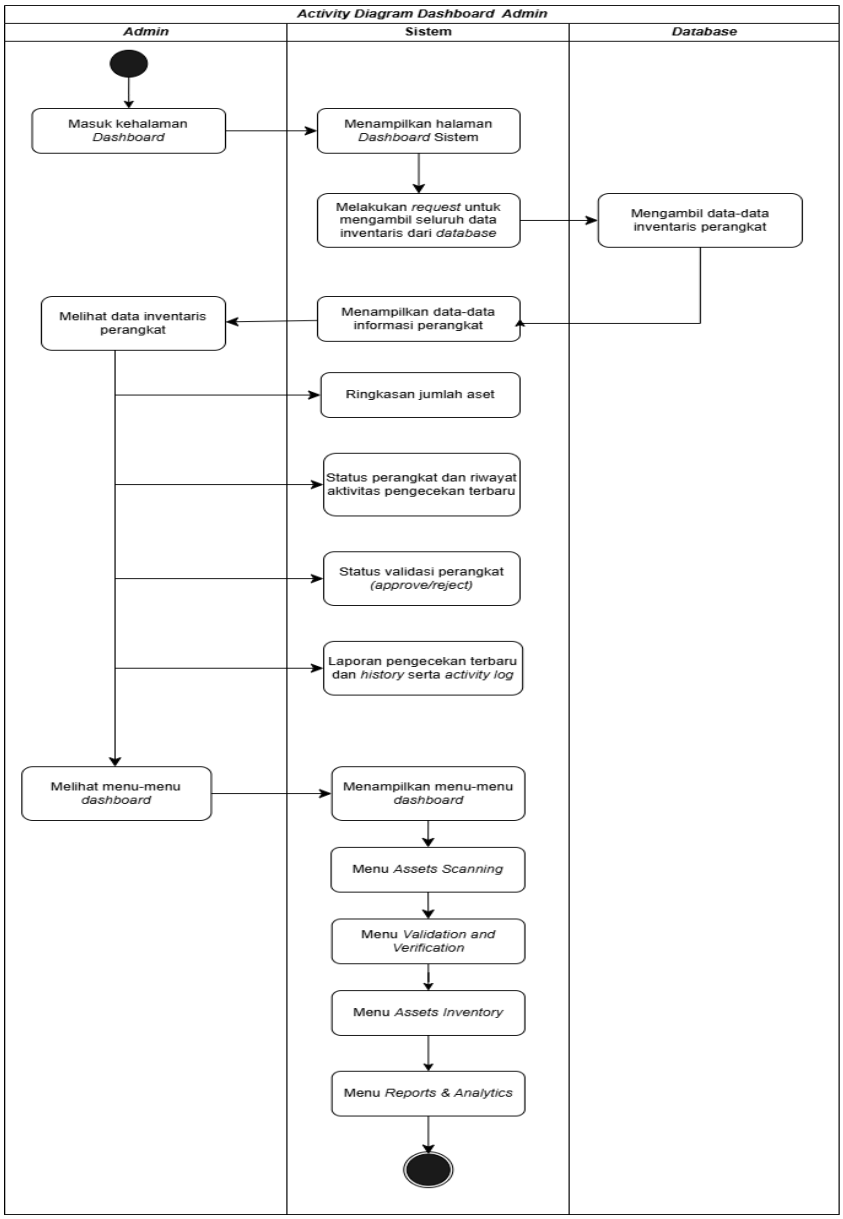
3.2.6.2 *Activity Diagram Login*



Gambar 3. 6 *Activity Diagram Login*

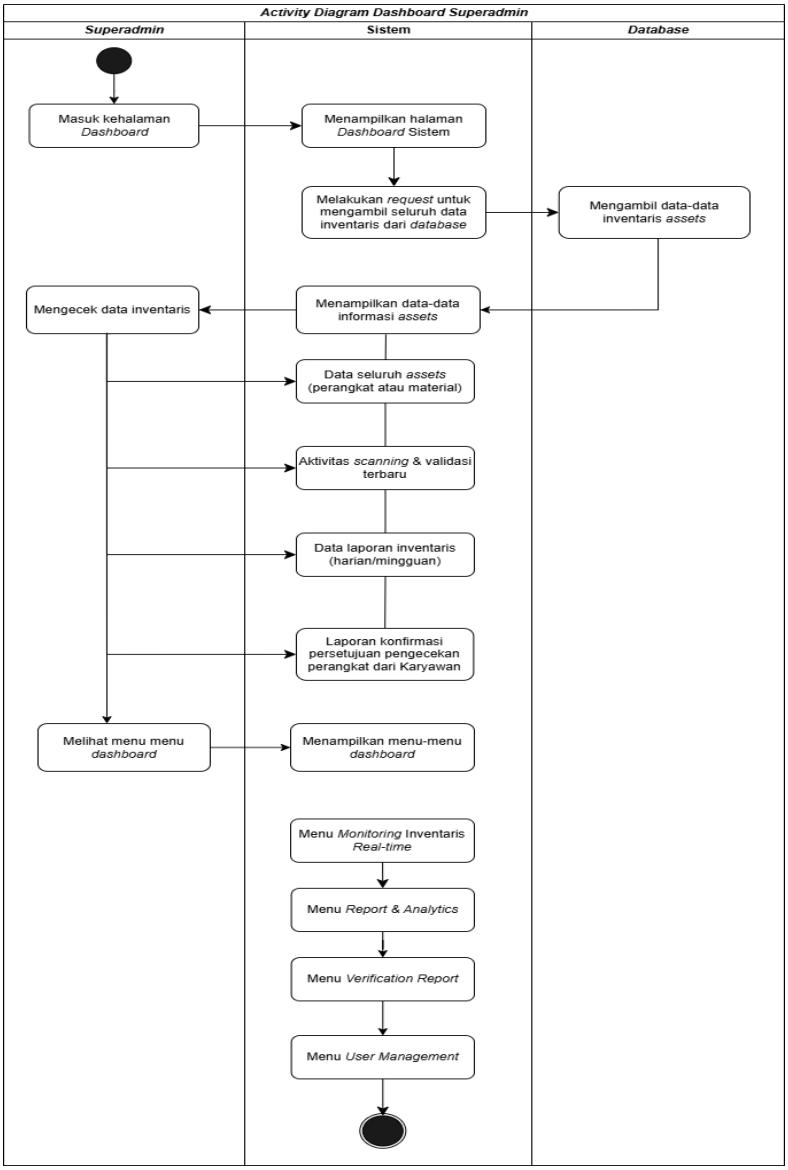
Proses *Activity Diagram Login* pada Gambar 3.6 menggambarkan alur ketika *Admin* atau *Superadmin* ingin masuk ke Sistem Inventaris *IT*. Proses dimulai dengan membuka aplikasi dan masuk ke *landing page*, kemudian menekan tombol *Get Started* atau *Sign In* untuk menuju halaman *login*. Setelah itu, pengguna mengisi form *login* dan mengonfirmasi data yang dimasukkan. Sistem lalu mengirim data tersebut ke *database* untuk divalidasi. Jika data tidak valid (misalnya *usersname* atau *password* salah), muncul notifikasi kesalahan dan pengguna diminta mengisi ulang. Jika data valid, sistem mengizinkan akses dan menampilkan halaman *Dashboard* Sistem Inventaris *IT* sebagai tanda *login* berhasil, dan proses selesai.

3.2.6.3 *Activity Diagram Dashboard Admin* dan *Superadmin*



Gambar 3. 7 *Activity Diagram Dashboard Admin*

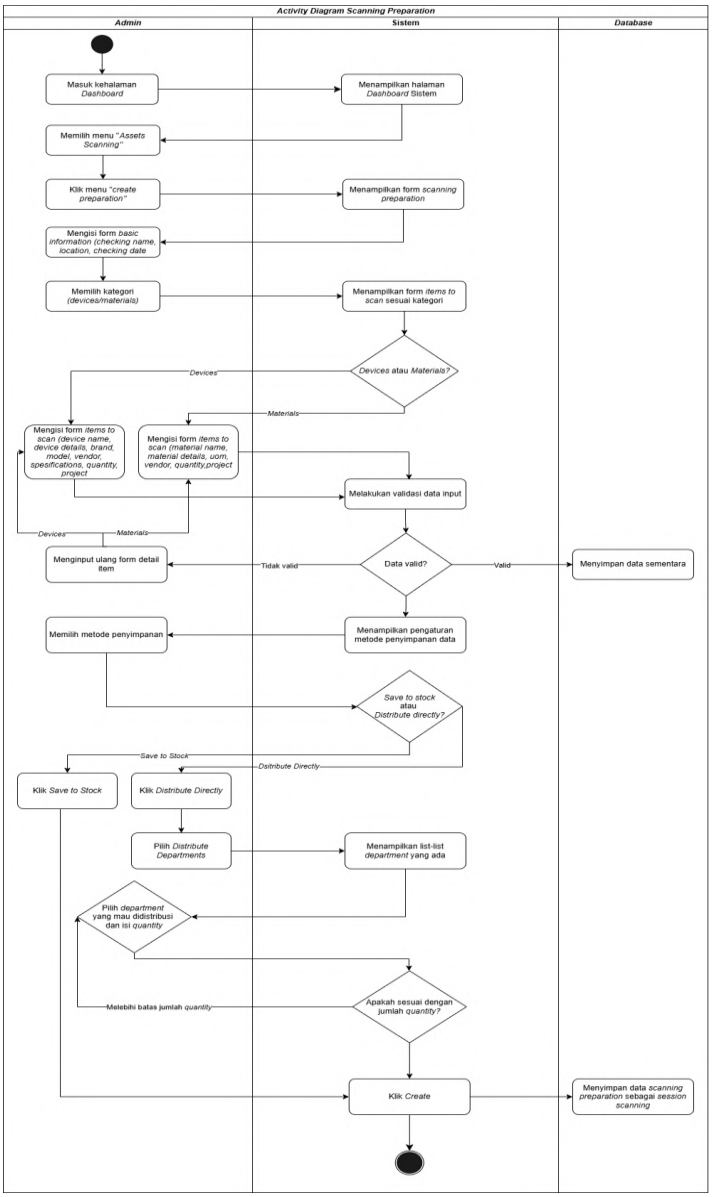
Proses *Activity Diagram Dashboard Admin* pada Gambar 3.7 dimulai ketika *Admin* masuk ke halaman *dashboard*, kemudian sistem mengambil seluruh data inventaris dari *database* dan menampilkan informasi berupa ringkasan jumlah aset, status perangkat, riwayat aktivitas pengecekan, status validasi *(approve/reject)*, serta laporan pengecekan terbaru. Selanjutnya, *Admin* dapat melihat data dan mengakses menu *dashboard* seperti *Assets Scanning*, *Validation and Verification*, *Assets Inventory*, dan *Reports & Analytics* untuk proses pengelolaan inventaris.



Gambar 3. 8 *Activity Diagram Dashboard Superadmin*

Proses *Activity Diagram Dashboard Superadmin* pada Gambar 3.8 dimulai ketika *Superadmin* masuk ke halaman *dashboard*, kemudian sistem mengambil seluruh data inventaris dari *database* dan menampilkan informasi berupa data seluruh aset perangkat atau material, aktivitas *scanning* dan validasi terbaru, laporan inventaris harian atau mingguan, serta laporan konfirmasi persetujuan pengecekan. Selanjutnya, *Superadmin* dapat memantau data tersebut dan mengakses menu *dashboard* seperti *Monitoring* Inventaris *Real-time, Report & Analytics, Verification Report,* dan *Users Management* untuk proses pengawasan dan pengelolaan sistem.

3.2.6.4 *Activity Diagram Scanning Preparation*

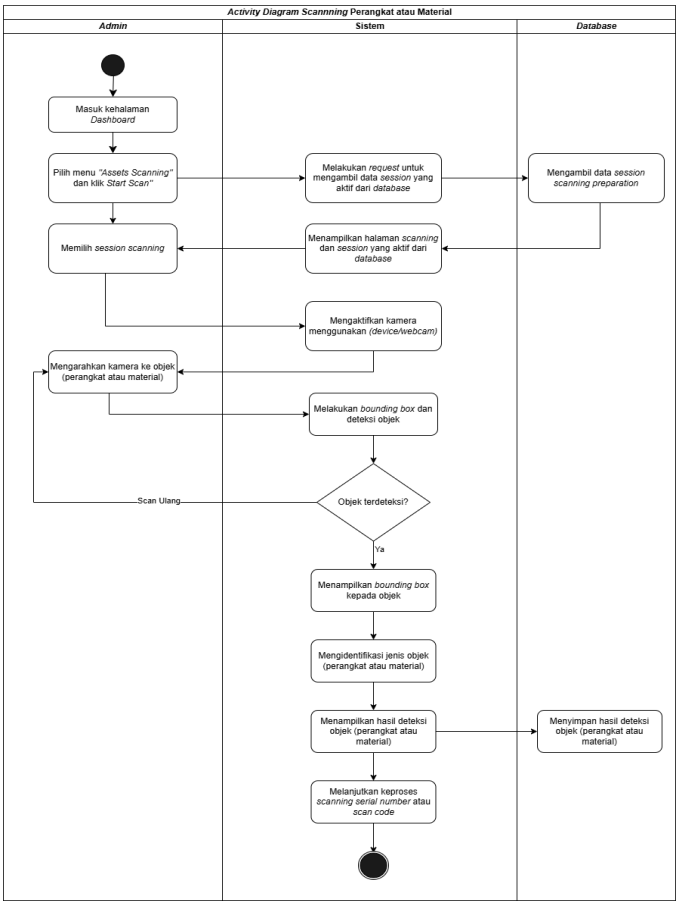


Gambar 3. 9 *Activity Diagram Scanning Preparation*

Proses *Activity Diagram Scanning Preparation* pada Gambar 3.9 dimulai ketika *Admin* masuk ke *dashboard* dan memilih menu *Assets Scanning* dan *create preparation*, kemudian sistem menampilkan form untuk diisi dengan *checking name, location*, dan *checking date*. *Admin* selanjutnya memilih kategori dan mengisi detail item, lalu sistem melakukan validasi data; jika tidak valid diminta perbaikan, dan jika valid data disimpan sementara dan lanjut pengaturan metode penyimpanan. *Admin* memilih metode *save to stock* atau *distribute directly*, mengisi

jumlah sesuai *quantity*, dan menekan *Create*, sistem memvalidasi kembali seluruh data dan menyimpannya ke *database* sebagai *session scanning*, menampilkan notifikasi berhasil, serta mengarahkan ke halaman *scanning*.

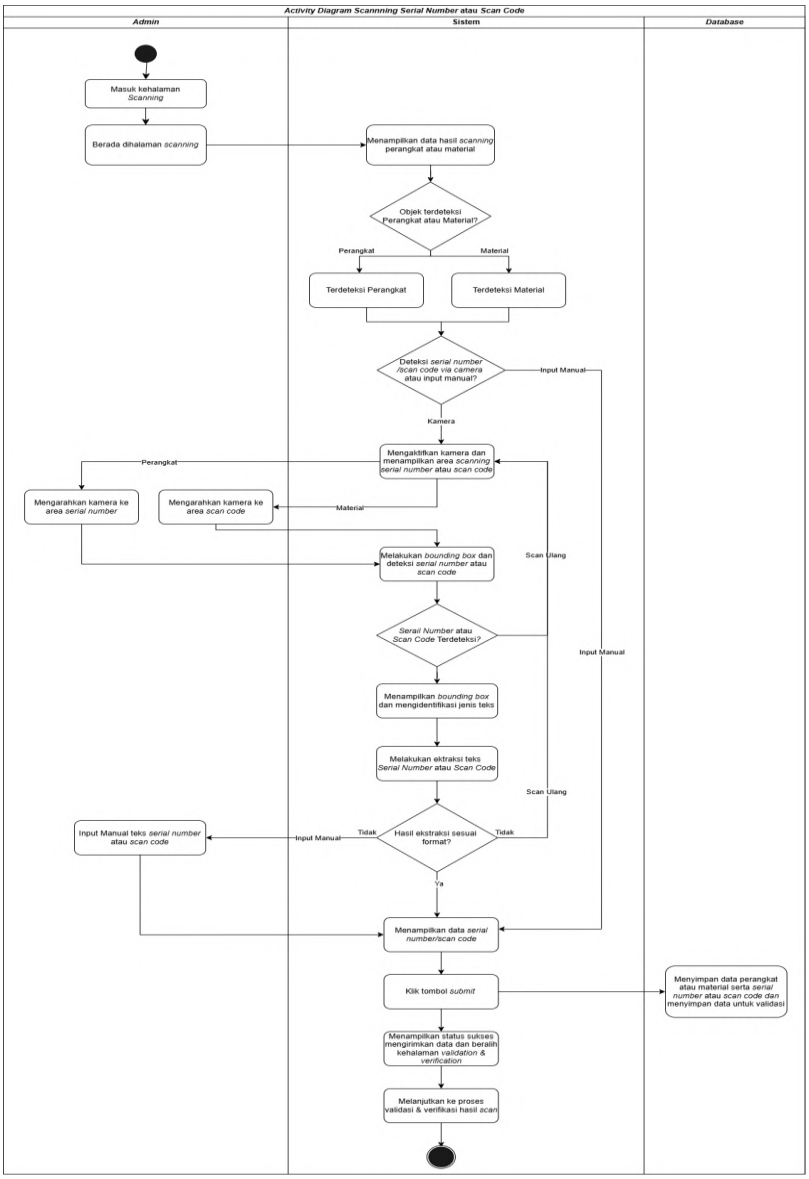
3.2.6.5 *Activity Diagram Scanning* Perangkat atau Material



Gambar 3. 10 *Activity Diagram Scanning* Perangkat atau Material

Proses *Activity Diagram Scanning* Perangkat atau Material pada Gambar 3.10 dimulai ketika *Admin* masuk ke halaman *Dashboard* dan memilih menu *Assets Scanning / Start Scan*, dan sistem menampilkan halaman *scanning* beserta *session* yang aktif. Selanjutnya *Admin* memilih *session* dan sistem mengaktifkan kamera dan kemudian mengarahkan kamera ke objek aset, dan sistem melakukan deteksi objek. Jika objek tidak terdeteksi atau kondisi kamera buram, sistem menampilkan pesan kesalahan dan *Admin* melakukan *scan* ulang. Jika objek berhasil terdeteksi, sistem menampilkan *bounding box* pada objek dan mengidentifikasi jenis objek (perangkat atau material). Hasil deteksi tersebut ditampilkan kepada *Admin*, dan dilanjutkan ke tahap *scanning* detail berupa *serial number* atau *scan code.*

3.2.6.6 *Activity Diagram Scanning Serial Number* atau *Scan code*

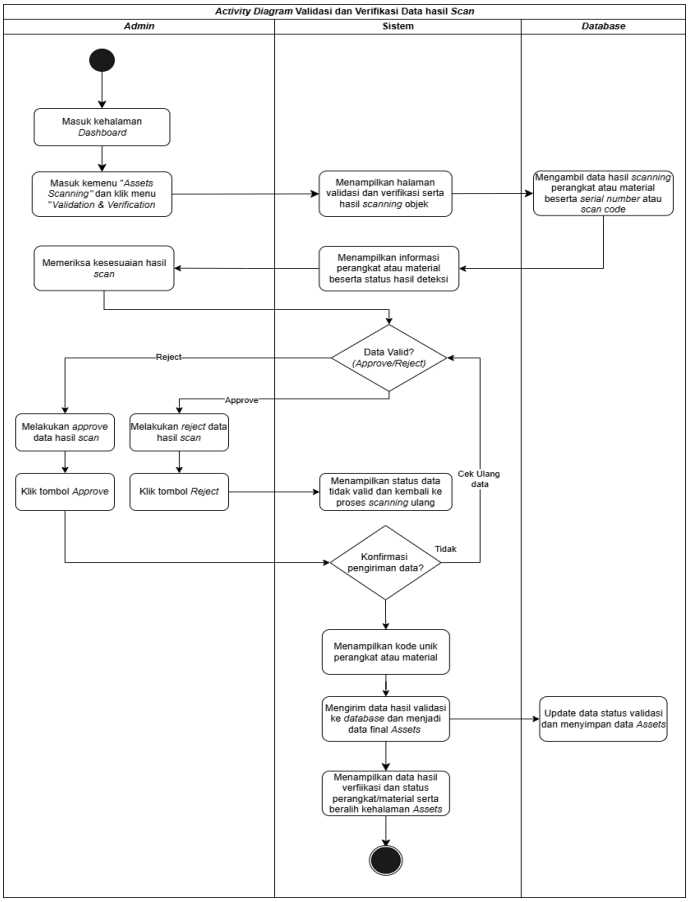


Gambar 3. 11 *Activity Diagram Scanning Serial Number* atau *Scan Code*

Proses *Activity Diagram Scanning Serial Number* atau *Scan Code* pada Gambar 3.11 dimulai ketika *Admin* berada dihalaman *scanning* dan berada pada proses lanjutan setelah deteksi perangkat atau material. Sistem menampilkan hasil *scanning* objek sebelumnya dan identifikasi jenis objek. Selanjutnya sistem mengaktifkan kamera dan *Admin* mengarahkan kamera ke area *serial number* atau *scan code*. Sistem melakukan deteksi area dan menampilkan *bounding box*, lalu mengekstrak teks menggunakan *OCR*. Jika *serial number* atau *scan code* tidak

terdeteksi atau hasil tidak sesuai format, *Admin* dapat melakukan *scan ulang* atau input manual. Jika data berhasil terbaca, sistem menampilkan hasil *serial number* atau *scan code*, kemudian beralih kehalaman validasi dan melanjutkan ke proses validasi dan verifikasi serta menyimpan hasil *scanning* ke *database*.

3.2.6.7 *Activity Diagram* Validasi dan Verifikasi Data hasil *Scan*

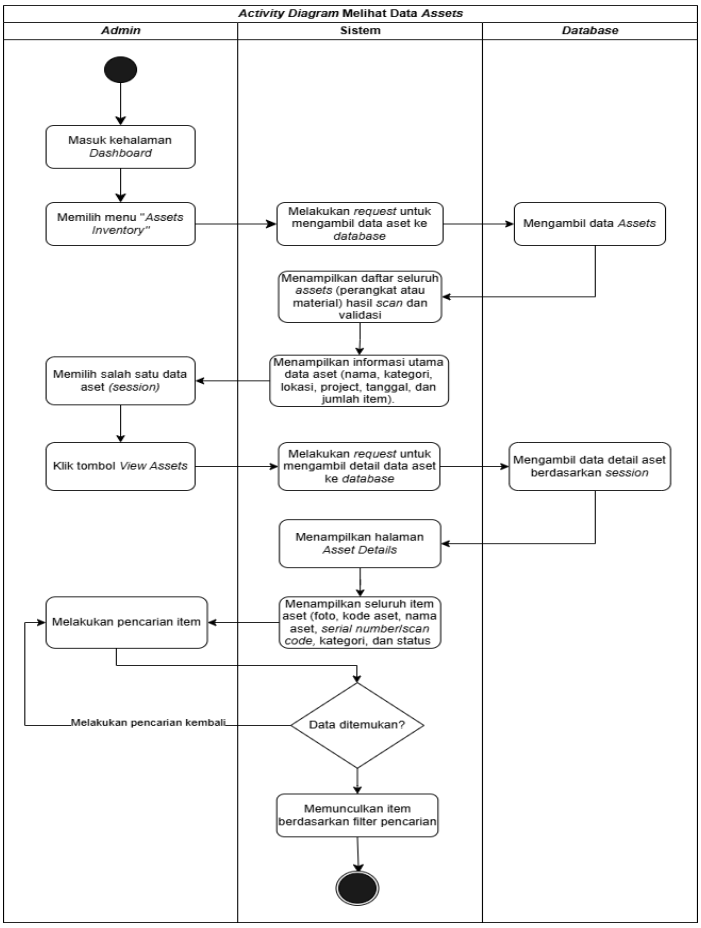


Gambar 3. 12 *Activity Diagram* Validasi dan Verifikasi Data hasil *Scan*

Proses *Activity Diagram* Validasi dan Verifikasi Data Hasil *Scan* pada Gambar 3.12 dimulai ketika *Admin* masuk ke halaman *Dashboard* dan memilih *"Validation & Verification".* Sistem kemudian menampilkan halaman validasi beserta hasil *scanning* perangkat atau material dengan mengambil data dari *database* dan menampilkan informasi serta status hasil deteksi. *Admin* lalu memeriksa kesesuaian data hasil *scan*, apakah data akan di-*approve* atau di-*reject*. Jika data tidak valid *(reject),* sistem menampilkan status tidak valid dan mengarahkan untuk dilakukan pengecekan ulang. Jika data valid *(approve), Admin* melakukan konfirmasi dan sistem menghasilkan kode unik perangkat atau material,

kemudian mengirim serta menyimpan data hasil validasi ke *database* sebagai data final *assets*. Setelah itu, sistem menampilkan hasil akhir verifikasi beserta status perangkat atau material dan mengarahkan ke halaman *Assets Inventory*.

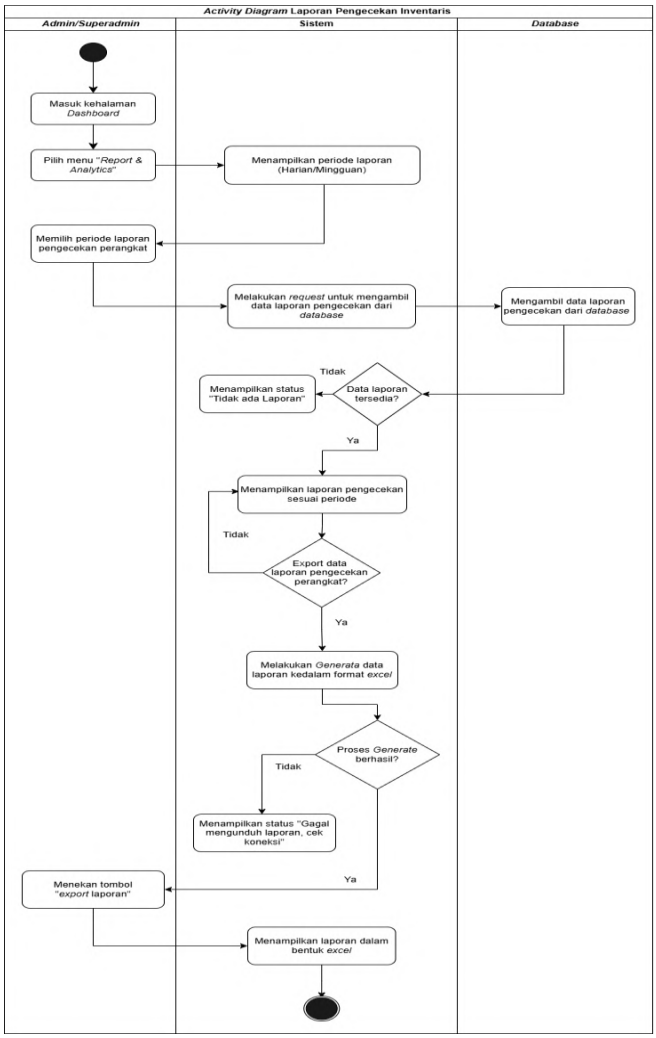
3.2.6.8 *Activity Diagram* Melihat Data *Assets*



Gambar 3. 13 *Activity Diagram* Melihat Data *Assets*

Proses *Activity Diagram* Melihat Data *Assets* pada Gambar 3.13 dimulai ketika *Admin* masuk ke halaman *Dashboard* dan memilih menu *"Assets Inventory"*. Sistem kemudian mengambil data *assets* dari *database* dan menampilkannya dalam bentuk daftar beserta informasi utama aset. *Admin* dapat memilih data *assets (session)* dengan klik *"View Assets"* untuk melihat seluruh item berdasarkan *session* tersebut, dimana sistem akan menampilkan data detail dari *database*. Selanjutnya, *Admin* dapat melakukan pencarian atau filter data *assets*. Jika data tidak ditemukan, sistem menampilkan pesan "Data tidak tersedia", jika tersedia, sistem menampilkan data secara lengkap dan *real-time* hingga proses selesai.

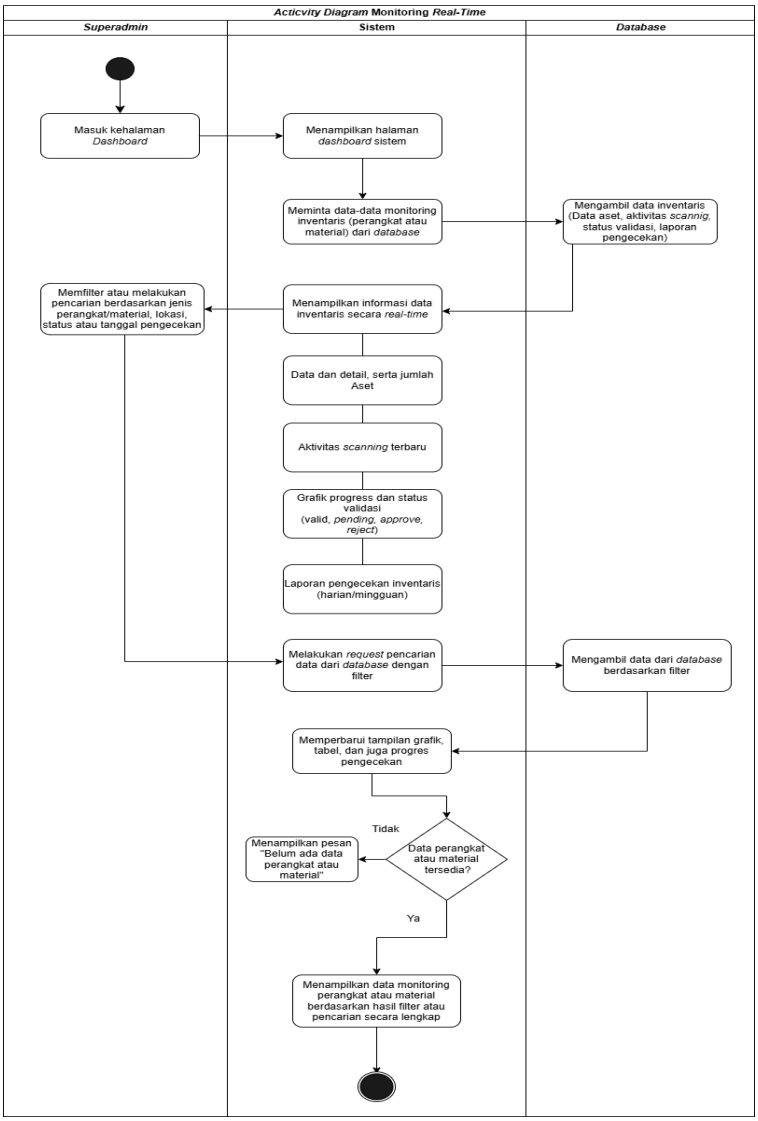
3.2.6.9 *Activity Diagram* Laporan Pengecekan Inventaris



Gambar 3. 14 *Activity Diagram* Laporan Pengecekan Inventaris

Proses *Activity Diagram* Laporan Pengecekan Inventaris Harian/Mingguan pada Gambar 3.14 dimulai dengan masuk menu "*Report & Analytics*", kemudian sistem menampilkan pilihan periode laporan (harian/mingguan). Pengguna memilih periode, dan sistem mengambil data laporan pengecekan perangkat atau material. Jika data tidak tersedia, muncul status "Tidak ada laporan". Jika data tersedia, sistem menampilkan laporan pengecekan sesuai periode dan jika tidak diekspor, proses selesai. Jika ya, sistem meng-*generate* laporan ke format *excel*. Jika terjadi kegagalan koneksi sistem menampilkan pesan "Gagal mengunduh laporan, cek koneksi!". Jika berhasil, maka akan menampilkan laporan dalam bentuk *excel*.

3.2.6.10 *Activity Diagram Monitoring real-time*

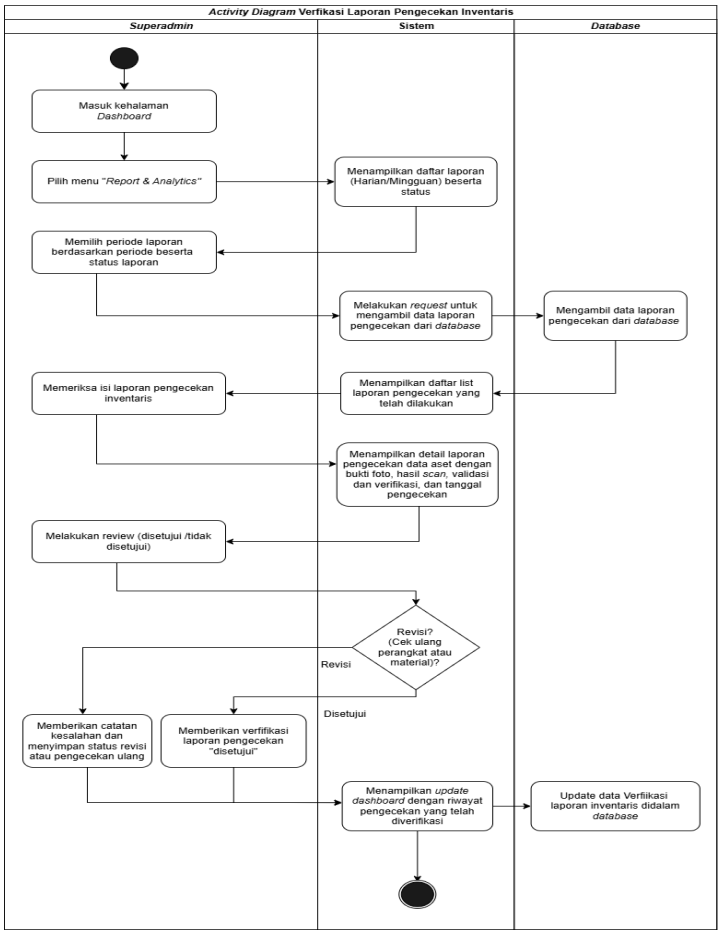


Gambar 3. 15 *Activity Diagram Monitoring Real-time*

Proses *Activity Diagram Monitoring Real-Time* pada Gambar 3.15 dimulai *Superadmin* masuk ke halaman *Dashboard*, kemudian sistem menampilkan halaman dan mengambil data monitoring inventaris dari *database* berupa data aset, status validasi, aktivitas *scanning*, serta riwayat pengecekan. Sistem menampilkan informasi monitoring secara *real-time* seperti jumlah aset, status validasi *(approve/reject)*, aktivitas *scanning* terbaru, grafik monitoring, dan riwayat pengecekan. *Superadmin* dapat melakukan pencarian berdasarkan periode, kategori, atau status, lalu sistem mengirim permintaan ke *database* dan menampilkan hasil sesuai filter. Jika tidak ditemukan, sistem menampilkan pesan

"Data tidak sesuai kriteria", jika belum tersedia menampilkan "Belum ada data perangkat atau material", dan jika tersedia sistem menampilkan data secara lengkap serta memperbaruinya secara *real-time* hingga proses selesai.

3.2.6.11 *Activity Diagram* Verifikasi Laporan Pengecekan

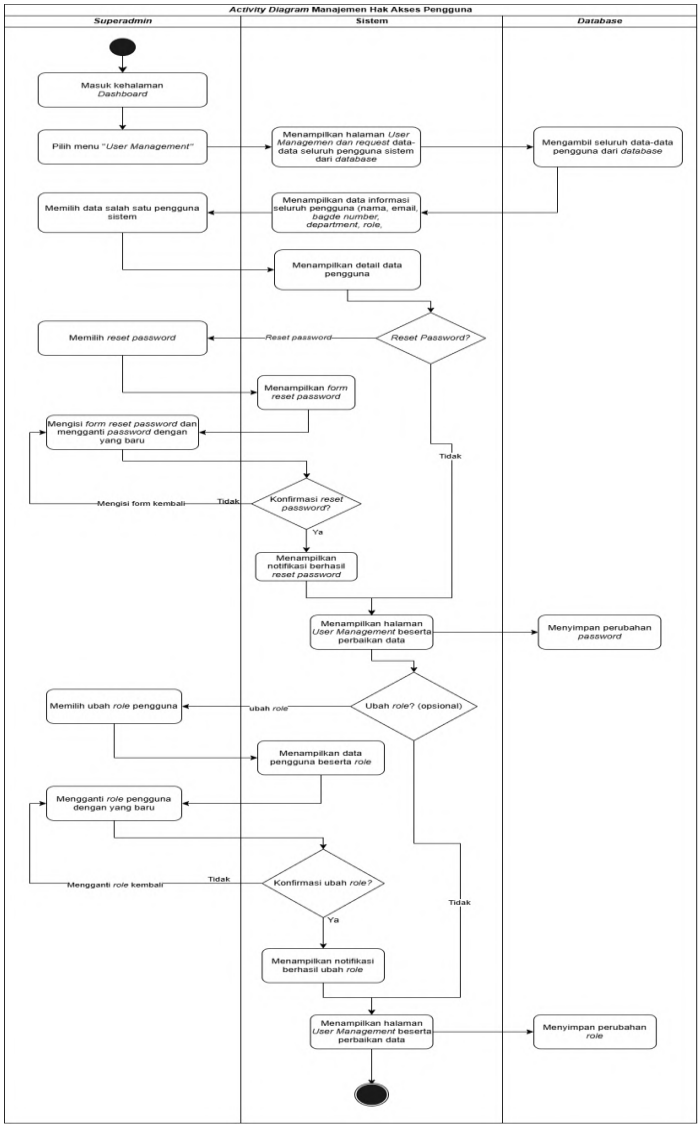


Gambar 3. 16 *Activity Diagram* Verifikasi Laporan Pengecekan

Proses *Activity Diagram* Verifikasi Laporan Pengecekan pada Gambar 3.16 dimulai saat *Superadmin* masuk ke halaman *Dashboard* dan memilih menu *"Report & Analytics",* kemudian memilih periode dan status laporan. Sistem menampilkan daftar laporan dari *database*, lalu *Superadmin* memilih laporan untuk melihat detail berupa data aset, bukti foto, hasil *scan*, validasi, dan tanggal pengecekan. Selanjutnya *Superadmin* melakukan *review* apakah laporan disetujui atau direvisi. Jika revisi, maka diberikan catatan dan sistem menyimpan status revisi atau pengecekan ulang ke *database*, jika disetujui, sistem menyimpan hasil verifikasi

dan menampilkan pembaruan pada *dashboard* berupa riwayat laporan yang telah diverifikasi hingga proses selesai.

3.2.6.12 *Activity Diagram* Manajemen Hak Akses Pengguna

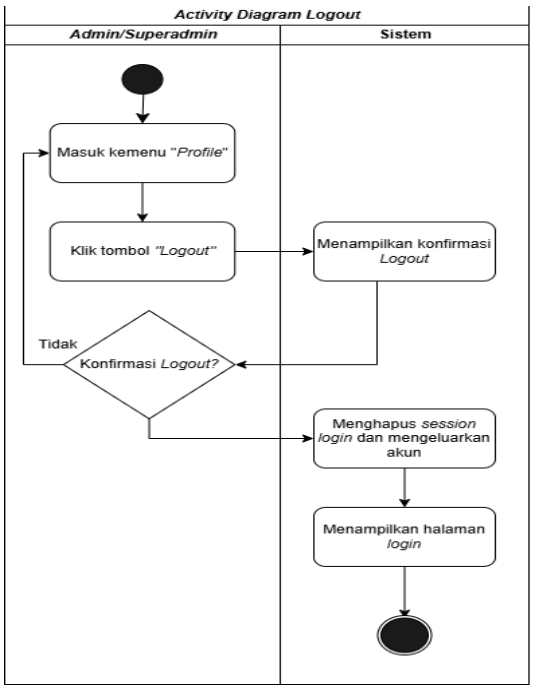


Gambar 3. 17 *Activity Diagram* Manajemen Hak Akses Pengguna

Proses *Activity Diagram* Manajemen Hak Akses Pengguna pada Gambar 3.17 dimulai ketika *Superadmin* masuk ke halaman *dashboard* dan memilih menu *"Users Management"*, sistem menampilkan halaman manajemen pengguna dengan mengambil data pengguna dari *database* dan menampilkannya dalam bentuk daftar. Selanjutnya *superadmin* memilih salah satu pengguna untuk melihat detail data, lalu menentukan apakah reset *password* atau perubahan *role*. Jika reset password

dipilih, sistem akan menampilkan *form* atau melakukan reset otomatis, kemudian setelah dikonfirmasi sistem menyimpan perubahan ke *database* dan menampilkan notifikasi berhasil. Jika perubahan *role* dilakukan, sistem akan memperbarui data *role* ke *database* dan menampilkan notifikasi berhasil. Setelah itu, sistem menampilkan kembali daftar pengguna yang telah diperbarui dan proses selesai.

3.2.6.13 *Activity Diagram Logout*

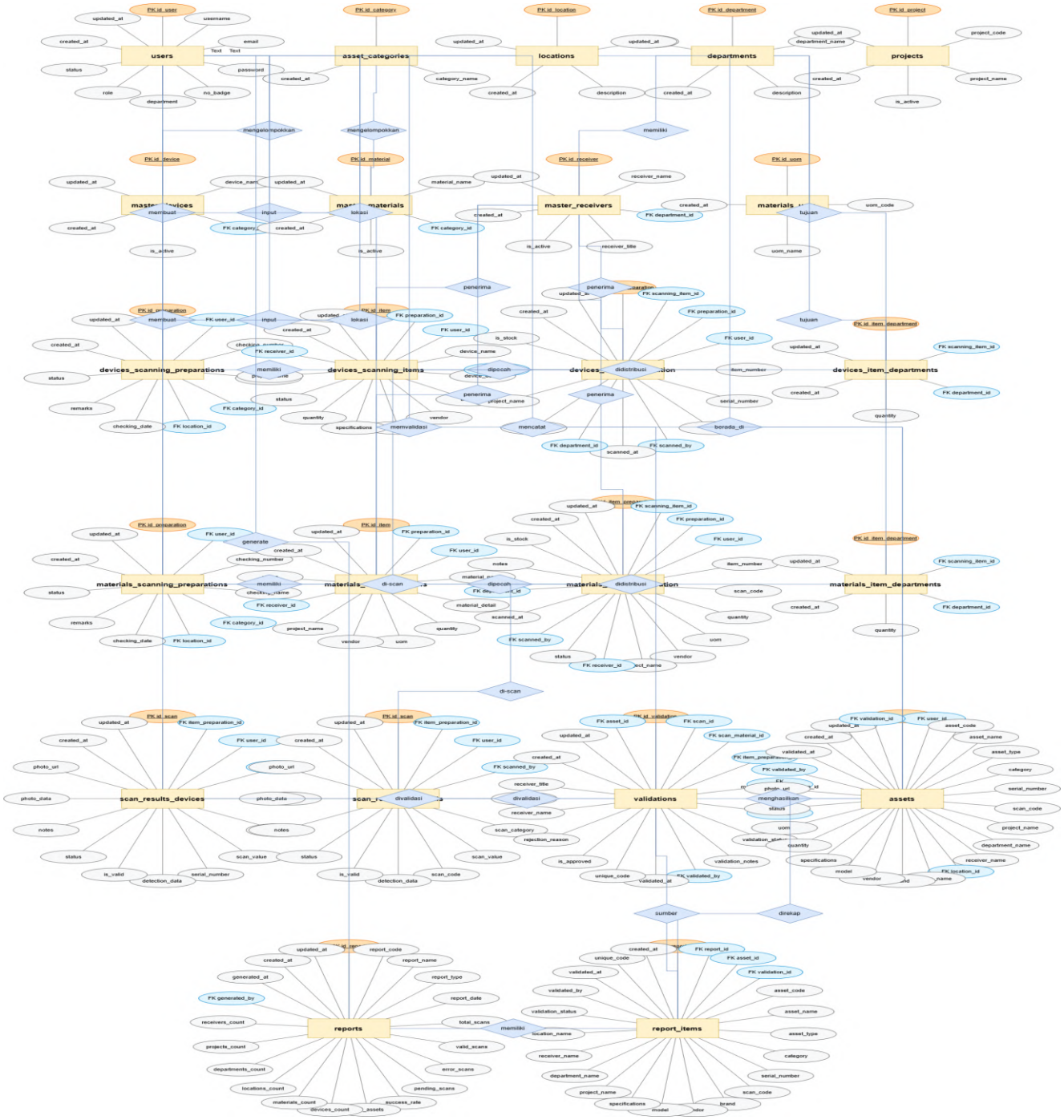


Gambar 3. 18 *Activity Diagram Logout*

Proses *Activity Diagram Logout* pada Gambar 3.18 ini dimulai dengan pengguna masuk ke menu *"Profile"* dan menekan tombol *"Logout",* kemudian sistem menampilkan dialog konfirmasi untuk memastikan keputusan pengguna. Jika pengguna memilih "Tidak", sistem menutup konfirmasi sesi tetap aktif dan pengguna tetap berada di aplikasi. Namun jika pengguna menekan "Ya", sistem akan menghapus *session login*, mengakhiri akses pengguna, dan mengalihkan ke halaman *login*. Proses ini memastikan keamanan sistem dan mencegah akses tanpa autentikasi.

3.2.7 ER Diagram

ER Diagram Sistem Inventaris *IT* pada gambar 3.19 menggambarkan alur proses dari entitas yang saling terhubung untuk mengelola data pengguna, aset, proses pengecekan, serta pelaporan.



Gambar 3. 19 *ER Diagram* Sistem Inventaris *IT*

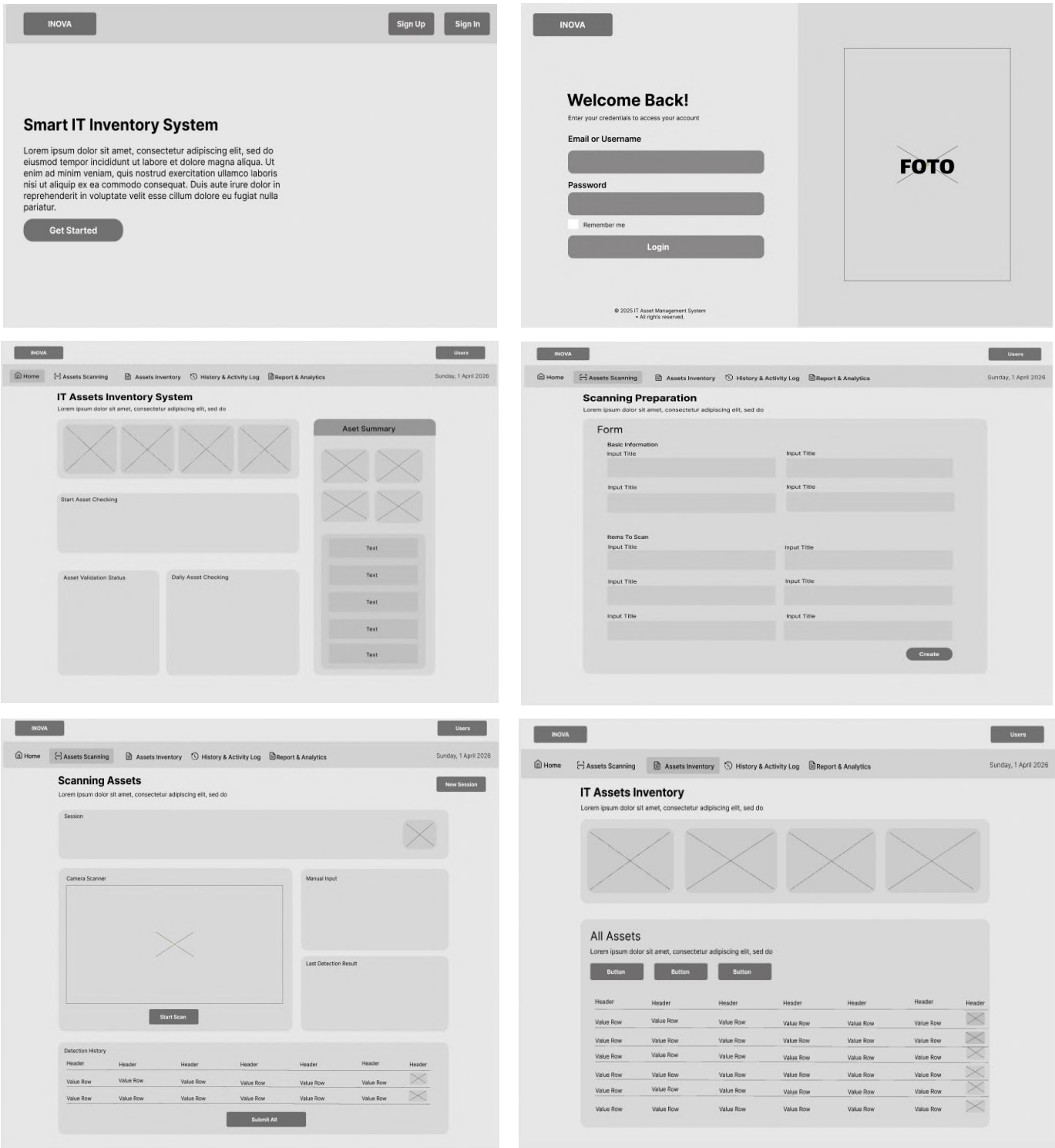
Pada *ERD* sistem inventaris *IT*, entitas utama yang menjadi pusat relasi adalah *users* yang menyimpan data pengguna seperti *Admin* dan *Superadmin*, lengkap dengan atribut seperti *username*, *email*, *role*, dan *department*. Entitas ini memiliki relasi dengan berbagai entitas lain melalui *foreign key*, khususnya dalam aktivitas pencatatan data seperti *scanning* dan validasi. Selanjutnya terdapat entitas *asset_categories* yang berfungsi untuk mengelompokkan jenis aset menjadi perangkat dan material, yang kemudian terhubung dengan entitas *master_devices* dan *master_materials* sebagai referensi data utama. Selain itu, entitas *locations*, *departments*, dan *projects* berperan sebagai entitas pendukung yang menyimpan informasi kontekstual terkait lokasi aset, unit organisasi, dan proyek, sehingga setiap data aset memiliki keterkaitan yang jelas.

ERD juga memiliki entitas *devices_scanning_preparations* dan *materials_scanning_preparations* yang menyimpan informasi sesi pengecekan, seperti tanggal, lokasi, dan *user* yang melakukan proses tersebut. Kedua entitas ini memiliki relasi dengan entitas detail yaitu *devices_scanning_items* dan *materials_scanning_items* yang berisi daftar item yang akan dilakukan pengecekan. Selanjutnya terdapat entitas *devices* dan *materials* yang menyimpan data hasil distribusi atau penyimpanan aset setelah proses *scanning* selesai. Untuk mencatat hasil pemindaian, digunakan entitas *scan_results_devices* dan *scan_results* yang menyimpan data seperti *serial number*, *scan code*, foto bukti, serta informasi waktu dan *user* yang melakukan *scanning*. Relasi antar entitas ini menunjukkan bahwa setiap hasil *scan* terhubung langsung dengan item, *preparation*, serta *user* yang melakukan aktivitas tersebut.

Pada tahap validasi dan penyimpanan akhir, entitas *validations* berfungsi menyimpan status hasil *scanning* seperti *approve* atau *reject*, serta kode unik aset. Entitas ini berelasi dengan *assets* sebagai data inventaris akhir yang memuat informasi lengkap seperti *asset_code*, nama aset, *serial number* atau *scan code*, *category*, *location*, dan *departement*. Selain itu, entitas *assets* juga terhubung dengan *reports* sebagai laporan utama dan *report_items* sebagai detail aset dalam laporan, dengan relasi *one-to-many*. Struktur ini menunjukkan keterkaitan data yang terintegrasi dari hasil validasi hingga pelaporan inventaris secara konsisten.

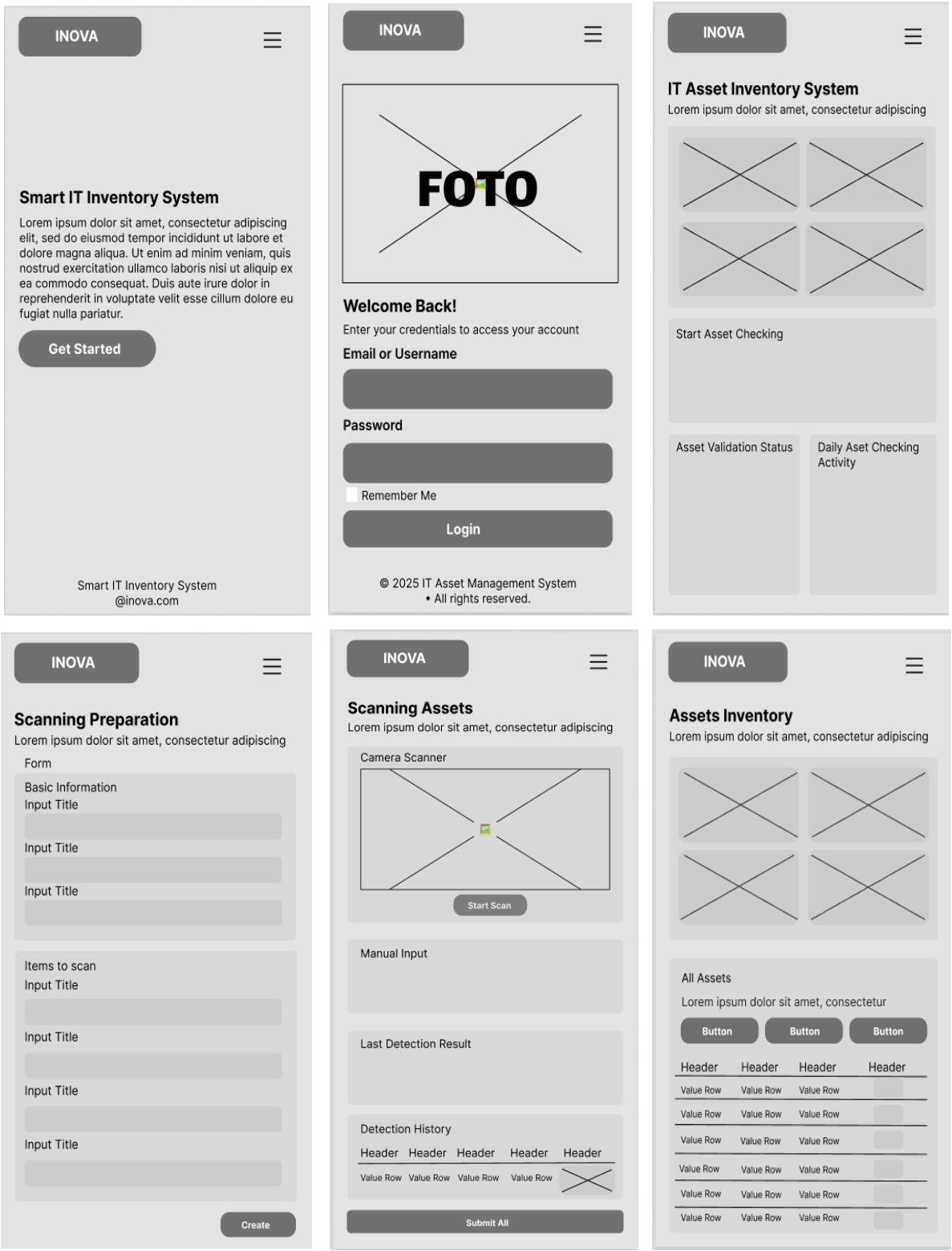
3.2.8 Perancangan Antarmuka (*Wireframe*)

1. *Web*



Gambar 3. 20 *Wireframe Web*

2. *Mobile*



Gambar 3. 21 *Wireframe Mobile*

BAB IV
IMPLEMENTASI DAN PENGUJIAN

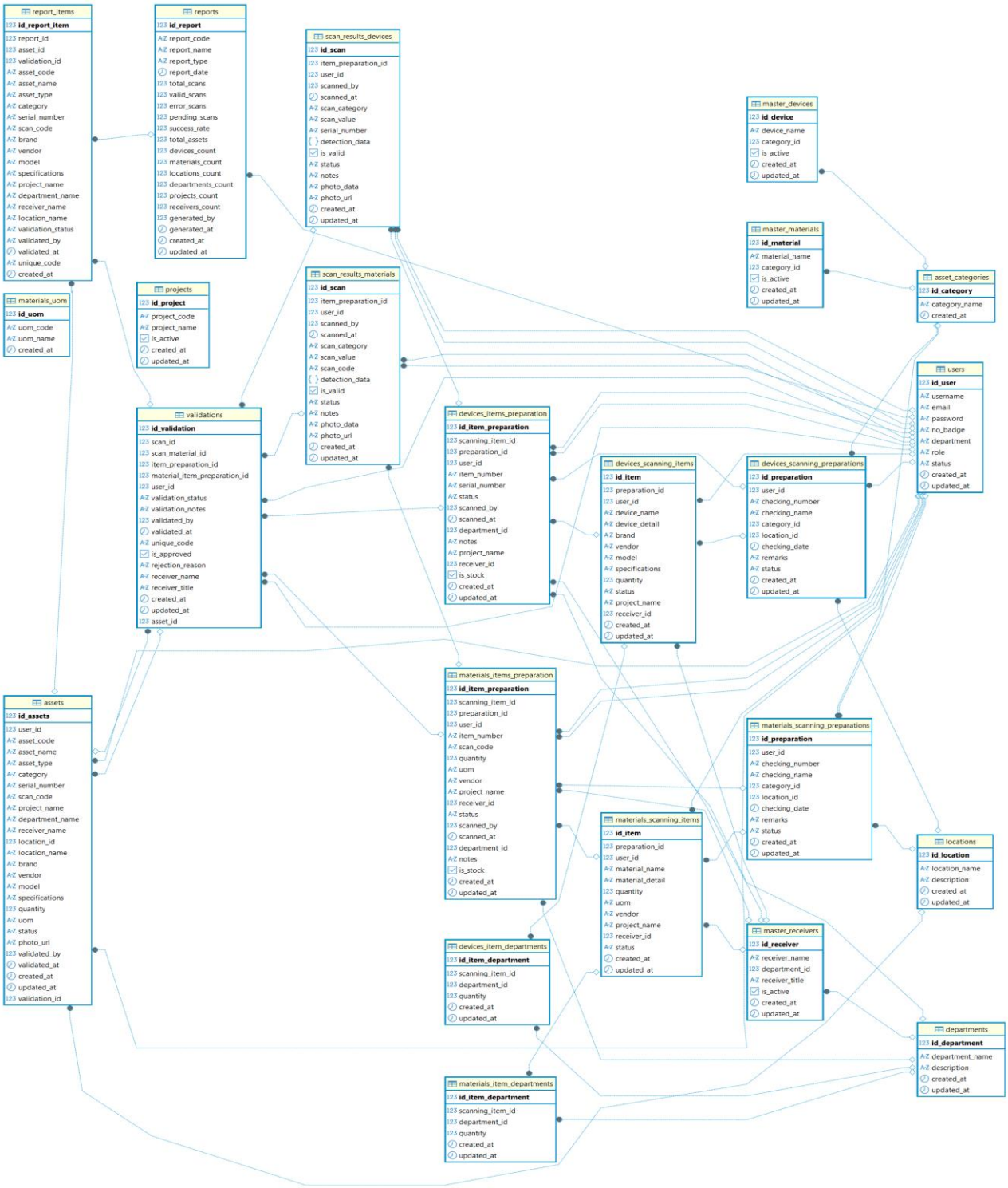
4.1 Hasil Implementasi

Implementasi pada proyek akhir ini dilakukan setelah tahap analisis kebutuhan dan perancangan sistem selesai dilaksanakan sesuai dengan metode perancangan yang digunakan. Tahap implementasi diawali dengan pembangunan basis data menggunakan *PostgreSQL* sebagai fondasi utama sistem untuk menyimpan data pengguna, data aset, hasil *scanning*, validasi data, serta laporan pengecekan inventaris. Struktur basis data dirancang berdasarkan *ERD* sehingga relasi antar data terorganisir dengan baik dan mendukung kebutuhan sistem.

Tahap implementasi selanjutnya dilakukan pada sisi *backend* menggunakan *framework Flask* untuk membangun *API* yang berfungsi sebagai penghubung antara *frontend*, basis data, serta proses *AI*. *API* yang dikembangkan mencakup autentikasi pengguna, pengelolaan data aset, proses *scanning*, validasi data, serta laporan pengecekan inventaris. Pada tahap ini juga dilakukan pengembangan model *AI* melalui pelabelan *dataset* untuk model *devices, materials, serial number,* dan *scan code* menggunakan *Roboflow*, serta *training* model *YOLO* di *Google Colab*. Model kemudian diintegrasikan ke dalam *Flask* dan *OCR* untuk ekstraksi teks dan memproses hasil *scanning* secara otomatis. Pengujian *API* dilakukan menggunakan *Postman* agar setiap *endpoint* berjalan dengan baik dan sesuai kebutuhan sistem.

Pada sisi *frontend*, implementasi menggunakan *Next.js* untuk membangun antarmuka *web* yang responsif dan mendukung perangkat *mobile*. Fitur utama seperti *dashboard*, *scanning* perangkat atau material, validasi hasil *scan*, data *assets*, monitoring *real-time*, serta laporan inventaris dikembangkan berdasarkan response *API* dari *backend*. Setelah itu, dilakukan *deployment* ke *server* internal berbasis *Ubuntu* dengan konfigurasi *PostgreSQL* dan *Apache*. Pengaturan *port* meliputi *frontend* pada *port* 443 *(HTTP)* dan 3004 (*HTTPS)*, serta *backend* pada port 80 *(HTTP)* dan 5001 *(HTTPS)* untuk menghindari permasalahan *mixed content*. Proses *remote server* dilakukan menggunakan *Remote SSH* untuk mempermudah konfigurasi dan pemeliharaan sistem.

4.1.1 Implementasi Basis Data



Gambar 4. 1 Implementasi Basis Data

4.1.2 Implementasi *API Backend*

Pada tahap implementasi *API backend*, sistem menggunakan arsitektur *REST API* berbasis *framework Flask* sebagai penghubung antara antarmuka *frontend*, basis data *PostgreSQL*, serta proses *AI* yang mencakup deteksi objek menggunakan *YOLO* dan ekstraksi teks menggunakan *OCR*. Setiap respons API disusun dalam format *JSON* yang memuat informasi status, *message*, dan data untuk mempermudah proses integrasi dengan *frontend* berbasis *Next.js*. Dalam pengembangan *API*, digunakan metode *HTTP* seperti *GET* untuk pengambilan data, *POST* untuk proses autentikasi dan penambahan data, *PUT* atau *PATCH* untuk pembaruan data, serta *DELETE* untuk penghapusan data sesuai kebutuhan fungsional sistem. *Endpoint* yang dikembangkan mencakup autentikasi pengguna, pengelolaan data aset, proses *scanning* dan validasi, serta laporan inventaris, sehingga sistem dapat berjalan secara terintegrasi dan *real-time*.

Tabel 4. 1 Implementasi *API*

No	*Endpoint*	*Methods*	*Response*

1.	*/api/register*	*POST*	{
			"message": "Registration successful",
			"success": true,
			"user": {
			"department": "IT",
			"email": "user_postman_01@gmail.com",
			"id": 2,
			"no_badge": "9090100",
			"status": "active",
			"username": "user_postman_01"
			}
			}

2.	*/api/login*	*POST*	{
			"message": "Login successful",
			"success": true,
			"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
			eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InVzZXJfcG9zdG1hbl8wMSIsImV4cCI6MTc3NjQ3NTI1NH0.
			ugSjMDR3IVqvCF8TMCwNaa1U4JfQAj-tzGNN0zHSPWI",
			"user": {
			"department": "IT",
			"email": "user_postman_01@gmail.com",
			"id": 2,
			"name": "user_postman_01",
			"no_badge": "9090100",
			"role": "karyawan",
			"status": "active",
			"username": "user_postman_01"
			}
			}

3.	*/api/update-profile*	*PUT*	{
			"message": "Profile updated successfully",
			"success": true,
			"user": {
			"created_at": "2026-04-17T08:10:59.282266",
			"department": "IT Support",
			"email": "user_postman_01_update@gmail.com",
			"id": 2,
			"no_badge": "9090100",
			"status": "active",
			"username": "user_postman_01_update"
			}
			}

4.	*/api/change-password*	*POST*	{
			"message": "Password changed successfully",
			"success": true
			}

5.	*/api/health*	*GET*	{
			"service": "auth-api",
			"status": "healthy",
			"timestamp": "2026-04-17T08:37:51.199985"
			}

6.	*/api/detect/camera*	*POST*	{
			"detected_items": [
			{
			"asset_type": "Laptop",
			"bbox": [
			279.8353271484375,
			7.1160888671875,
			1163.1285400390625,
			720.0
			],
			"brand": "N/A",
			"category": "Perangkat",
			"confidence": 0.965,
			"id": "LAP-392d690b-001",
			"location": "",
			"needs_serial_scan": true,
			"serial_number": "",
			"status": "device_detected",
			"timestamp": "20260417_084539"
			}
			],
			"message": "Berhasil mendeteksi 1 perangkat",
			"original_image_path": "temp\\camera_1776390339.jpg",
			"original_image_url": "/uploads/camera_1776390339.jpg",
			"result_image_path": "static/results\\device_20260417_084539_392d690b\\camera_1776390339.jpg",
			"result_image_url": "/detection_results/camera_1776390339.jpg",
			"success": true,
			"total_detected": 1
			}

7.	*/api/detect/camera/simple*	*POST*	{
			"detected_items": [
			{
			"asset_type": "Laptop",
			"category": "Perangkat",
			"confidence": 0.965
			}
			],
			"message": "Berhasil mendeteksi 1 perangkat",
			"success": true,
			"total_detected": 1
			}

8.	*/api/detect/material/camera*	*POST*	{
			"detected_items": [
			{
			"asset_type": "Cable LAN",
			"bbox": [
			24.32452392578125,
			186.5531005859375,
			707.9798583984375,
			837.3726806640625
			],
			"brand": "",
			"category": "Material",
			"confidence": 0.952,
			"id": "MAT-d79cec8d-001",
			"location": "",
			"needs_scan_code": true,
			"scan_code": "",
			"status": "material_detected",
			"timestamp": "20260417_090906",
			"uom": "PCS",
			"vendor": ""
			}
			],
			"message": "Berhasil mendeteksi 1 material",
			"original_image_path": "temp\\material_camera_1776391746.jpg",
			"result_image_path": "static/results\\material_20260417_090906_d79cec8d\\material_camera_1776391746.jpg",
			"success": true,
			"total_detected": 1
			}

9.	*/api/serial/detect/ camera*	*POST*	{
			"best_serial": {
			"bbox": [
			448.3153381347656,
			102.87403869628906,
			1189.428466796875,
			250.56948852539062
			],
			"confidence": 0.787,
			"detected_text": "SN2400205577883",
			"is_valid": true,
			"type": "SN"
			},
			"message": "Found 2 serial numbers",
			"result_image_path": "static/results\\serial_20260417_091508_c3db5e45\\serial_camera_1776392108.jpg"
			"serial_detections": [
			{
			"bbox": [
			448.3153381347656,
			102.87403869628906,
			1189.428466796875,
			250.56948852539062
			],
			"confidence": 0.787,
			"detected_text": "SN2400205577883",
			"is_valid": true,
			"type": "SN"
			},
			{
			"bbox": [
			453.3816223144531,
			101.23878479003906,
			1201.3250732421875,
			254.69451904296875
			],
			"confidence": 0.779,
			"detected_text": "SN2400205577883",
			"is_valid": true,
			"type": "Serial Number"
			}
			],
			"success": true,
			"total_detected": 2,
			"valid_serials": 2
			}

10.	*/api/serial/assign/ <device_id>*	*POST*	{
			"device_id": "DEV001",
			"message": "Serial number assigned to device DEV001",
			"serial_number": "SN12345678",
			"success": true
			}

11.	*/api/scan-code/detect/camera*	*POST*	{
			"best_detection": {
			"bbox": [
			68.63226318359375,
			529.8251953125,
			620.7869873046875,
			618.7633056640625
			],
			"confidence": 0.829,
			"detected_text": "IT-MT-CL-2603-0001",
			"is_valid": true,
			"matched_material": "Cable LAN",
			"type": "Scan Code",
			"validation_message": "Valid Cable LAN scan code"
			},
			"message": "Found 1 scan code(s)",
			"result_image_path": "static/results\\scancode_20260417_093456_9b650414\\scancode_camera_1776393296.jpg",
			"scan_code_detections": [
			{
			"bbox": [
			68.63226318359375,
			529.8251953125,
			620.7869873046875,
			618.7633056640625
			],
			"confidence": 0.829,
			"detected_text": "IT-MT-CL-2603-0001",
			"is_valid": true,
			"matched_material": "Cable LAN",
			"type": "Scan Code",
			"validation_message": "Valid Cable LAN scan code"
			}
			],
			"success": true,
			"total_detected": 1,
			"valid_detections": 1
			}

12.	*/api/scan-code/validate*	*POST*	{
			"is_valid": true,
			"message": "Valid scan code",
			"scan_code": "IT-MT-RJ-0001",
			"success": true
			}

13.	*/api/ocr/extract-serial*	*POST*	{
			"extracted_serial": "SN24000205577883",
			"is_valid": true,
			"success": true
			}

14.	*/api/ocr/validate-serial*	*POST*	{
			"original_text": "SN-ABC123456",
			"success": true,
			"validated_serial": "SNABC123456"
			}

15.	*/api/location/all*	*GET*	{ "locations": [{ "area": "General", "category": "Office", "description": "Facility", "id_location": 14, "location_code": "LOC-014", "location_name": "Clinic" }, { "area": "General", "category": "Office", "description": "Office", "id_location": 4, "location_code": "LOC-004", "location_name": "Engineering Office Lv1" }, { "area": "General", "category": "Office", "description": "Office", "id_location": 5, "location_code": "LOC-005", "location_name": "Engineering Office Lv2" }, { "area": "General", "category": "Office", "description": "Office", "id_location": 6, "location_code": "LOC-006", "location_name": "Engineering Office Lv3" }, { "area": "General", "category": "Office", "description": "Office",

16.	*/api/location/search?q=...*	*GET*	{
			"locations": [],
			"success": true,
			"total": 0
			}

17.	*/api/location/assign-multiple*	*POST*	{
			"failed_assets": [],
			"failed_count": 0,
			"location_code": "LOC-001",
			"location_name": "Main Office Lv1",
			"message": "Successfully assigned 1 assets",
			"success": true,
			"success_count": 1
			}

18.	*/api/location/asset/<asset_id>*	*GET*	{
			"location": {
			"asset_code": "AST-20260417-J4MC9P",
			"assigned_at": "Fri, 17 Apr 2026 10:16:35 GMT",
			"location_id": 1,
			"location_name": "Main Office Lv1"
			},
			"success": true
			}

19.	*/api/location/<location_code>*	*GET*	{
			"location": {
			"area": "General",
			"category": "Office",
			"description": "Office",
			"id_location": 1,
			"location_code": "LOC-001",
			"location_name": "Main Office Lv1"
			},
			"success": true
			}

20.	*/api/department/all*	*GET*	{ "departments": [{ "department_code": "DEPT-024", "department_name": "Blasting & Painting", "description": "Blasting & Painting Department", "id_department": 24 }, { "department_code": "DEPT-026", "department_name": "Canteen", "description": "Canteen Department", "id_department": 26 }, { "department_code": "DEPT-012", "department_name": "Contract", "description": "Contract Department", "id_department": 12 }, { "department_code": "DEPT-020", "department_name": "E&I and Automation", "description": "Electrical & Instrumentation and Automation Department", "id_department": 20 }, { "department_code": "DEPT-017", "department_name": "Engineering", "description": "Engineering Department", "id_department": 17 }, { "department_code": "DEPT-015", "department_name": "Finance", "description": "Finance Department", "id_department": 15 },

21.	*/api/department/search?q=...*	*GET*	{
			"departments": [
			{
			"department_code": "DEPT-003",
			"department_name": "IYM",
			"description": "IYM Department",
			"id_department": 3
			}
			],
			"success": true,
			"total": 1
			}

22.	*/api/department/<id>*	*GET*	{
			"department": {
			"created_at": "Fri, 17 Apr 2026 07:41:44 GMT",
			"department_code": "DEPT-001",
			"department_name": "Secondee",
			"description": "Secondee Department - Expatriate Staff",
			"id_department": 1,
			"updated_at": "Fri, 17 Apr 2026 07:41:44 GMT"
			},
			"success": true
			}

23.	*/api/department/code/<code>*	*GET*	{
			"department": {
			"created_at": "Fri, 17 Apr 2026 07:41:44 GMT",
			"department_code": "DEPT-002",
			"department_name": "QA & QC",
			"description": "Quality Assurance & Quality Control Department",
			"id_department": 2,
			"updated_at": "Fri, 17 Apr 2026 07:41:44 GMT"
			},
			"success": true
			}

24.	*/api/projects/list*	*GET*	{ "data": [{ "id_project": 9, "is_active": true, "project_code": "BETA", "project_name": "Beta" }, { "id_project": 7, "is_active": true, "project_code": "BP_KASKIDA_FPU", "project_name": "BP KASKIDA FPU" }, { "id_project": 10, "is_active": true, "project_code": "CHANGHUA", "project_name": "Changhua" }, { "id_project": 8, "is_active": true, "project_code": "EMPIRE", "project_name": "Empire" }, { "id_project": 4, "is_active": true, "project_code": "FPSO_PETROBRAS_P_84", "project_name": "FPSO PETROBRAS P-84" }, { "id_project": 5, "is_active": true, "project_code": "FPSO_PETROBRAS_P_85", "project_name": "FPSO PETROBRAS P-85" },
25.	*/api/master-devices/list*	*GET*	{ "data": [{ "category_id": null, "device_name": "Anviz", "id_device": 1, "is_active": true }, { "category_id": null, "device_name": "Converter", "id_device": 2, "is_active": true }, { "category_id": null, "device_name": "Fingerprint", "id_device": 3, "is_active": true }, { "category_id": null, "device_name": "Ipphone", "id_device": 4, "is_active": true }, { "category_id": null, "device_name": "Jabra Speaker", "id_device": 5, "is_active": true }, { "category_id": null, "device_name": "Keyboard", "id_device": 6, "is_active": true },

26.	*/api/master-materials/list*	*GET*	{ "data": [{ "category_id": null, "id_material": 1, "is_active": true, "material_name": "Cable LAN" }, { "category_id": null, "id_material": 11, "is_active": true, "material_name": "elbow" }, { "category_id": null, "id_material": 2, "is_active": true, "material_name": "Flexible" }, { "category_id": null, "id_material": 3, "is_active": true, "material_name": "FO" }, { "category_id": null, "id_material": 10, "is_active": true, "material_name": "Isolasi Rubber" }, { "category_id": null, "id_material": 7, "is_active": true, "material_name": "Junction Box" },
27.	*/api/master-receivers/list*	*GET*	{ "data": [{ "department_id": 7, "department_name": "HR & Admin", "id_receiver": 69, "is_active": true, "receiver_name": "Aan wahyudi", "receiver_title": "Senior HR Officer" }, { "department_id": 6, "department_name": "Structure", "id_receiver": 187, "is_active": true, "receiver_name": "Aang Prasetyo", "receiver_title": "Production Engineer" }, { "department_id": 1, "department_name": "Secondee", "id_receiver": 7, "is_active": true, "receiver_name": "ABDUL RASHEED MOHAMED NIZAR", "receiver_title": "OPERATION MANAGER" }, { "department_id": 2, "department_name": "QA & QC", "id_receiver": 100, "is_active": true, "receiver_name": "Abdul Syahbro", "receiver_title": "Senior QA Engineer" }, { "department_id": 5, "department_name": "Works", "id_receiver": 126,

28.	*/api/devices/scanning-preparation/create*	*POST*	{
			"checking_number": "SCAN-20260417-DDTBPW",
			"message": "Devices scanning preparation created successfully",
			"preparation_id": 2,
			"success": true,
			"total_items_created": 5
			}
			
			{
			"checking_number": "SCAN-20260417-2EQ31L",
			"message": "Devices scanning preparation created successfully",
			"preparation_id": 3,
			"success": true,
			"total_items_created": 3
			}

29.	/api/devices/scanning-preparation/list	GET	{
			"count": 3,
			"data": [
			{
			"category_id": 1,
			"category_name": "Devices",
			"checking_date": "Fri, 17 Apr 2026 00:00:00 GMT",
			"checking_name": "Stock Laptop",
			"checking_number": "SCAN-20260417-2EQ31L",
			"created_at": "Fri, 17 Apr 2026 10:49:15 GMT",
			"created_by_name": "Clinton Alfaro Siagian",
			"id_preparation": 3,
			"items": [
			{
			"brand": "Dell",
			"device_detail": "Laptop for stock",
			"device_name": "Laptop",
			"id_item": 3,
			"model": "Latitude 3240",
			"project_name": "FPSO PETROBRAS P-84",
			"quantity": 3,
			"scanned_count": 0,
			"specifications": "Intel i5, 8GB RAM, 256GB SSD
			"status": "pending",
			"vendor": "PT DUTA"
			}
			],
			"location_id": 1,
			"location_name": "Main Office Lv1",
			"progress": 0,
			"project_name": "FPSO PETROBRAS P-84",
			"remarks": "Save to stock from Postman",
			"scannedCount": 0,
			"status": "pending",
			"totalItems": 1,
			"totalQty": 3,
			"type": "device",
			"updated_at": "Fri, 17 Apr 2026 10:49:15 GMT",

30.	/api/devices/scanning-preparation/<id>	GET	{
			"data": {
			"category_id": 1,
			"category_name": "Devices",
			"checking_date": "Fri, 17 Apr 2026 00:00:00 GMT",
			"checking_name": "Checking Laptop Dell",
			"checking_number": "SCAN-20260417-DDTBPW",
			"created_at": "Fri, 17 Apr 2026 10:48:09 GMT",
			"created_by_name": "Clinton Alfaro Siagian",
			"id_preparation": 2,
			"items": [
			{
			"brand": "Dell",
			"created_at": "Fri, 17 Apr 2026 10:48:09 GMT",
			"departments": [
			{
			"department_id": 1,
			"department_name": "Secondee",
			"quantity": 2
			},
			{
			"department_id": 2,
			"department_name": "QA & QC",
			"quantity": 3
			}
			],
			"device_detail": "Laptop Dell New Version",
			"device_name": "Laptop",
			"id_item": 2,
			"model": "Latitude 3240",
			"project_name": "FPSO PETROBRAS P-84",
			"quantity": 5,
			"receivers": [
			{
			"department_id": 1,
			"department_name": "Secondee",
			"is_stock": true,
			"item_index": 0,

31.	/api/devices/scanning-preparation/<id>	PUT	{
			"message": "Devices scanning preparation updated successfully",
			"preparation_id": 1,
			"success": true
			}

32.	/api/devices/scanning-preparation/<id>	DELETE	{
			"message": "Devices scanning preparation deleted successfully",
			"success": true
			}

33.	*/api/devices/scanning-preparation/<id>/progress*	*GET*	{
			"data": {
			"overall_percentage": 20,
			"progress": [
			{
			"brand": "Dell",
			"id_item": 1,
			"item_name": "Laptop",
			"model": "Latitude 3240",
			"percentage": 20,
			"quantity": 5,
			"scanned": 1
			}
			],
			"scan_results": [
			{
			"brand": "Dell",
			"department_id": 3,
			"department_name": "IYM",
			"id_item": 1,
			"id_scan": 1,
			"is_valid": false,
			"item_name": "Laptop",
			"item_number": "ITEM-1-1-1",
			"item_preparation_id": 1,
			"item_quantity": 5,
			"model": "Latitude 3240",
			"photo_url": "/uploads/scan_photos/scan_20260417_100011_011ff9e2.jpg",
			"project_name": "FPSO PETROBRAS P-84",
			"receiver_id": 30,
			"receiver_name": "Edang Gunawan",
			"scan_category": "Devices",
			"scan_value": "Laptop",
			"scanned_at": "Fri, 17 Apr 2026 10:00:11 GMT",
			"scanning_item_id": 1,
			"serial_number": "sn123213jk12",
			"status": "submitted"
			}

34.	*/api/materials/scanning-preparation/create*	*POST*	{
			"checking_number": "SCAN-20260417-DALC26",
			"message": "Materials scanning preparation created successfully",
			"preparation_id": 1,
			"success": true,
			"total_items_created": 5
			}
			
			{
			"checking_number": "SCAN-20260417-6JFPWR",
			"message": "Materials scanning preparation created successfully",
			"preparation_id": 2,
			"success": true,
			"total_items_created": 10
			}

35.	*/api/materials/scanning-preparation/list*	*GET*	{
			"count": 2,
			"data": [
			{
			"category_id": 2,
			"category_name": "Materials",
			"checking_date": "Fri, 17 Apr 2026 00:00:00 GMT",
			"checking_name": "Material Scan Test",
			"checking_number": "SCAN-20260417-6JFPWR",
			"created_at": "Fri, 17 Apr 2026 11:11:09 GMT",
			"created_by_name": "Clinton Alfaro Siagian",
			"id_preparation": 2,
			"items": [
			{
			"id_item": 2,
			"material_detail": "CAT6 Blue",
			"material_name": "Cable LAN",
			"project_name": "Gamma",
			"quantity": 10.0,
			"scanned_count": 0,
			"status": "pending",
			"uom": "PCS",
			"vendor": "Belden"
			}
			],
			"location_id": 1,
			"location_name": "Main Office Lv1",
			"progress": 0,
			"project_name": "Gamma",
			"remarks": "Testing from Postman",
			"scannedCount": 0,
			"status": "pending",
			"totalItems": 1,
			"totalQty": 10.0,
			"type": "material",
			"updated_at": "Fri, 17 Apr 2026 11:11:09 GMT",
			"user_id": 1
			},

36.	*/api/materials/scanning-preparation/<id>*	*GET*	{
			"data": {
			"category_id": 2,
			"category_name": "Materials",
			"checking_date": "Fri, 17 Apr 2026 00:00:00 GMT",
			"checking_name": "Material Distribution Test",
			"checking_number": "SCAN-20260417-DALC26",
			"created_at": "Fri, 17 Apr 2026 11:10:34 GMT",
			"created_by_name": "Clinton Alfaro Siagian",
			"id_preparation": 1,
			"items": [
			{
			"created_at": "Fri, 17 Apr 2026 11:10:34 GMT",
			"departments": [
			{
			"department_id": 1,
			"department_name": "Secondee",
			"quantity": 3.0
			}
			],
			"id_item": 1,
			"item_name": "Cable LAN",
			"project_name": "Gamma",
			"quantity": "5.00",
			"receivers": [
			{
			"department_id": 1,
			"department_name": "Secondee",
			"item_index": 0,
			"receiver_id": 10,
			"receiver_name": "GHOSH SUKANTA"
			},
			{
			"department_id": 1,
			"department_name": "Secondee",
			"item_index": 1,
			"receiver_id": 11,
			"receiver_name": "VELLIAN SELVARAJ"

37.	*/api/materials/scanning-preparation/<id>*	*PUT*	{
			"message": "Materials scanning preparation updated successfully",
			"preparation_id": 1,
			"success": true,
			"total_items_created": 5
			}

38.	*/api/materials/scanning-preparation/<id>*	*DELETE*	{
			"message": "Materials scanning preparation deleted successfully",
			"success": true
			}

39.	*/api/materials/scanning-preparation/<id>/progress*	*GET*	{
			"data": {
			"overall_percentage": 0,
			"progress": [],
			"scan_results": [],
			"total_scanned": 0,
			"total_target": 0,
			"type": "material"
			},
			"success": true
			}

40.	*/api/materials/uom*	*GET*	{
			"data": [
			{
			"uom_code": "BOX",
			"uom_name": "Box"
			},
			{
			"uom_code": "KG",
			"uom_name": "Kilogram"
			},
			{
			"uom_code": "METER",
			"uom_name": "Meter"
			},
			{
			"uom_code": "PACK",
			"uom_name": "Pack"
			},
			{
			"uom_code": "PCS",
			"uom_name": "Pieces"
			},
			{
			"uom_code": "ROLL",
			"uom_name": "Roll"
			},
			{
			"uom_code": "UNIT",
			"uom_name": "Unit"
			}
			],
			"success": true
			}

41.	*/api/scanning-preparation/list-all*	*GET*	{
			"data": [
			{
			"category_id": 2,
			"category_name": "Materials",
			"checking_date": "Fri, 17 Apr 2026 00:00:00 GMT",
			"checking_name": "Material Scan Test",
			"checking_number": "SCAN-20260417-6JFPWR",
			"created_at": "Fri, 17 Apr 2026 11:11:09 GMT",
			"created_by_name": "Clinton Alfaro Siagian",
			"id_preparation": 2,
			"items": [
			{
			"id_item": 2,
			"material_detail": "CAT6 Blue",
			"material_name": "Cable LAN",
			"project_name": "Gamma",
			"quantity": 10.0,
			"scanned_count": 0,
			"status": "pending",
			"uom": "PCS",
			"vendor": "Belden"
			}
			],
			"location_id": 1,
			"location_name": "Main Office Lv1",
			"progress": 0,
			"project_name": "Gamma",
			"remarks": "Testing from Postman",
			"scannedCount": 0,
			"status": "pending",
			"totalItems": 1,
			"totalQty": 10.0,
			"type": "material",
			"updated_at": "Fri, 17 Apr 2026 11:11:09 GMT",
			"user_id": 1
			},
			{

42.	*/api/scanning-preparation/sync-status*	*POST*	{
			"devices_updated": 0,
			"materials_updated": 0,
			"message": "Updated 0 device sessions and 0 material sessions to completed",
			"success": true
			}

43.	*/api/devices/items-preparation/<int: item_prep_id>*	*GET*	{
			"data": {
			"brand": "Dell",
			"created_at": "Fri, 17 Apr 2026 07:49:43 GMT",
			"department_id": 3,
			"device_name": "Laptop",
			"id_item_preparation": 1,
			"item_number": "ITEM-1-1-1",
			"model": "Latitude 3240",
			"preparation_id": 1,
			"project_name": "FPSO PETROBRAS P-84",
			"receiver_id": 30,
			"scanned_at": "Fri, 17 Apr 2026 10:00:11 GMT",
			"scanning_item_id": 1,
			"serial_number": null,
			"specifications": "Intel i5, 8GB RAM, 256GB SSD",
			"status": "scanned"
			},
			"success": true
			}

44.	*/api/devices/items-preparation/<prepId>/item/<itemId>/available*	*GET*	{
			"data": {
			"brand": "Dell",
			"created_at": "Fri, 17 Apr 2026 11:02:32 GMT",
			"department_id": 1,
			"device_name": "Laptop",
			"id_item_preparation": 14,
			"is_stock": false,
			"item_number": "ITEM-1-1-1",
			"model": "Latitude 3240",
			"notes": null,
			"preparation_id": 1,
			"project_name": "FPSO PETROBRAS P-84",
			"receiver_id": 10,
			"scanned_at": null,
			"scanned_by": null,
			"scanning_item_id": 1,
			"serial_number": null,
			"specifications": "Intel i5, 8GB RAM, 256GB SSD",
			"status": "pending",
			"updated_at": "Fri, 17 Apr 2026 11:02:32 GMT",
			"user_id": 1
			},
			"success": true
			}

45.	*/api/materials/items-preparation/<int: item_prep_id>*	*GET*	{
			"data": {
			"created_at": "Fri, 17 Apr 2026 11:11:09 GMT",
			"department_id": null,
			"id_item_preparation": 6,
			"item_number": "ITEM-2-1-1",
			"material_name": "Cable LAN",
			"preparation_id": 2,
			"project_name": "Gamma",
			"quantity": "1.00",
			"receiver_id": null,
			"scan_code": null,
			"scanned_at": null,
			"scanning_item_id": 2,
			"status": "pending",
			"uom": "PCS",
			"vendor": "Belden"
			},
			"success": true
			}

46.	/api/materials/items-preparation/<prepId>/item/<itemId>/available	GET	{
			"data": {
			"created_at": "Fri, 17 Apr 2026 11:11:09 GMT",
			"department_id": null,
			"id_item_preparation": 6,
			"is_stock": false,
			"item_number": "ITEM-2-1-1",
			"material_name": "Cable LAN",
			"notes": null,
			"preparation_id": 2,
			"project_name": "Gamma",
			"quantity": "1.00",
			"receiver_id": null,
			"scan_code": null,
			"scanned_at": null,
			"scanned_by": null,
			"scanning_item_id": 2,
			"status": "pending",
			"uom": "PCS",
			"updated_at": "Fri, 17 Apr 2026 11:11:09 GMT",
			"user_id": 1,
			"vendor": "Belden"
			},
			"success": true
			}

47.	/api/scan-results/create-device	POST	{
			"message": "Device scan result saved successfully",
			"photo_url": null,
			"scan_id": 2,
			"success": true
			}

48.	/api/scan-results/device/<id>	PUT	{
			"message": "Scan result updated successfully",
			"success": true
			}

50.	/api/scan-results/device/<id>	DELETE	{
			"message": "Scan result deleted successfully",
			"success": true
			}

51.	/api/scan-results/create-material	POST	{
			"message": "Material scan result saved successfully",
			"photo_url": null,
			"scan_id": 1,
			"success": true
			}

52.	/api/scan-results/material/<id>	PUT	{
			"message": "Scan result updated successfully",
			"success": true
			}

53.	/api/scan-results/material/<id>	DELETE	{
			"message": "Scan result deleted successfully",
			"success": true
			}

54.	/api/scan-results/check-serial?serial=..	GET	{
			"exists": true,
			"success": true
			}

56.	/api/scan-results/check-scan-code?code=...	GET	{
			"exists": true,
			"success": true
			}

57.	/api/validations/create	POST	{
			"message": "Validation created successfully",
			"success": true,
			"unique_code": "VAL-20260417131351-DUBV0O",
			"validation_id": 3
			}

58.	/api/validations	GET	{
			"count": 4,
			"data": [
			{
			"brand": null,
			"checking_name": "Check LAN",
			"checking_number": "SCAN-20260417-ST50QT",
			"created_at": "Fri, 17 Apr 2026 13:14:57 GMT",
			"created_by_name": "Clinton Alfaro Siagian",
			"departments": [
			{
			"department_id": 11,
			"department_name": "IT",
			"quantity": 1
			}
			],
			"device_checking_name": null,
			"device_checking_number": null,
			"device_item_prep_id": null,
			"device_name": null,
			"device_photo": null,
			"device_preparation_id": null,
			"id_validation": 4,
			"is_approved": false,
			"item_name": "Cable LAN",
			"location_name": "Site Office 2",
			"material_checking_name": "Check LAN",
			"material_checking_number": "SCAN-20260417-ST50QT",
			"material_item_prep_id": 1,
			"material_name": "Cable LAN",
			"material_photo": null,
			"material_preparation_id": 1,
			"model": null,
			"photo_url": null,
			"project_name": "Empire",
			"receivers": [
			{
			"department_name": "IT",

59.	/api/validations/<id>	PUT	{
			"message": "Validation approved successfully",
			"success": true
			}

60.	/api/validations/<id>	DELETE	{
			"message": "Validation deleted successfully",
			"success": true
			}

61.	/api/validations/<id>/detail	GET	{
			"data": {
			"asset_id": null,
			"brand": null,
			"checking_date": "Fri, 17 Apr 2026 00:00:00 GMT",
			"checking_name": "Check LAN",
			"checking_number": "SCAN-20260417-ST50QT",
			"created_at": "Fri, 17 Apr 2026 12:58:31 GMT",
			"created_by_name": "Clinton Alfaro Siagian",
			"departments": [
			{
			"department_name": "IT",
			"quantity": 1
			}
			],
			"detection_data": {
			"asset_type": "Cable LAN",
			"category": "Material",
			"confidence": 100,
			"source": "manual"
			},
			"device_checking_date": "Fri, 17 Apr 2026 00:00:00 GMT",
			"device_checking_name": "Daas Laptop",
			"device_checking_number": "SCAN-20260417-WY3T4Z",
			"device_detection": null,
			"device_name": null,
			"device_photo": null,
			"id_location": 4,
			"id_validation": 1,
			"is_approved": false,
			"item_name": "Cable LAN",
			"item_preparation_id": 1,
			"location_name": "Engineering Office Lv1",
			"material_checking_date": "Fri, 17 Apr 2026 00:00:00 GMT",
			"material_checking_name": "Check LAN",
			"material_checking_number": "SCAN-20260417-ST50QT",
			"material_detection": {
			"asset_type": "Cable LAN",

62.	*/api/validations/bulk*	*POST*	{
			"message": "1 validations approved",
			"success": true,
			"updated_count": 1
			}
			
			{
			"message": "1 validations rejected",
			"success": true,
			"updated_count": 1
			}

63.	*/api/validations/bulk-delete*	*POST/ DELETE*	{
			"message": "Validation deleted successfully",
			"success": true
			}

64.	*/api/assets*	*GET*	{ "count": 2, "data": [{ "asset_code": "AST-20260417-URBV80", "asset_name": "Laptop", "asset_type": "device", "brand": "Dell", "category": "Device", "created_at": "Fri, 17 Apr 2026 13:34:19 GMT", "department_name": "IT", "id_assets": 2, "location_id": 5, "location_name": "Engineering Office Lv2", "model": "LATITUDE 3240", "photo_url": null, "project_name": "FPSO PETROBRAS P-84", "quantity": "1.00", "receiver_name": "Gian Sunarto", "scan_code": null, "serial_number": "SN23T12H32", "specifications": "INTEL I5 GB", "status": "active", "uom": null, "updated_at": "Fri, 17 Apr 2026 13:34:19 GMT", "user_id": 1, "validated_at": "Fri, 17 Apr 2026 13:34:19 GMT", "validated_by": 1, "validated_by_name": "Clinton Alfaro Siagian", "validation_id": 3, "vendor": "DUTA COMPUTER" }, { "asset_code": "AST-20260417-E9WYEH", "asset_name": "Laptop", "asset_type": "device", "brand": "Dell", "category": "Device",

65.	*/api/assets/<id>*	*GET*	{
			"data": {
			"asset_code": "AST-20260417-E9WYEH",
			"asset_name": "Laptop",
			"asset_type": "device",
			"brand": "Dell",
			"category": "Device",
			"created_at": "Fri, 17 Apr 2026 13:34:13 GMT",
			"created_by_name": "Clinton Alfaro Siagian",
			"department_name": "IT",
			"id_assets": 1,
			"location_id": 5,
			"location_name": "Engineering Office Lv2",
			"model": "LATITUDE 3240",
			"photo_url": null,
			"project_name": "FPSO PETROBRAS P-84",
			"quantity": "1.00",
			"receiver_name": "Wahyu Hidayat",
			"scan_code": null,
			"serial_number": "SN123G1U2I3G21",
			"specifications": "INTEL I5 GB",
			"status": "active",
			"uom": null,
			"updated_at": "Fri, 17 Apr 2026 13:34:13 GMT",
			"user_id": 1,
			"validated_at": "Fri, 17 Apr 2026 13:34:13 GMT",
			"validated_by": 1,
			"validated_by_name": "Clinton Alfaro Siagian",
			"validation_id": 4,
			"vendor": "DUTA COMPUTER"
			},
			"success": true
			}

68.	*/api/assets/<id>*	*DELETE*	{
			"message": "Asset deleted successfully",
			"success": true
			}

69.	*/api/assets/by-preparation/<prepId>*	*GET*	{
			"count": 1,
			"data": [
			{
			"asset_code": "AST-20260417-E9WYEH",
			"asset_name": "Laptop",
			"asset_type": "device",
			"brand": "Dell",
			"category": "Device",
			"created_at": "Fri, 17 Apr 2026 13:34:13 GMT",
			"department_name": "IT",
			"device_brand": "Dell",
			"device_model": "LATITUDE 3240",
			"device_name": "Laptop",
			"device_specs": "INTEL I5 GB",
			"device_vendor": "DUTA COMPUTER",
			"id_assets": 1,
			"item_number": "ITEM-1-1-1",
			"item_serial": null,
			"location_name": "Engineering Office Lv2",
			"model": "LATITUDE 3240",
			"photo_url": null,
			"project_name": "FPSO PETROBRAS P-84",
			"quantity": "1.00",
			"receiver_name": "Wahyu Hidayat",
			"scan_code": null,
			"serial_number": "SN123G1U2I3G21",
			"specifications": "INTEL I5 GB",
			"status": "active",
			"uom": null,
			"validated_at": "Fri, 17 Apr 2026 13:34:13 GMT",
			"validated_by_name": "Clinton Alfaro Siagian",
			"vendor": "DUTA COMPUTER"
			}
			],
			"session_info": {
			"location_name": "Engineering Office Lv2",
			"session_date": "Fri, 17 Apr 2026 00:00:00 GMT",

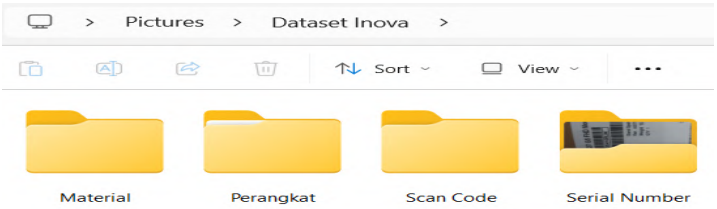
70.	*/api/assets/preparations/completed*	*GET*	{
			"count": 0,
			"data": [],
			"success": true
			}

4.1.3 Implementasi Model *AI*

Pada subbab ini akan menjelaskan implementasi model *AI* yang akan digunakan dalam sistem, mulai dari proses pengumpulan data, pra-pemrosesan, pelabelan *dataset*, hingga pelatihan dan *deployment* model. Model *AI* yang dikembangkan mencakup deteksi objek menggunakan *YOLO* serta ekstraksi teks menggunakan *OCR* yang terintegrasi dalam *backend* berbasis *Flask*, sehingga mampu mendukung proses *scanning* perangkat atau material serta validasi data secara otomatis dan *real-time*.

4.1.3.1 Pengumpulan Data

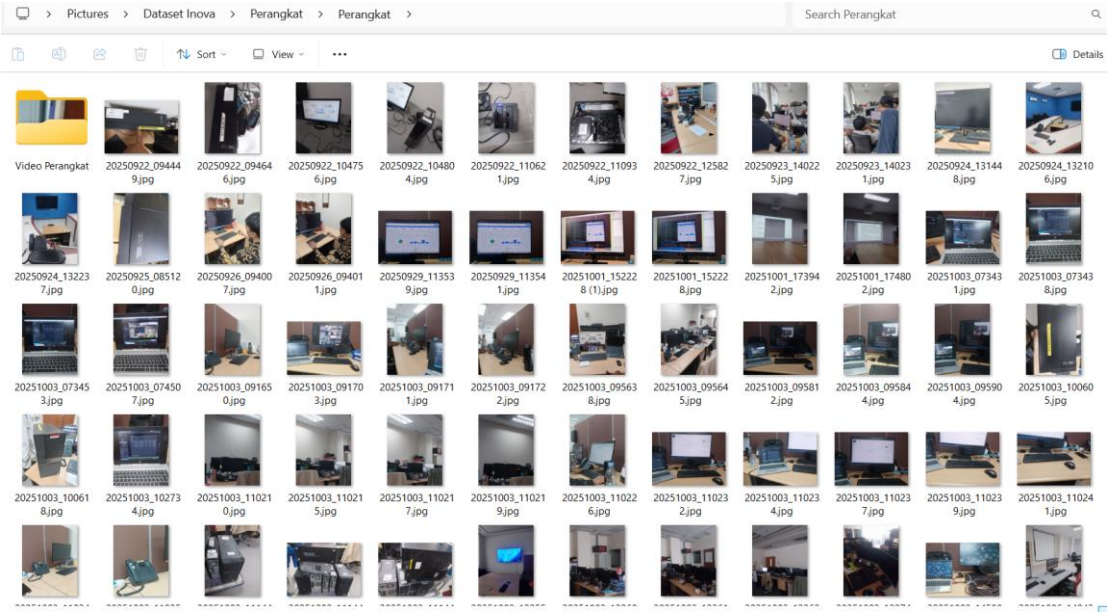
Pada tahap implementasi pengumpulan data, sistem menggunakan *dataset* yang diperoleh secara mandiri melalui pengambilan gambar dan video terhadap objek perangkat dan material di lingkungan perusahaan. Selain itu, dikumpulkan data berupa *serial number* dan *scan code* sebagai kebutuhan validasi inventaris. Data mencakup perangkat seperti laptop, *monitor*, *PC*, dan lainnya, serta material seperti *RJ45*, *Modular Jack*, dan komponen jaringan lainnya. *Dataset* kemudian disusun dalam folder utama *Dataset Inova* yang terdiri dari kategori Perangkat, Material, *Serial Number*, dan *Scan* Code untuk mendukung proses pengelolaan dan pelatihan model.

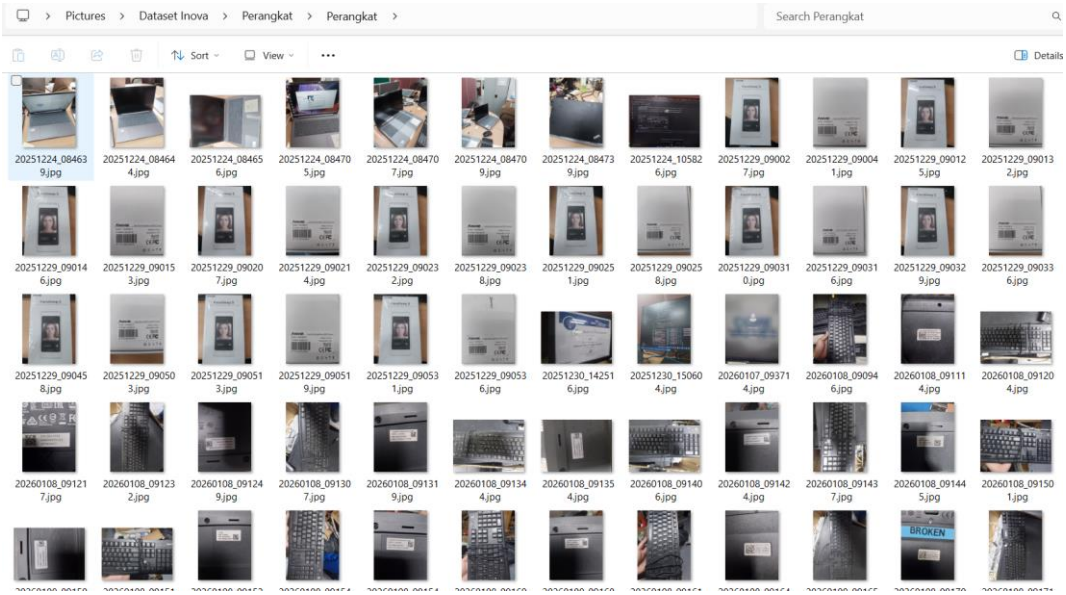


Gambar 4. 2 Struktur folder *Dataset Inova*

1. Pengumpulan Data Perangkat

   Proses pengumpulan data perangkat dilakukan melalui pengambilan gambar dan video terhadap berbagai perangkat di lingkungan perusahaan, meliputi *Anviz*, *Converter*, *Fingerprint*, *Ipphone*, *Jabra Speaker*, *Keyboard*, *Laptop*, *Monitor*, *Mouse*, *PC*, *Print Label*, *Printer*, *Telepon*, *TV*, dan *Webcam.*

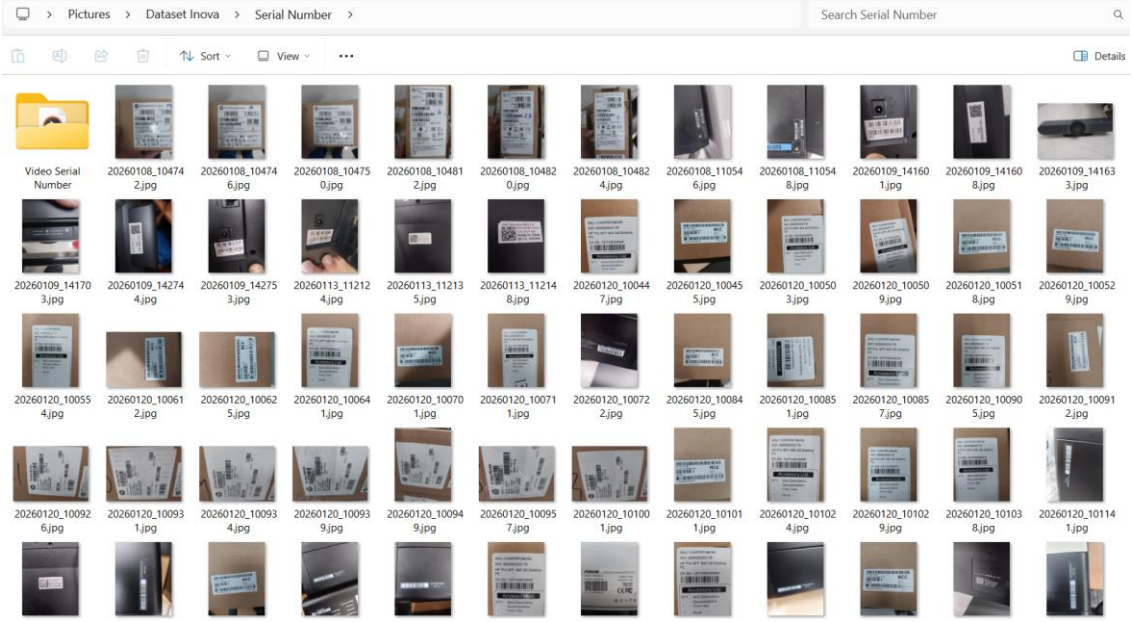


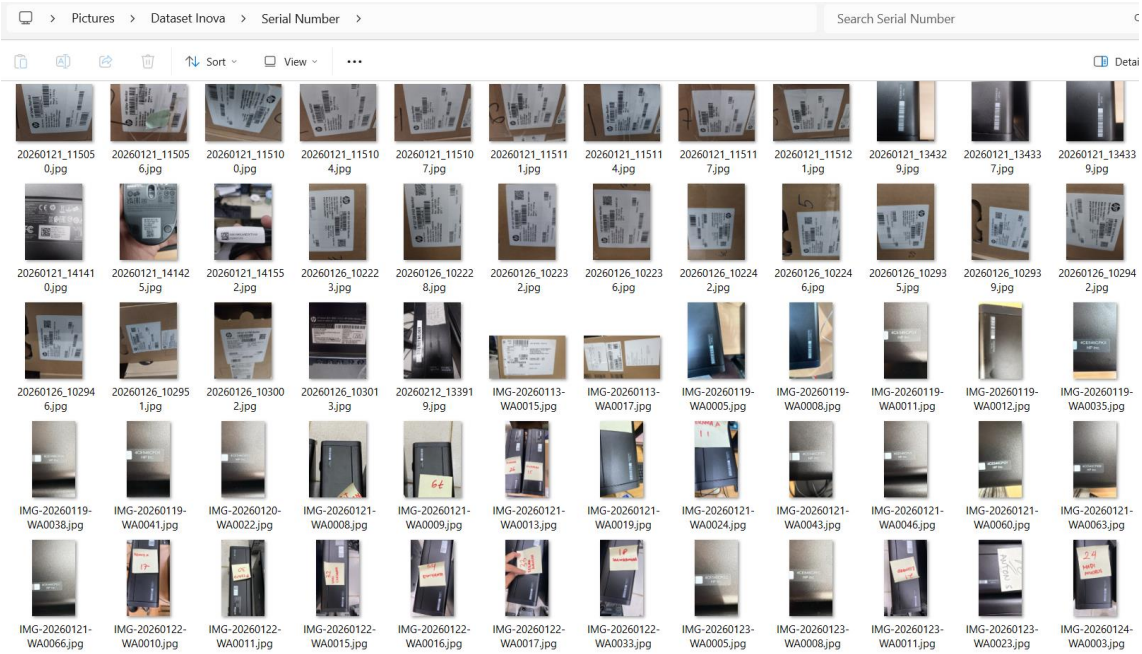


Gambar 4. 3 Pengumpulan Data Perangkat

2. Pengumpulan Data *Serial Number*

Proses pengumpulan data *serial number* dilakukan melalui pengambilan gambar dan video terhadap *serial number* yang terdapat pada masing-masing perangkat di lingkungan perusahaan sebagai kebutuhan dalam proses validasi inventaris. Data *serial number* yang dikumpulkan digunakan untuk memastikan keakuratan identifikasi setiap perangkat dalam sistem.

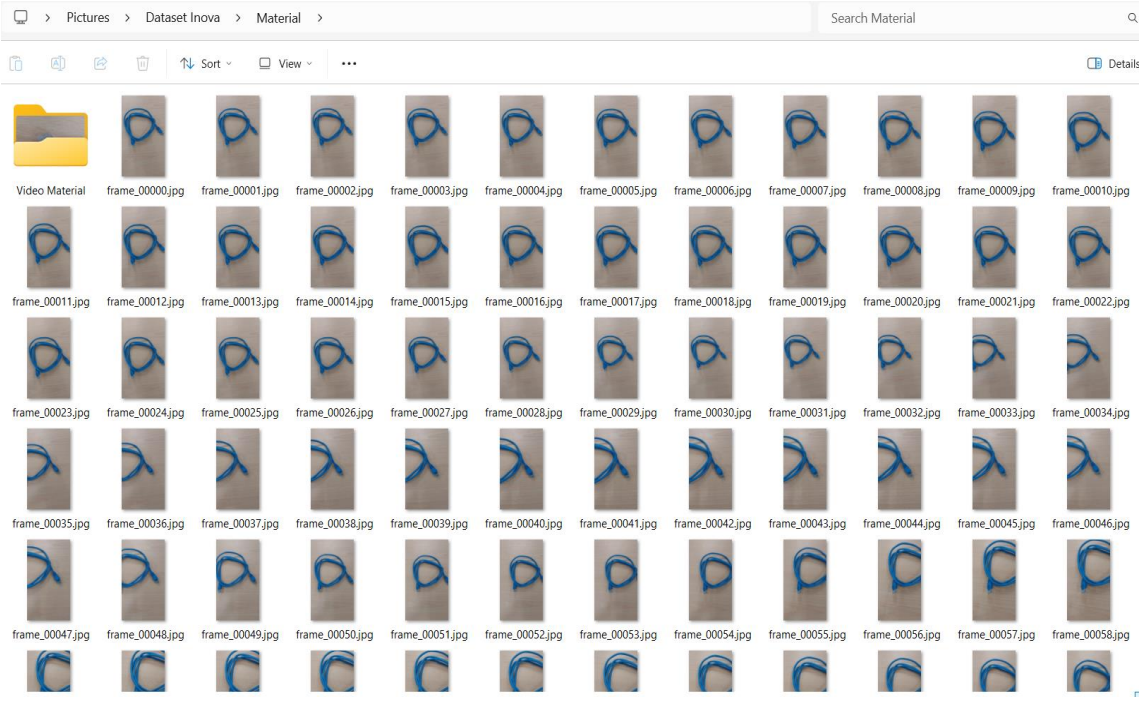


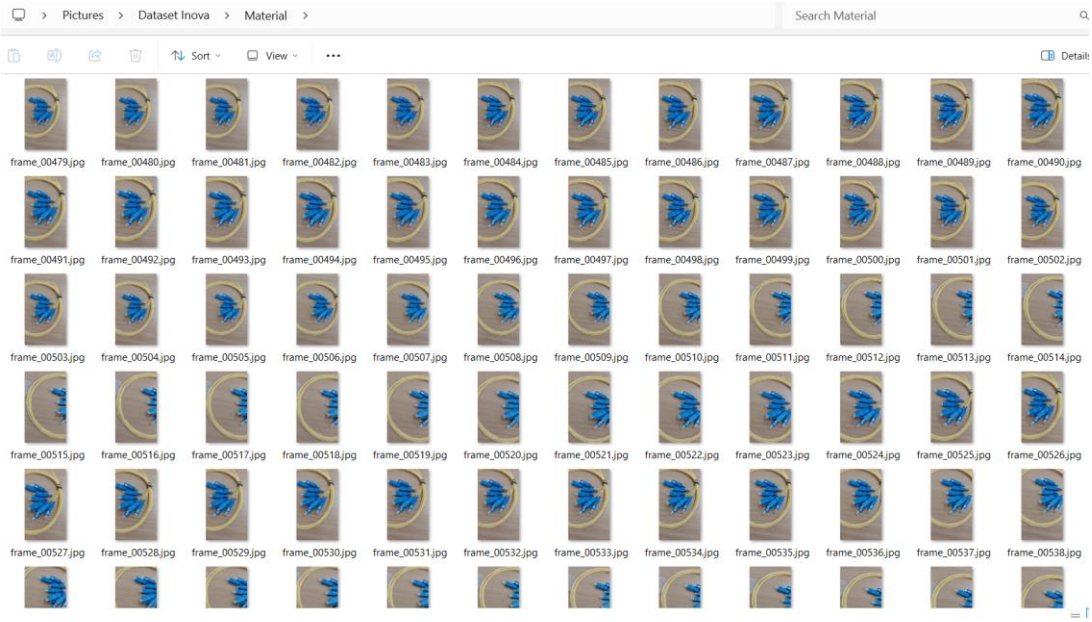


Gambar 4. 4 Pengumpulan Data *Serial Number*

3. Pengumpulan Data Material

Proses pengumpulan data material dilakukan melalui pengambilan gambar dan video terhadap berbagai material di lingkungan perusahaan, meliputi *Cable LAN*, *Flexible*, *FO*, *Trunking*, Pipa, Klem, *Junction Box*, *RJ45*, *Modular Jack*, *Isolasi Rubber*, dan *elbow*.

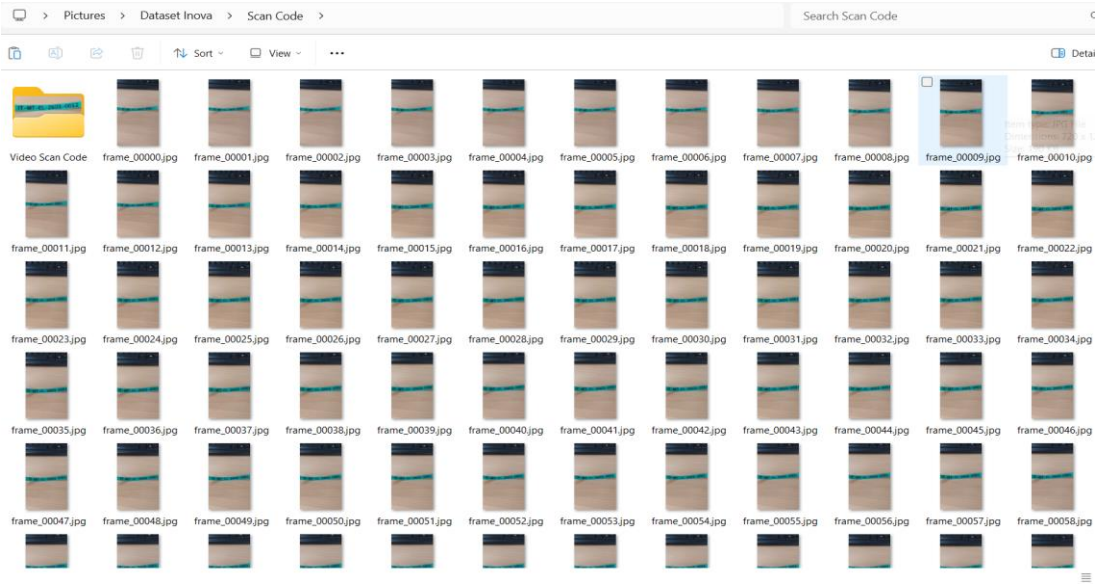




Gambar 4. 5 Pengumpulan Data Material

4. Pengumpulan Data *Scan Code*

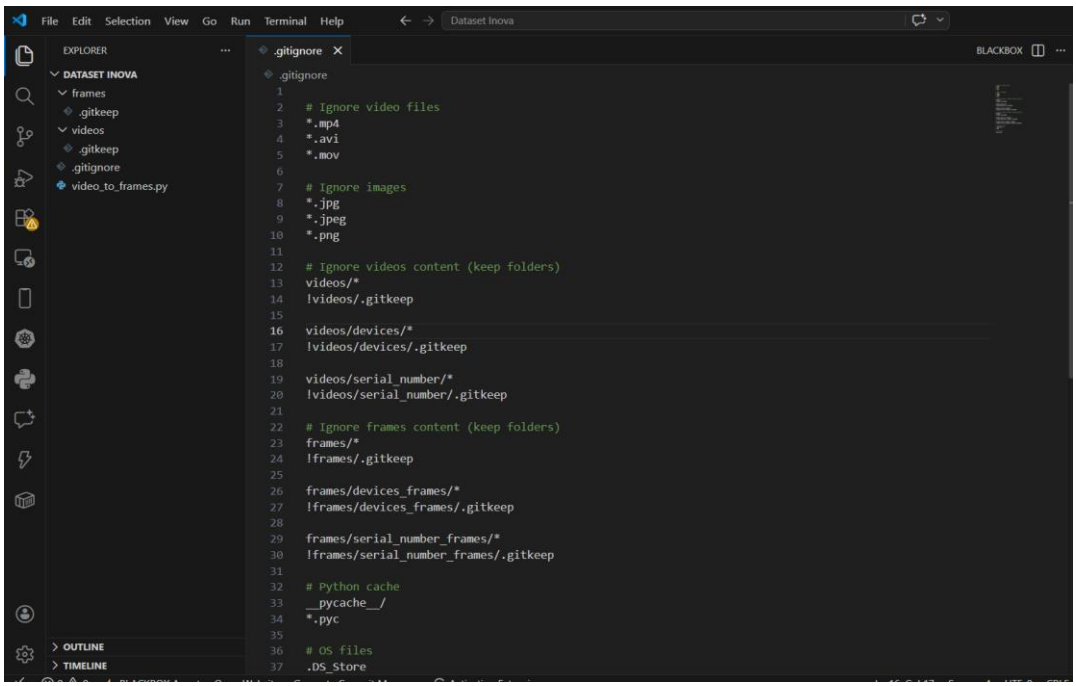
Proses pengumpulan data *scan code* dilakukan melalui pembuatan dan pengambilan gambar serta video terhadap *scan code* yang dibuat sesuai dengan format perusahaan sebagai kebutuhan dalam proses validasi inventaris. *Scan code* dibuat menggunakan kertas dan print label sesuai dengan kode masing-masing material. Data *scan code* yang dikumpulkan digunakan untuk memastikan keakuratan identifikasi setiap material dalam sistem.



Gambar 4. 6 Pengumpulan Data *Scan Code*

4.1.3.2 Implementasi Pra-pemrosesan Data

Tahap pra-pemrosesan data dilakukan untuk menyiapkan *dataset* agar siap digunakan dalam proses pelatihan model. Data yang berupa video (bukan foto) yang telah dikumpulkan kemudian diproses dengan melakukan ekstraksi *frame* menggunakan *script Python Dataset Inova* yang telah dikembangkan. Proses ini bertujuan untuk mengubah video menjadi kumpulan gambar sehingga dapat digunakan sebagai *dataset* pelatihan model *computer vision*. Setiap video dari kategori perangkat, material, *serial number*, dan *scan code* diekstraksi menjadi beberapa *frame*, kemudian disusun berdasarkan kategori untuk mempermudah pelabelan dan pelatihan model.

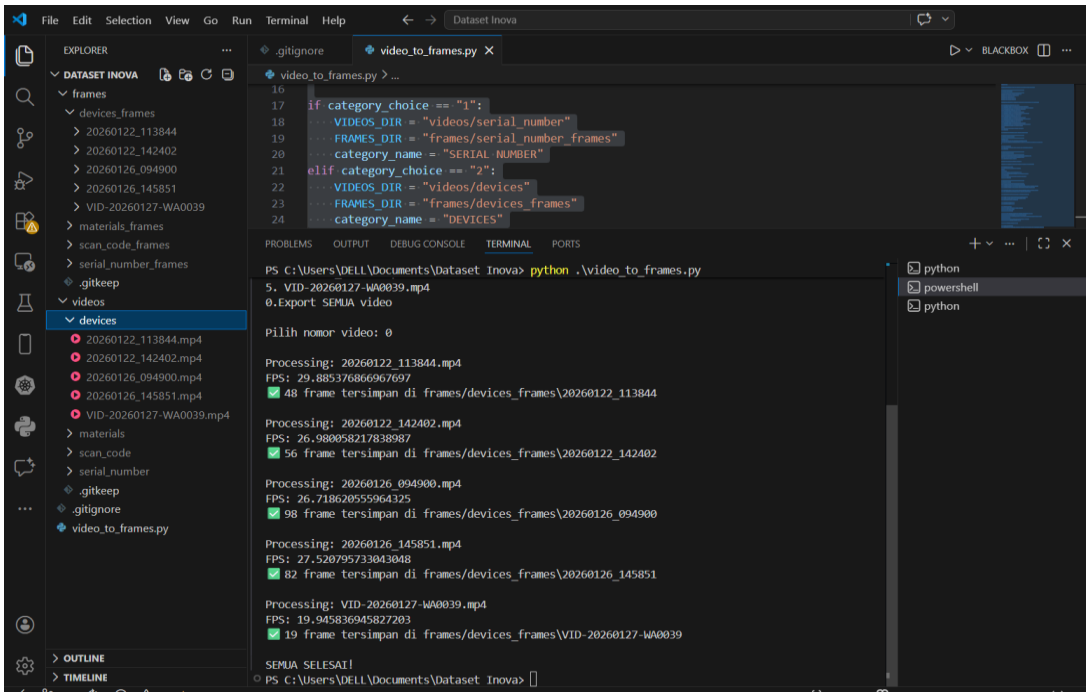


Gambar 4. 7 Struktur folder pengolahan *Dataset Inova*

Struktur *script Dataset INOVA* disusun terorganisir dengan memisahkan folder *videos* dan *frames*, di mana folder *videos* berisi data mentah berdasarkan kategori seperti *devices*, *materials*, *serial_number*, dan *scan_code*, sedangkan folder *frames* berisi hasil ekstraksi frame yang dikelompokkan sesuai kategori masing-masing seperti *devices_frames*, *materials_frames*, dan *scan_code_frames*. Proses ekstraksi dilakukan menggunakan script Python (*video_to_frames.py*) yang memanfaatkan *library OpenCV*, di mana pengguna memilih kategori video yang akan diproses,

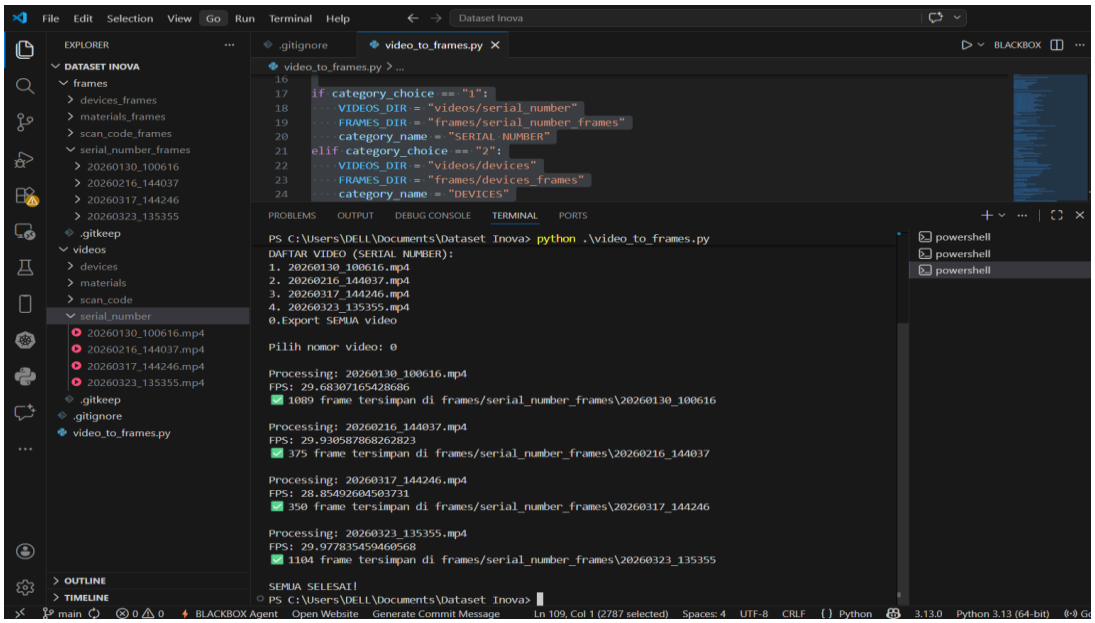
kemudian sistem membaca video dalam folder tersebut dan mengekstrak frame secara berkala. Hasil ekstraksi disimpan dalam *subfolder* sesuai nama video untuk memudahkan pengelolaan *dataset*, pelabelan, dan pelatihan model.

1. Ekstraksi *frames* Perangkat



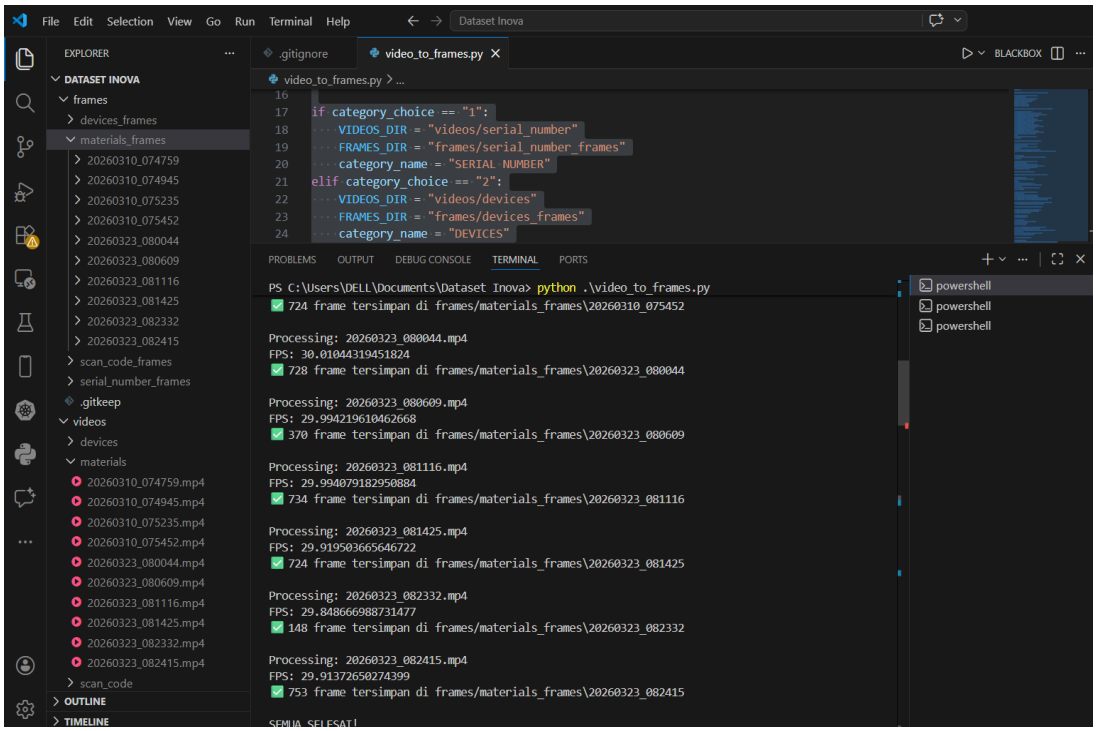
Gambar 4. 8 Ekstraksi *frames* Perangkat

2. Ekstraksi *frames Serial Number*



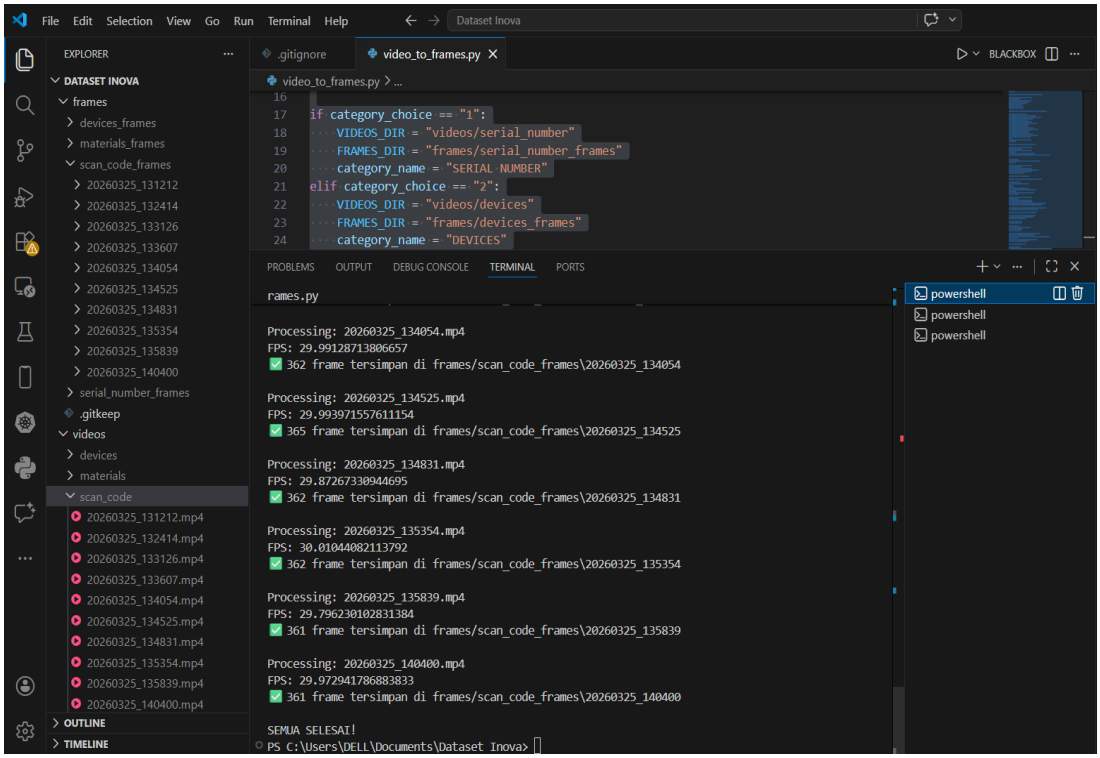
Gambar 4. 9 Ekstraksi *frames Serial Number*

3. Ekstraksi *frames* Material



Gambar 4. 10 Ekstraksi *frames* Material

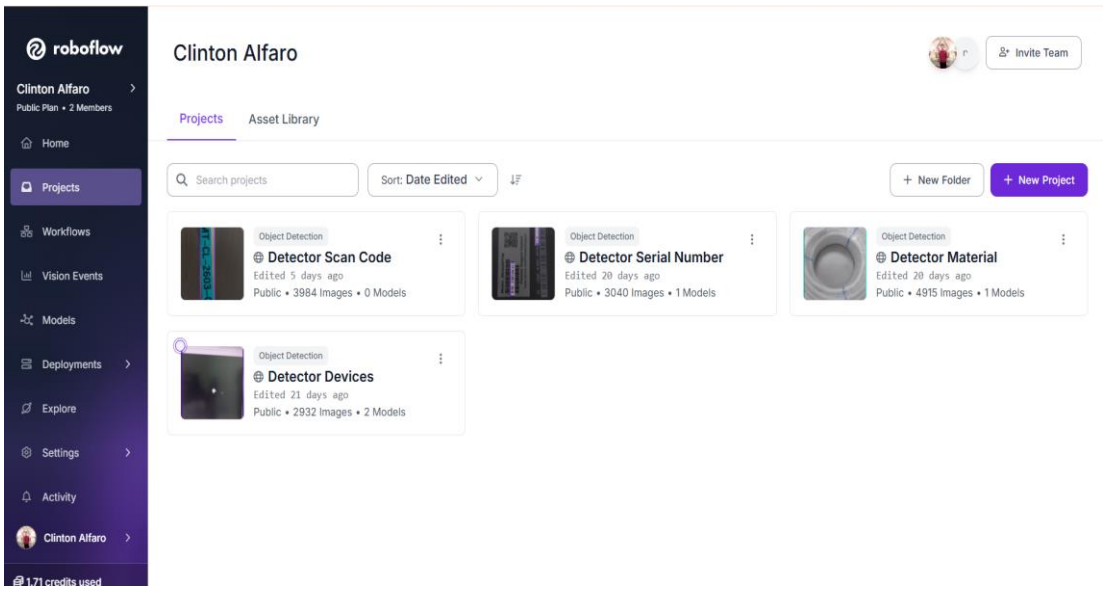
4. Ekstraksi *frames Scan Code*



Gambar 4. 11 Ekstraksi *frames Scan Code*

4.1.3.3 Implementasi Pelabelan Data *(Labeling)*

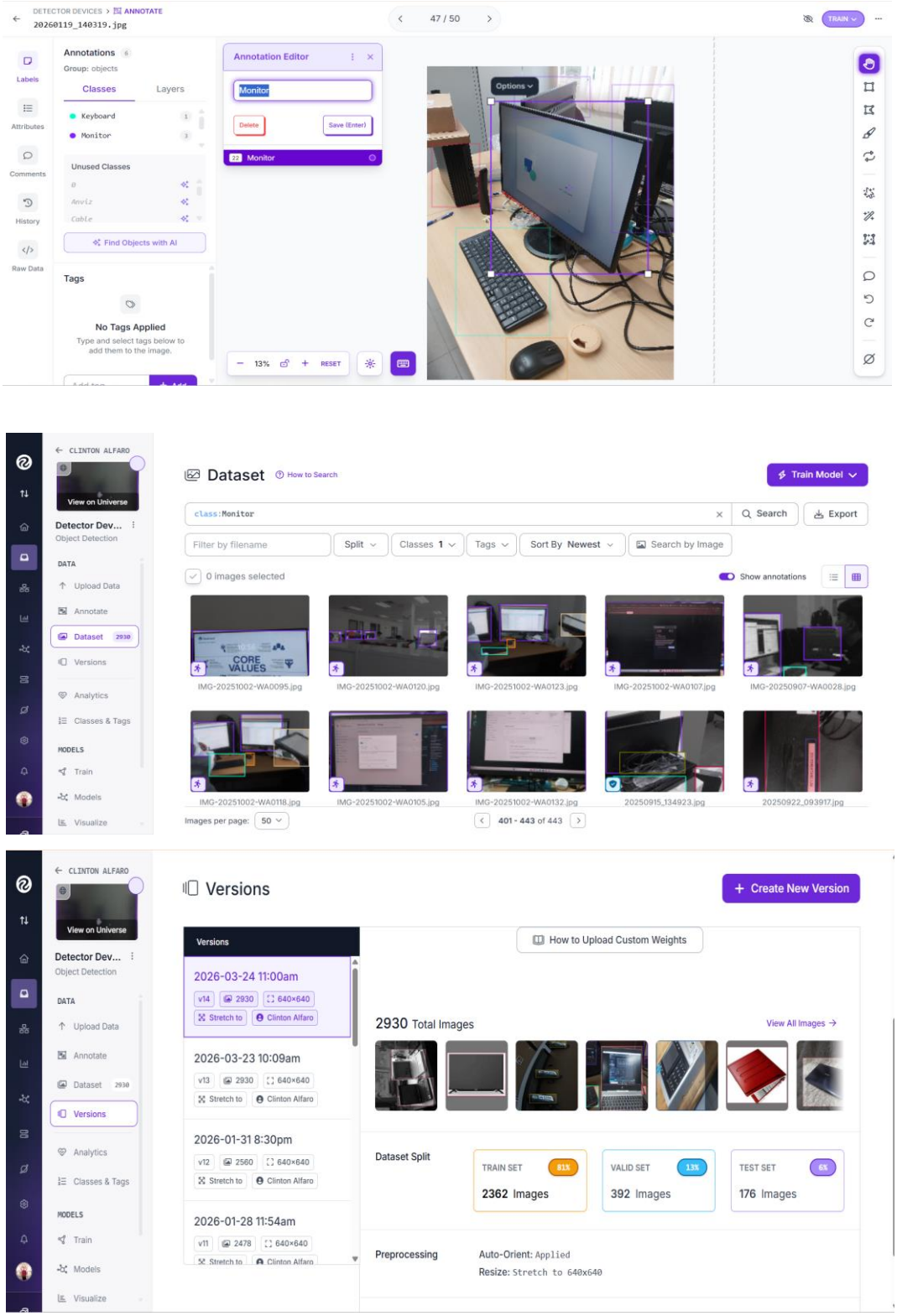
Tahap pelabelan data dilakukan untuk memberikan anotasi pada *dataset* yang telah dipersiapkan agar dapat digunakan dalam proses pelatihan model. Data gambar dari kategori perangkat, material, *serial number*, dan *scan code* diberi label sesuai dengan objek yang terdeteksi menggunakan *software* pelabelan *Roboflow*. Proses ini bertujuan untuk menentukan posisi dan kelas objek dalam gambar sehingga model dapat mengenali dan membedakan setiap objek dengan baik. Hasil pelabelan lalu disimpan dalam format yang sesuai dengan kebutuhan model deteksi objek untuk digunakan pada tahap *training*.



Gambar 4. 12 *Workspace labeling* Data *Inova*

1. Pelabelan Data Perangkat

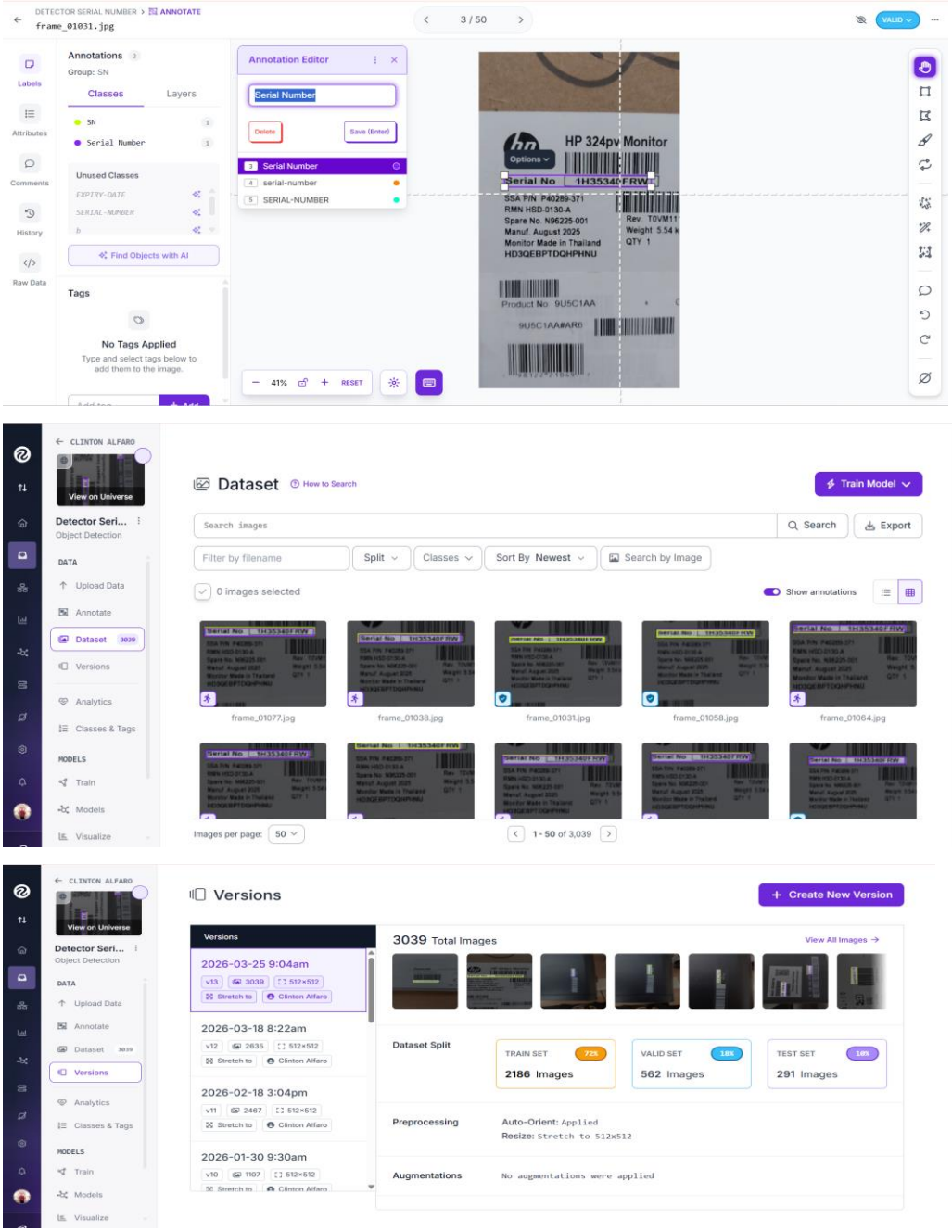
   Proses pelabelan data perangkat dilakukan terhadap *dataset* yang telah dikumpulkan untuk mengidentifikasi setiap objek perangkat dalam gambar. Total dataset yang digunakan sebanyak 2932 gambar, yang kemudian dibagi menjadi data latih (*train set*) sebesar 81% (2362 gambar), data validasi (*valid set*) sebesar 13% (392 gambar), dan data uji (*test set*) sebesar 6% (176 gambar). Pembagian ini bertujuan untuk memastikan model dapat dilatih, divalidasi, dan diuji secara optimal.



Gambar 4. 13 *Labeling* Data Perangkat

2. Pelabelan Data *Serial Number*

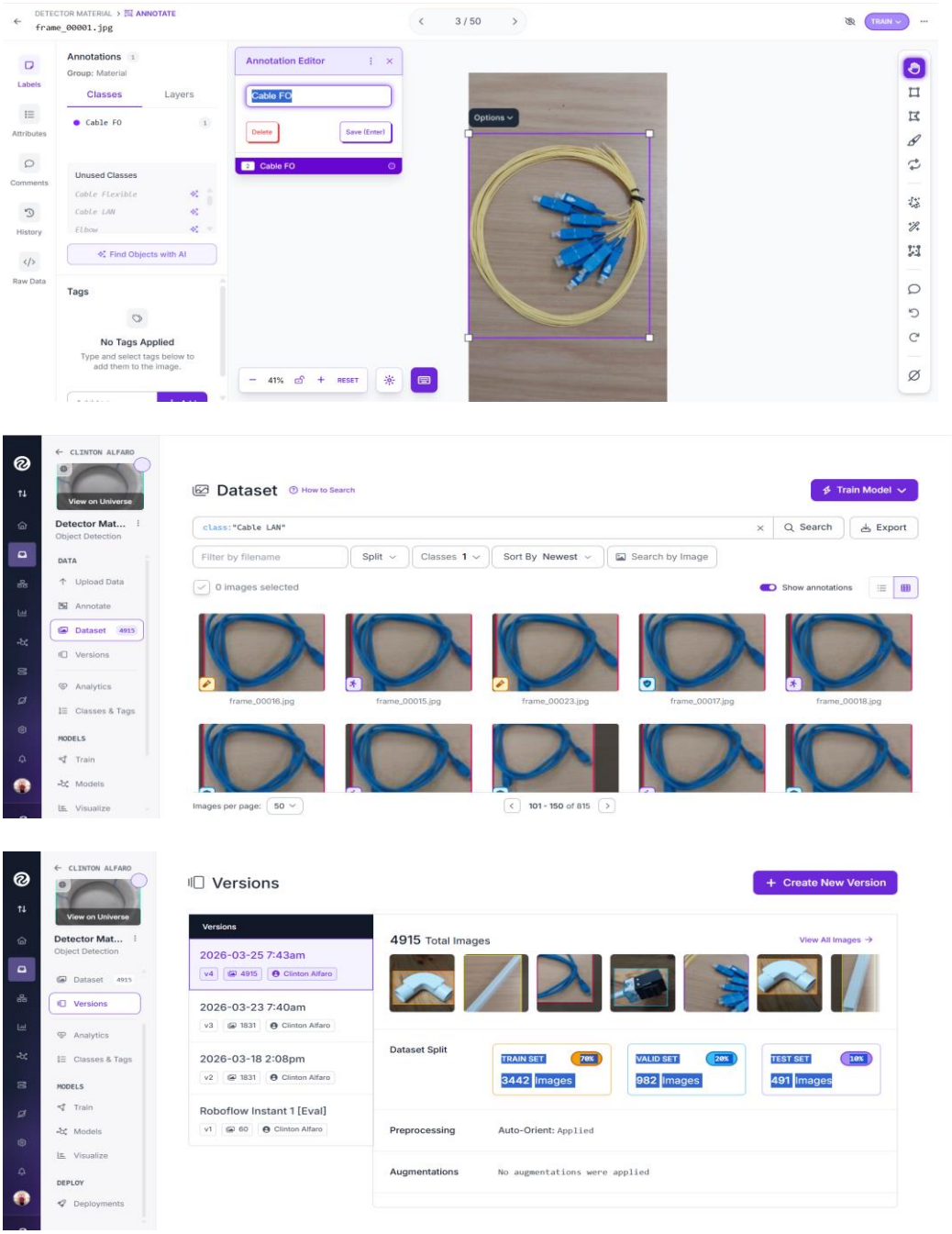
Proses pelabelan data serial number dilakukan untuk menandai area teks *serial number* pada setiap gambar sebagai bagian dari proses validasi inventaris. Total dataset yang digunakan sebanyak 3040 gambar, dengan pembagian data latih (*train set*) sebesar 72% (2186 gambar), data validasi (*valid set*) sebesar 18% (562 gambar), dan data uji (*test set*) sebesar 10% (291 gambar).



Gambar 4. 14 *Labeling* data *Serial Number*

3. Pelabelan Data Material

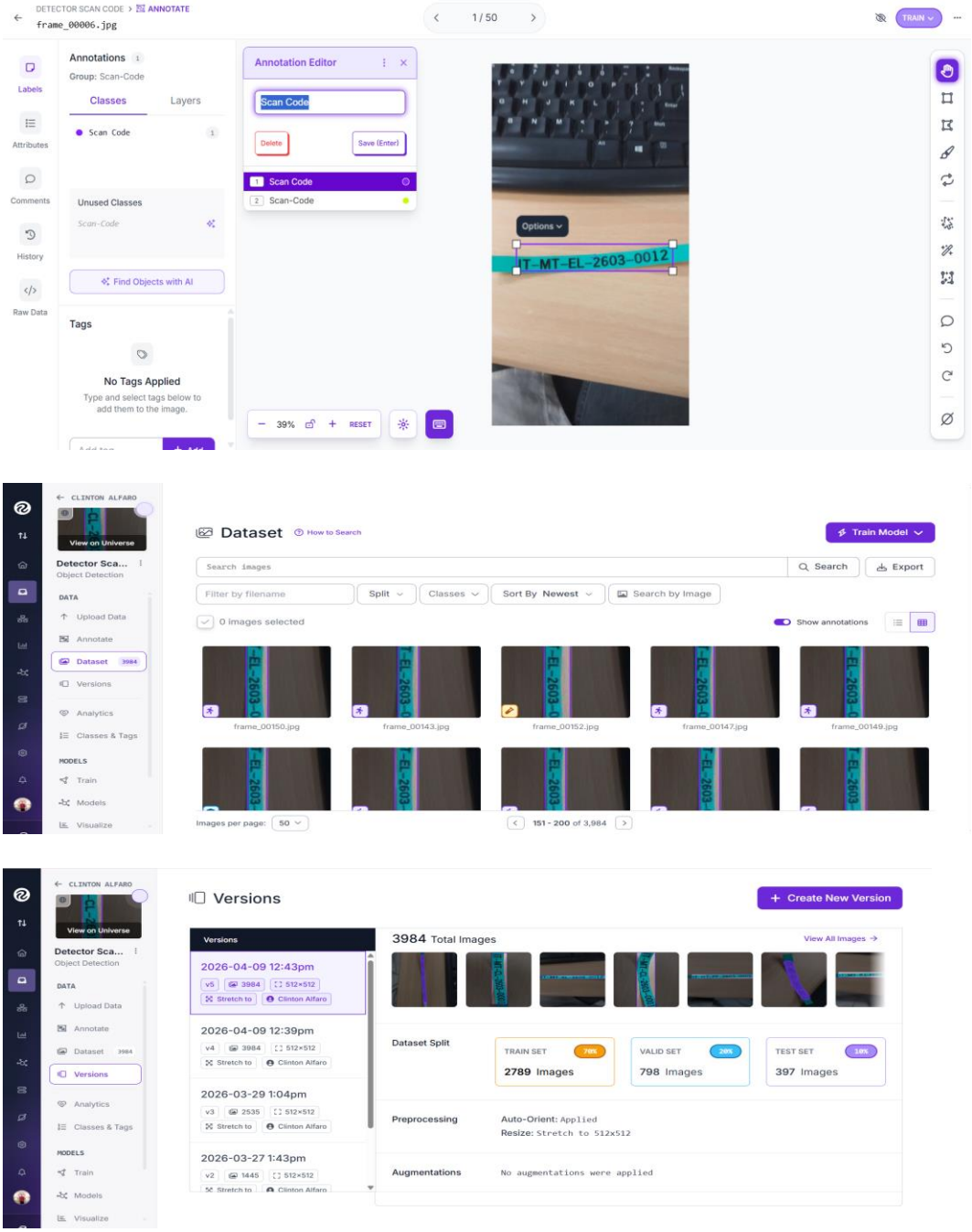
Proses pelabelan data material dilakukan untuk mengidentifikasi jenis material yang terdapat pada gambar sesuai dengan kategori yang telah ditentukan. Total dataset sebanyak 4915 gambar, dengan pembagian data latih (*train set*) sebesar 70% (3442 gambar), data validasi (*valid set*) sebesar 20% (982 gambar), dan data uji (*test set*) sebesar 10% (491 gambar).



Gambar 4. 15 *Labeling* Data Material

4. Pelabelan Data *Scan Code*

Proses pelabelan data *scan code* dilakukan untuk menandai area kode pada setiap gambar yang digunakan dalam proses identifikasi material. Total *dataset* yang digunakan sebanyak 3984 gambar, dengan pembagian data latih (*train set*) sebesar 70% (2789 gambar), data validasi (*valid set*) sebesar 20% (798 gambar), dan data uji (*test set*) sebesar 10% (397 gambar).

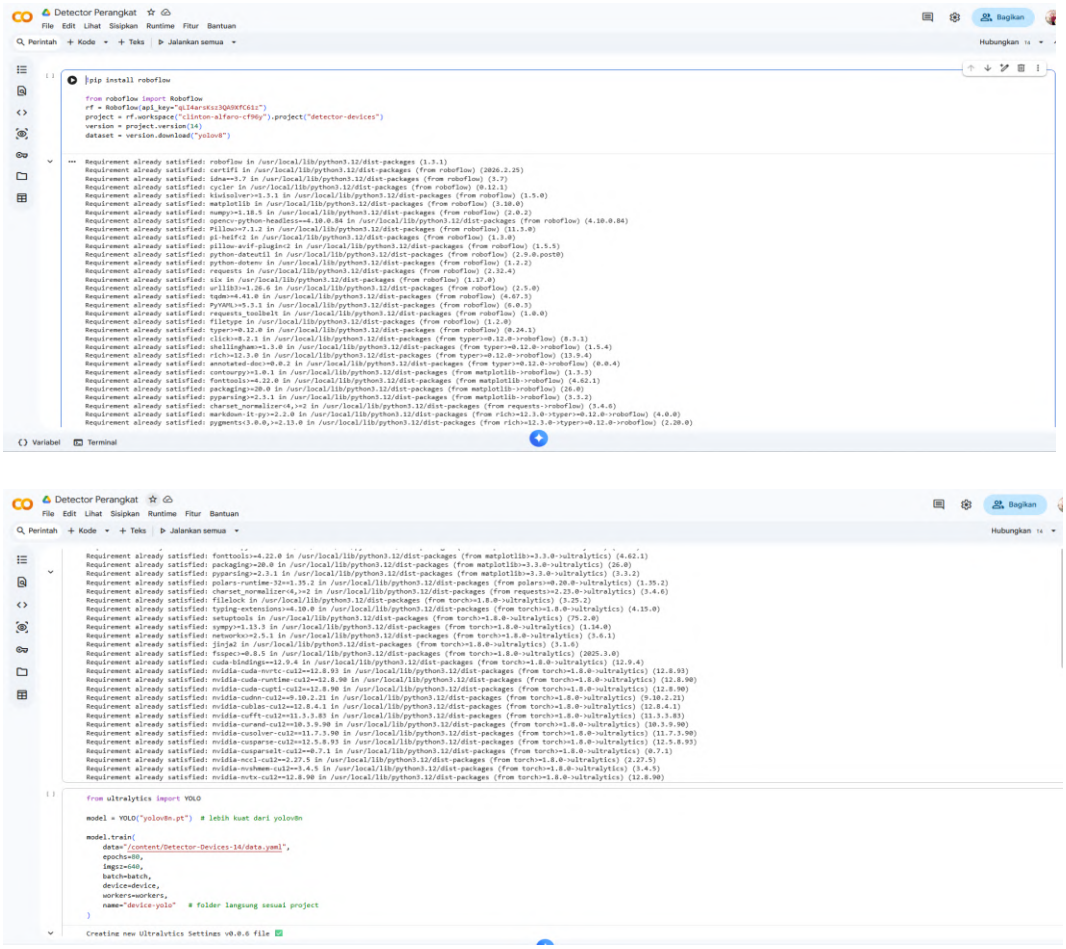


Gambar 4. 16 *Labeling* Data *Scan Code*

4.1.3.4 Implementasi Pelatihan Model *(Training)*

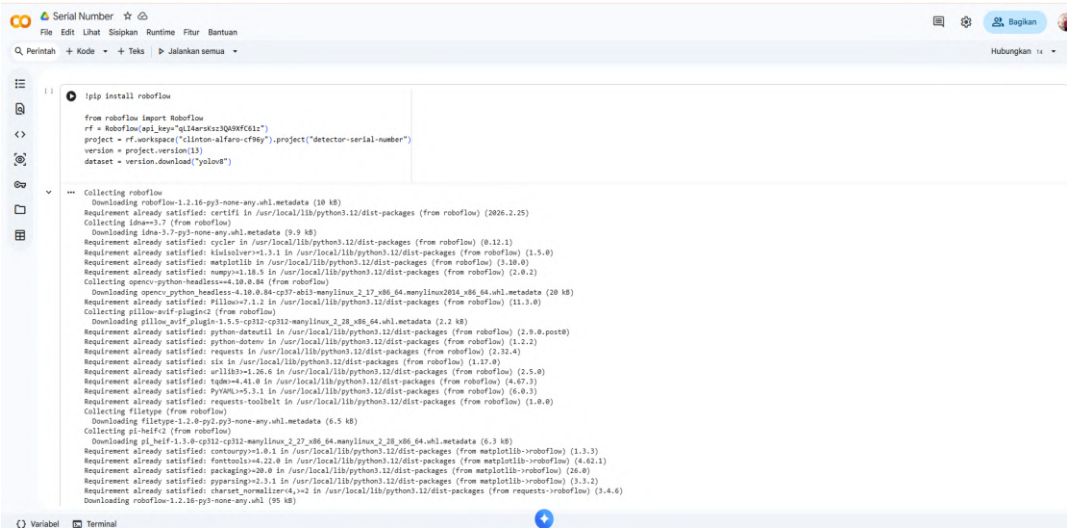
Tahap pelatihan model dilakukan untuk melatih model *AI* menggunakan *dataset* yang telah melalui proses pelabelan dimana proses *training* dilakukan menggunakan *Google Colab* dengan memanfaatkan *GPU NVIDIA T4* untuk mempercepat proses *training*. *Dataset* yang telah disiapkan diimpor menggunakan *library Roboflow* dan dilatih menggunakan model deteksi objek berbasis *YOLOv8*. Setiap kategori *dataset* dilatih secara terpisah agar model dapat bekerja secara optimal sesuai dengan jenis data yang digunakan.

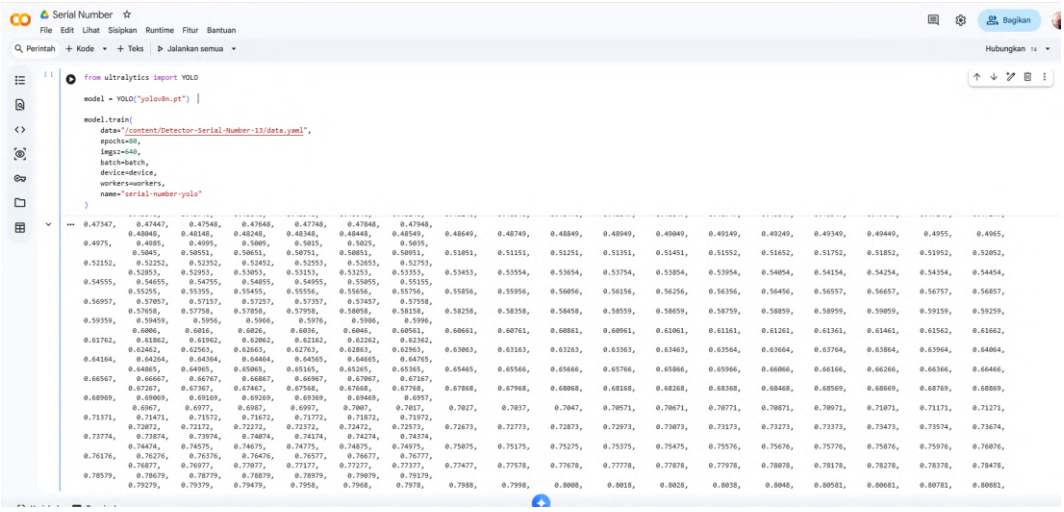
1. Pelatihan Model *Dataset* Perangkat



Gambar 4. 17 *Training* Model *Dataset* Perangkat

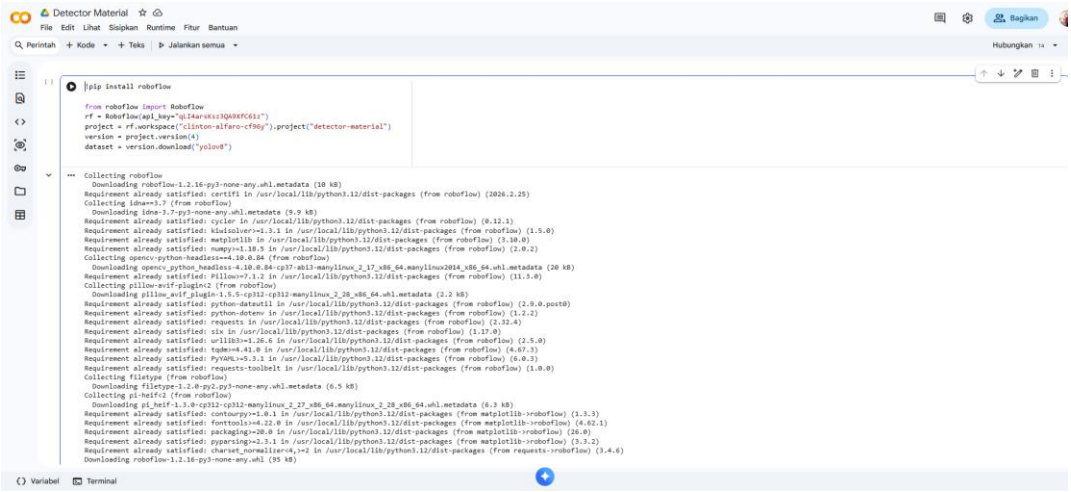
2. Pelatihan Model *Dataset Serial Number*

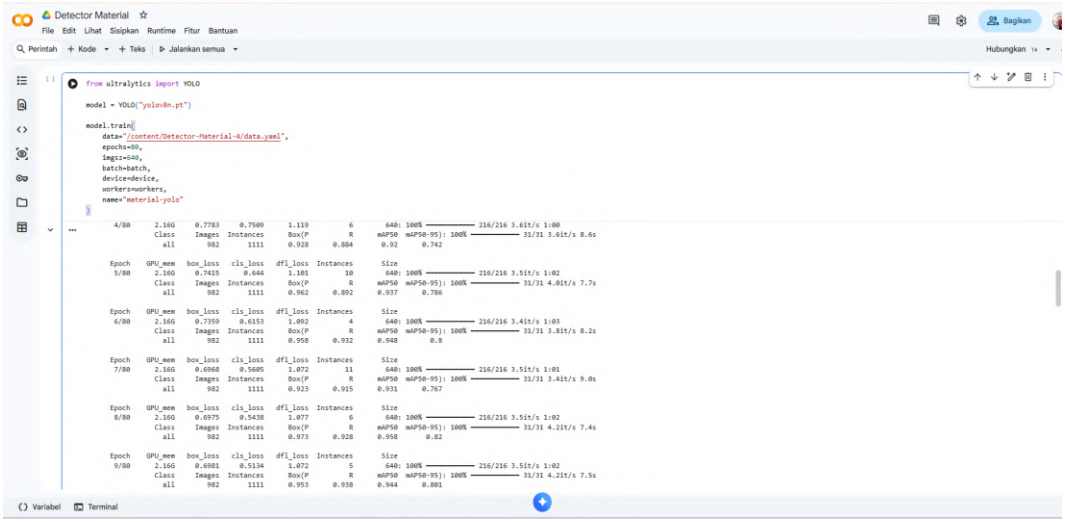




Gambar 4. 18 *Training* Model *Dataset Serial Number*

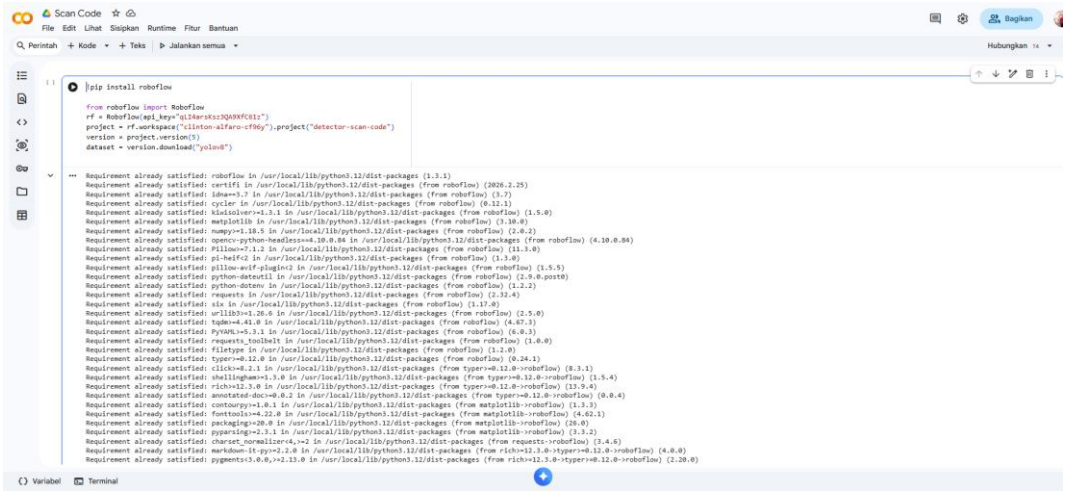
3. Pelatihan Model *Dataset* Material

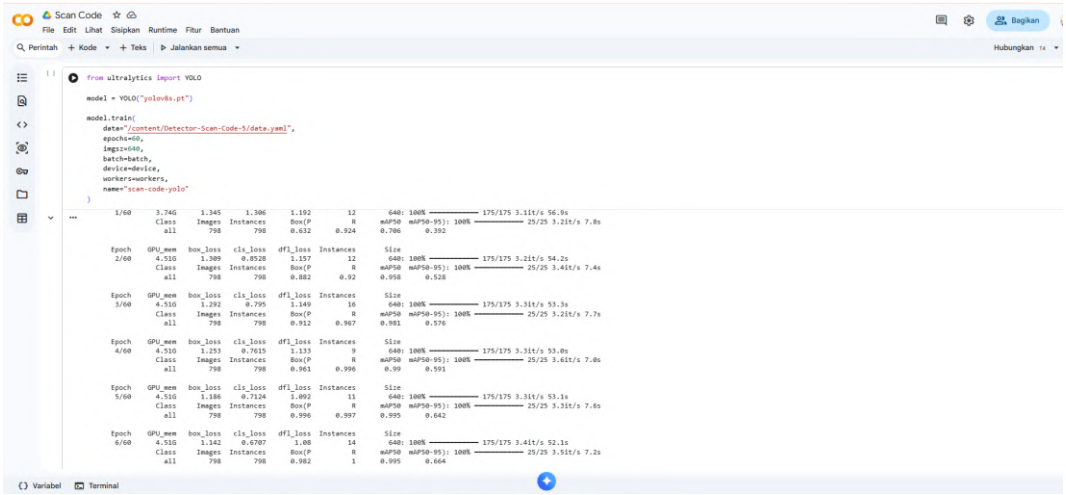




Gambar 4. 19 *Training* Model *Dataset* Material

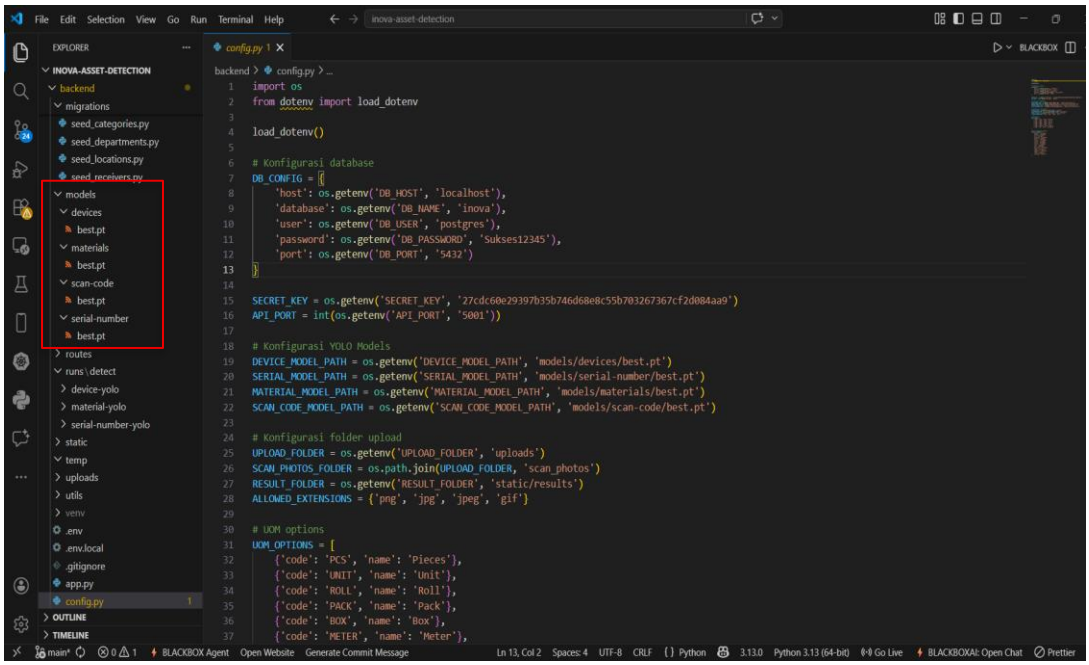
4. Pelatihan Model *Dataset Scan Code*





Gambar 4. 20 *Training* Model *Dataset Scan Code*

4.1.3.5 Implementasi *Deployment* Model



Gambar 4. 21 *Deployment* Model

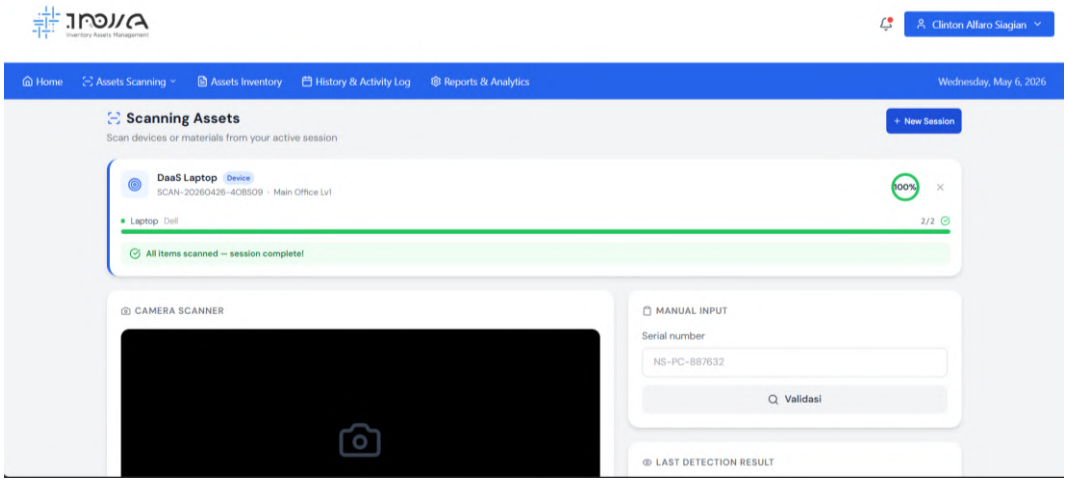
Pada tahap implementasi *deployment* model, seluruh model *AI* yang telah dilatih meliputi model deteksi perangkat, material, serial number, dan scan code diintegrasikan ke dalam sistem *backend* berbasis *Flask* dalam struktur project *INOVA*. Model hasil *training* disimpan dalam folder *models* pada *backend* dengan format file *.pt* dan dikonfigurasi melalui file konfigurasi menggunakan *environment variable* untuk memudahkan pengelolaan serta fleksibilitas dalam pengaturan *path* model. Sistem memanfaatkan *library YOLO* untuk melakukan proses inferensi terhadap gambar yang diunggah oleh pengguna, di mana hasil deteksi akan disimpan pada folder *runs/detect* dan diproses lebih lanjut untuk kebutuhan validasi inventaris. Selain itu, sistem juga mengatur folder *upload* dan hasil deteksi guna mendukung proses *scanning* secara *real-time* serta memastikan data yang dihasilkan tersimpan dengan terstruktur. Dengan integrasi ini, model *AI* dapat digunakan secara langsung oleh sistem melalui *API backend* sehingga mampu mendukung proses deteksi objek, identifikasi data, serta validasi inventaris secara otomatis, cepat, dan terintegrasi.

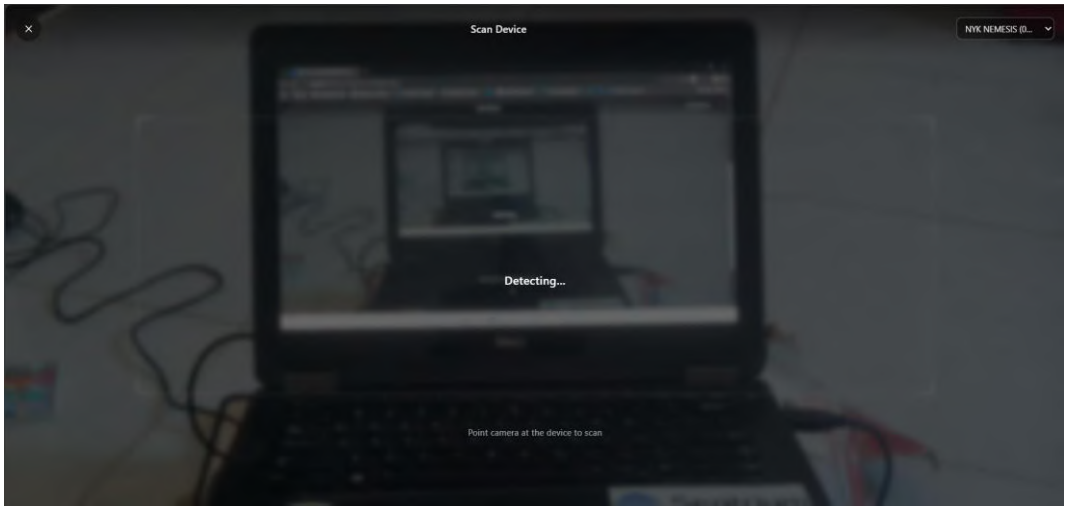
4.1.4 Proses Deteksi Objek dan Ekstraksi Data *YOLO* dan *OCR*

Pada proses ini menjelaskan alur kerja proses deteksi objek menggunakan *YOLO* dan ekstraksi teks menggunakan *OCR* secara menyeluruh, mulai dari aktivasi kamera oleh pengguna hingga data hasil *scanning* tersimpan dan siap diverifikasi. Proses ini merupakan inti dari sistem yang mengintegrasikan teknologi *computer vision* secara *real-time* untuk mendukung otomatisasi validasi inventaris aset *IT* di PT XYZ Indonesia.

1. Aktivasi Kamera dan Pengiriman *Frame* ke *Backend*

   Proses deteksi objek dan ekstraksi data pada sistem dilakukan dengan memanfaatkan model *YOLO* yang telah dilakukan *labeling* dan *training* dari *Google Colab* serta model *OCR* yang telah diintegrasikan pada *backend* sistem. Proses deteksi objek dan ekstraksi data dimulai ketika pengguna mengaktifkan kamera pada fitur *scanning* didalam sistem. Disini sistem secara otomatis menangkap *frame* gambar secara *real-time* melalui perangkat kamera yang terhubung. *Frame* yang berhasil ditangkap kemudian dikirimkan dari *frontend* ke *backend* sistem melalui *endpoint API* telah dikonfigurasi menggunakan *framework Flask*. Pengiriman data dilakukan secara berkelanjutan selama sesi *scanning* berlangsung, sehingga sistem mampu memproses setiap objek yang terdeteksi dalam kamera secara langsung tanpa memerlukan *input* manual dari pengguna.

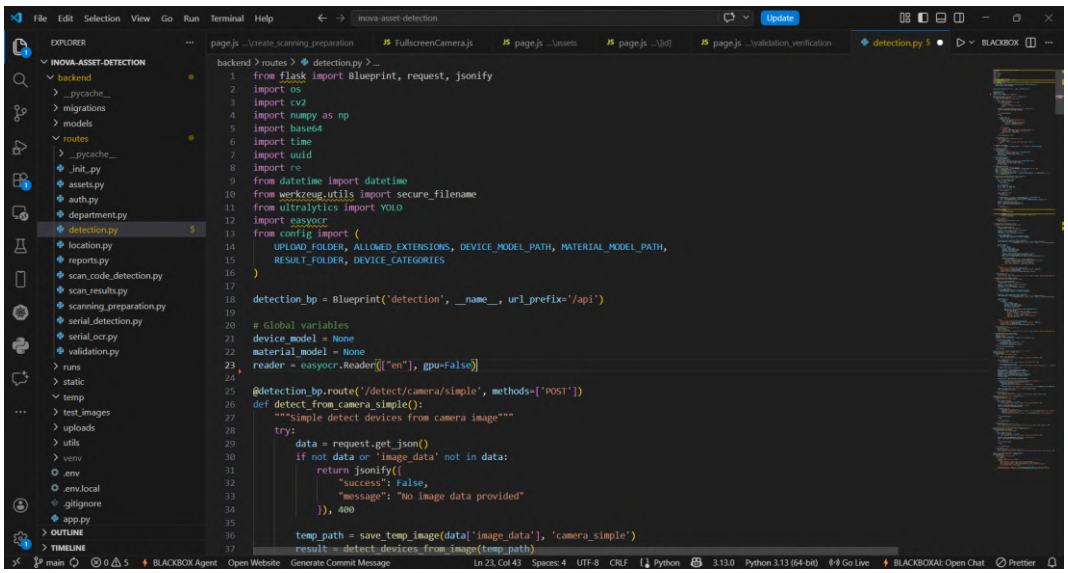


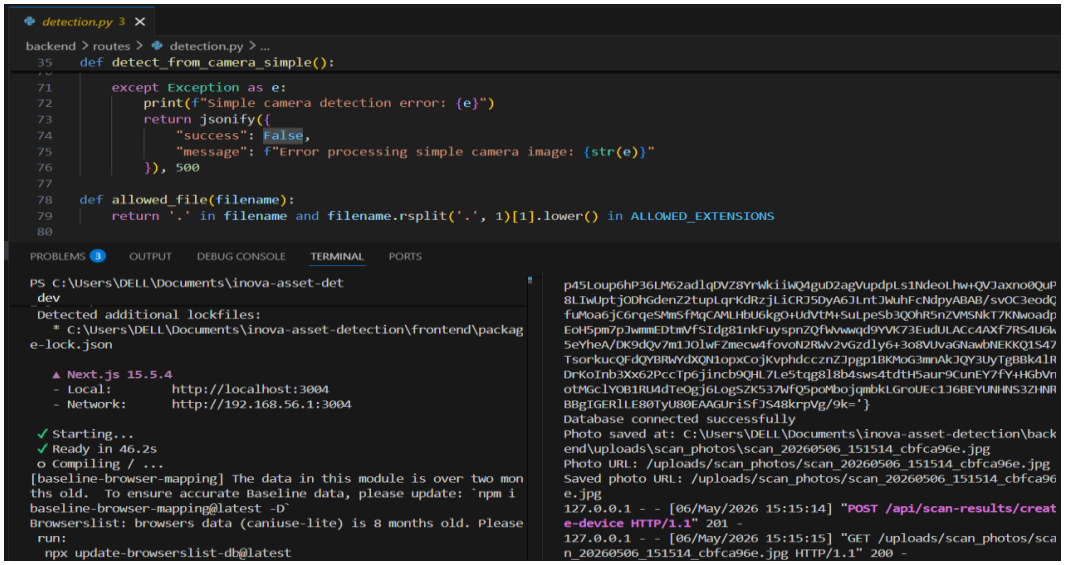


Gambar 4. 22 Tampilan fitur *scanning* dengan kamera aktif

2. Deteksi Objek (Perangkat/Material) oleh *YOLO*

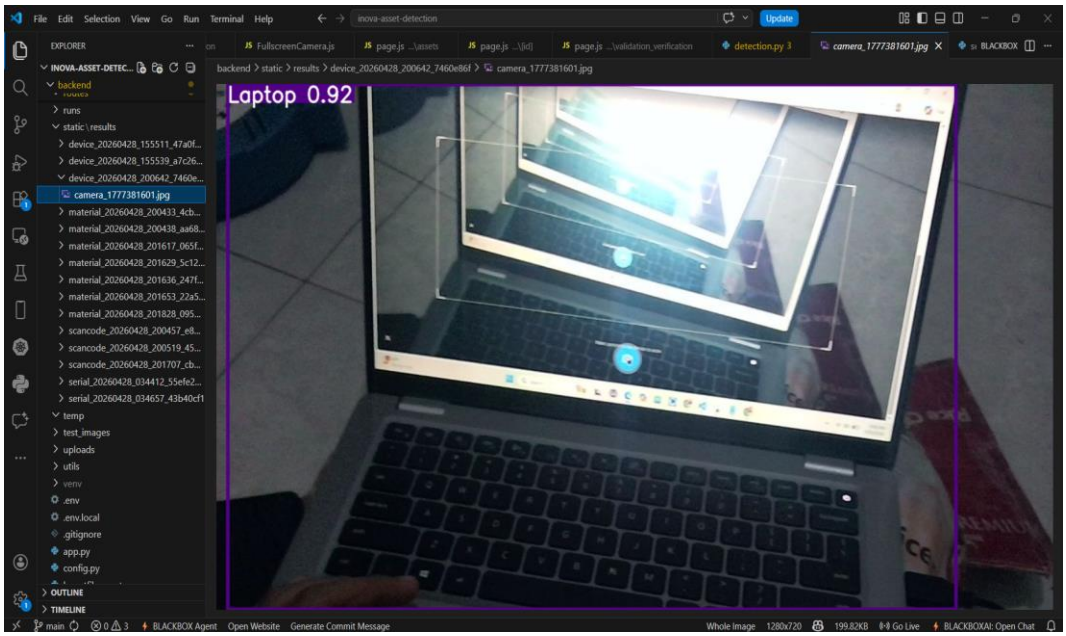
Setelah *frame* gambar tersebut diterima oleh *backend* sistem, proses deteksi objek kemudian akan dilakukan melalui modul pada *routes/detection.py* yang menjadi titik pemrosesan utama. Pada tahap ini, model *YOLOv8n* yang telah dilatih sebelumnya melakukan inferensi terhadap gambar masukan untuk mengidentifikasi objek yang ada di dalamnya. Model *YOLO* akan mengklasifikasikan objek ke dalam dua kategori utama, yaitu perangkat (*devices*) seperti *laptop*, *monitor*, *PC*, *keyboard*, dan perangkat *IT* lainnya, maupun material seperti *RJ45*, *Modular Jack*, *Cable LAN*, *Trunking*, dan komponen jaringan lainnya.





Gambar 4. 23 Pemrosesan deteksi objek

Proses deteksi kemudian akan menghasilkan *bounding box* yang menandai posisi dan area objek dalam gambar beserta nilai *confidence score* yang menunjukkan tingkat keyakinan model terhadap hasil deteksi. Hasil gambar beserta *bounding box* tersebut kemudian disimpan secara terstruktur ke dalam folder *static /results* dengan pengkategorian sesuai jenis objek yang terdeteksi, yaitu ke dalam *subfolder devices*, *material*, *serial_number*, atau *scan_code*, sehingga memudahkan pengelolaan dan penelusuran data hasil deteksi.



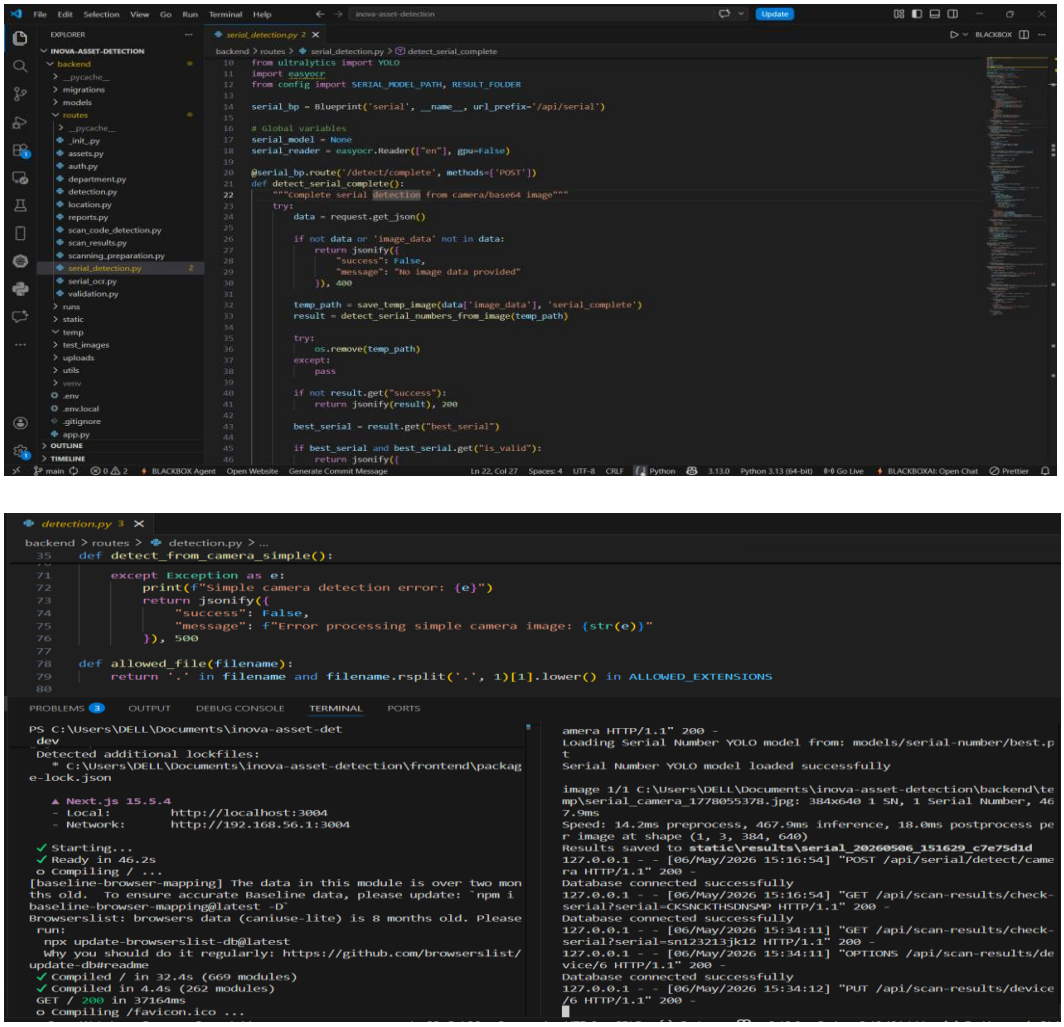
Gambar 4. 24 Hasil deteksi dan *Bounding Box* pada Perangkat

Gambar 4. 25 Hasil deteksi dan *Bounding Box* pada Material

3. Deteksi area *Serial Number* atau *Scan Code*

Setelah objek berhasil teridentifikasi, sistem akan secara otomatis melanjutkan deteksi ke tahap berikutnya sesuai dengan jenis objek yang berhasil terdeteksi. Apabila objek yang terdeteksi adalah perangkat *(devices)*, sistem akan melanjutkan proses *scanning* terhadap area *serial number* pada perangkat tersebut menggunakan modul pada *routes/serial_detection.py*.

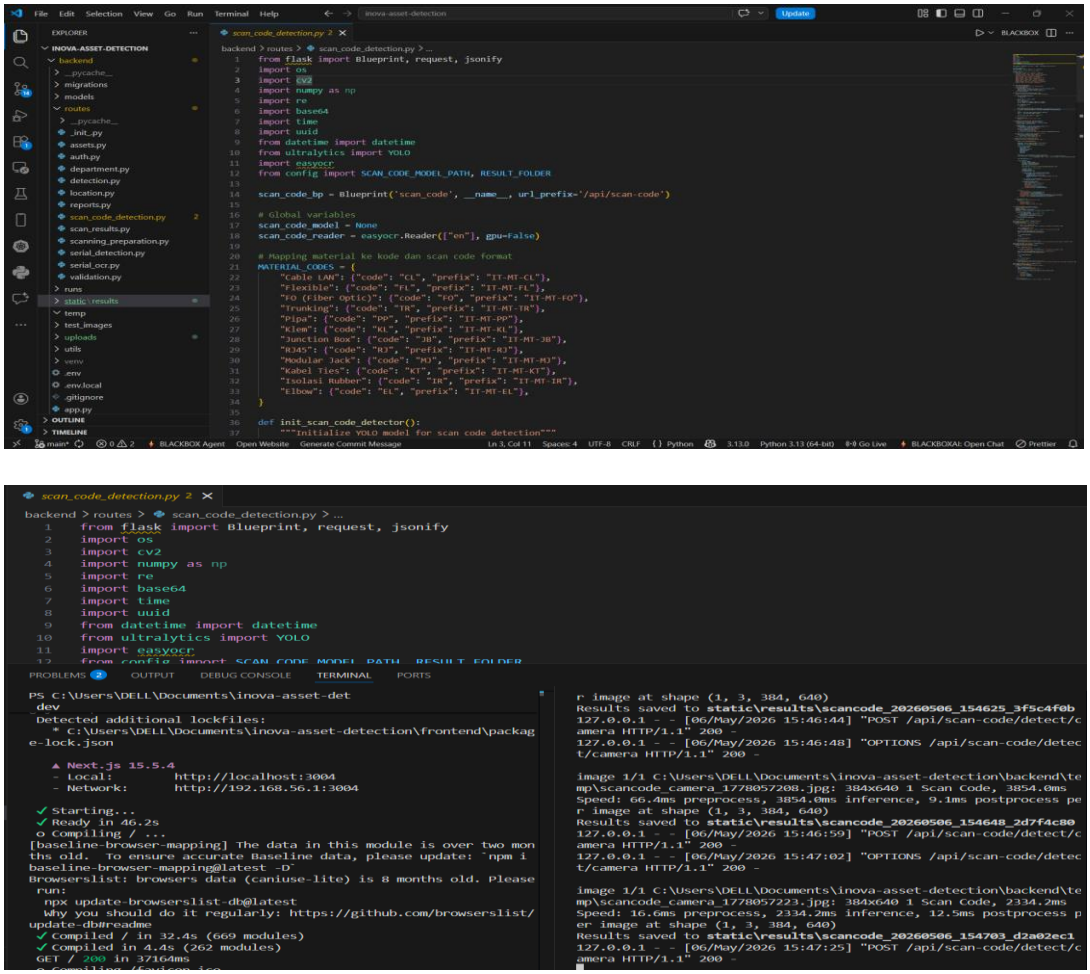
Gambar 4. 26 Deteksi *Serial Number*



Gambar 4. 27 Pemrosesan Deteksi *Serial Number*

Jika objek yang terdeteksi adalah material, sistem akan melanjutkan proses *scanning* terhadap area *scan code* material tersebut dengan menggunakan modul *scan_code_detection*.

Gambar 4. 28 Deteksi *Scan Code*



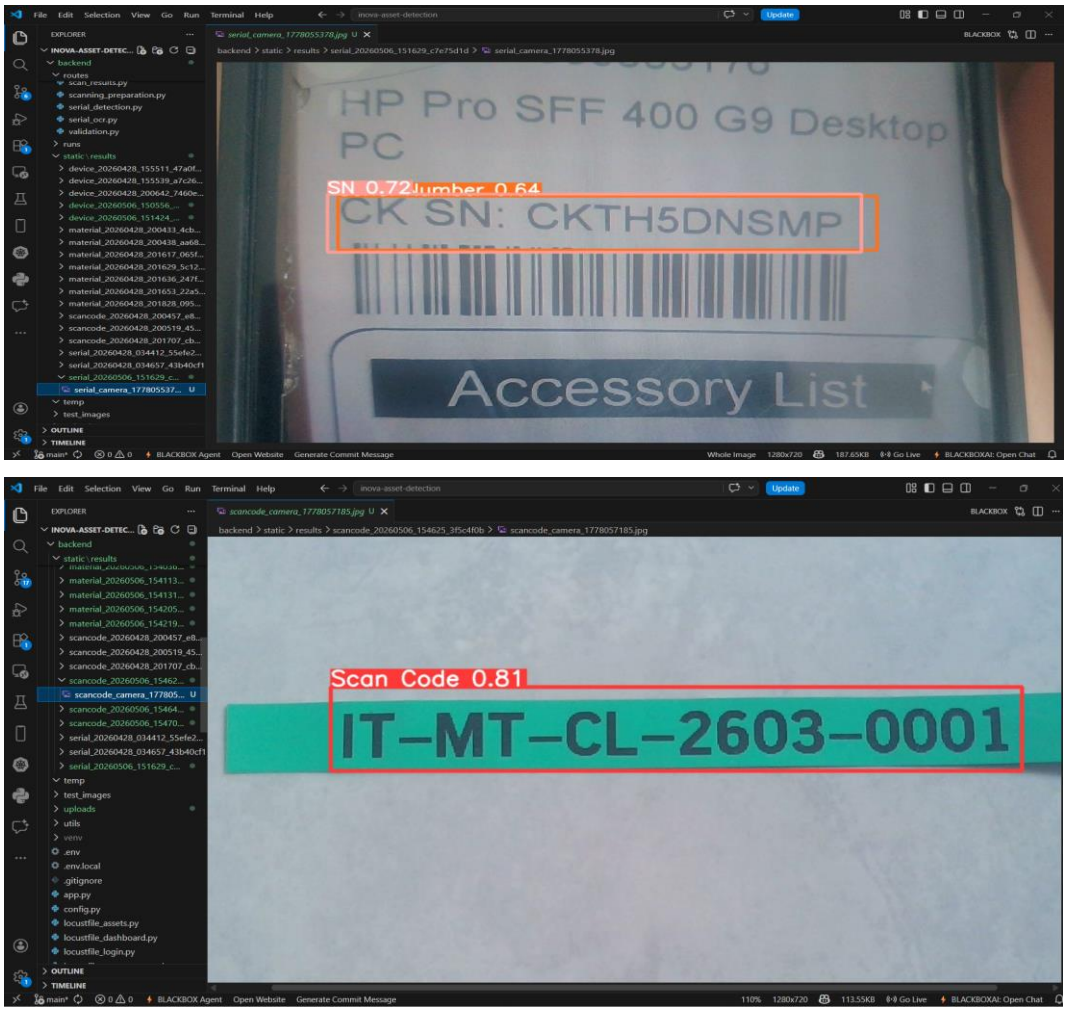
Gambar 4. 29 Pemrosesan deteksi *Scan Code*

Kedua modul tersebut menggunakan model *YOLO* terpisah yang telah dilakukan *labeling* dan *training* sebelumnya untuk mengenali dan melokalisasi area *serial number* maupun *scan code* pada masing-masing objek. Model kemudian menerapkan *bounding box* secara spesifik pada area yang mengandung informasi identifikasi tersebut, sehingga area teks yang relevan dapat diisolasi dengan tepat sebelum diserahkan ke proses ekstraksi selanjutnya.

4. Ekstraksi teks *Serial Number* atau *Scan Code*

Area *serial number* atau *scan code* yang telah berhasil dilokalisasi dan di-*crop* berdasarkan koordinat *bounding box* kemudian diproses oleh modul *serial_ocr* untuk dilakukan ekstraksi teks secara otomatis menggunakan *OCR* dimana *OCR* akan menganalisis citra area yang telah

diisolasi, melakukan segmentasi karakter, mengekstraksi fitur visual dari setiap karakter, dan merekonstruksi rangkaian teks yang terbaca sehingga menghasilkan nilai *serial number* atau *scan code* dalam bentuk teks digital. Hasil ekstraksi teks kemudian disimpan ke dalam folder *static/results* bersama dengan gambar hasil *scanning* sebagai bukti autentik bahwa proses pengecekan telah dilakukan.

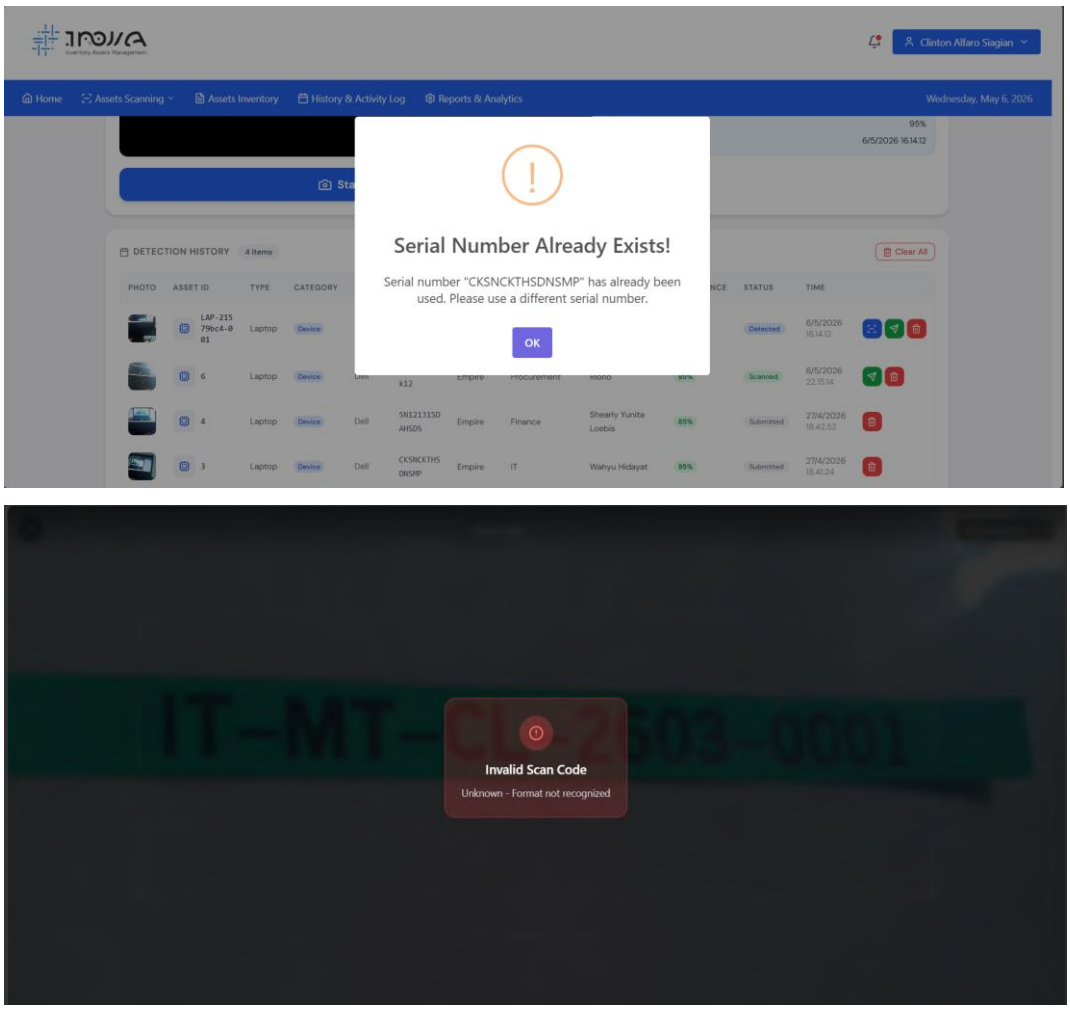


Gambar 4. 30 Hasil *Bounding Box* dan Ekstraksi *Serial Number* dan *Scan Code*

5. Validasi Format dan Penanganan Kondisi hasil *Scan*

Setelah teks *serial number* atau *scan code* berhasil diekstraksi, sistem secara otomatis melakukan proses validasi terhadap hasil tersebut sebelum data disimpan ke dalam basis data. Validasi dilakukan dalam dua aspek utama. Pertama, sistem memeriksa kesesuaian format teks yang terbaca dengan pola format *serial number* atau *scan code* yang telah

ditentukan. Apabila format teks tidak sesuai dengan ketentuan yang berlaku, sistem akan menampilkan peringatan (*alert*) yang menginformasikan bahwa format *serial number* atau *scan code* tidak sesuai sehingga pengguna dapat melakukan *scanning* ulang. Kedua, sistem memeriksa kemungkinan duplikasi data dengan mencocokkan teks hasil ekstraksi terhadap data yang telah tersimpan sebelumnya. Apabila *serial number* atau *scan code* yang sama telah tercatat sebelumnya dalam sesi atau data inventaris yang aktif, sistem akan menampilkan peringatan duplikasi agar pencatatan ganda dapat dihindari. Hanya data yang lolos kedua proses validasi tersebut yang akan diteruskan untuk ditampilkan dan disimpan ke sistem.



Gambar 4. 31 *Alert* duplikasi dan format data hasil ekstraksi tidak sesuai

4.1.5 Implementasi *UI*

Tahap implementasi *UI* ini yaitu implementasi antarmuka pengguna (*Users Interface / UI*) yang dirancang dalam sistem, meliputi tampilan halaman, interaksi pengguna dengan sistem, serta penjelasan fungsi dari setiap komponen yang tersedia. Implementasi ini bertujuan untuk memastikan sistem dapat digunakan dengan mudah dan memberikan pengalaman pengguna yang baik. Berikut merupakan hasil implementasi *UI* yang telah dikembangkan.

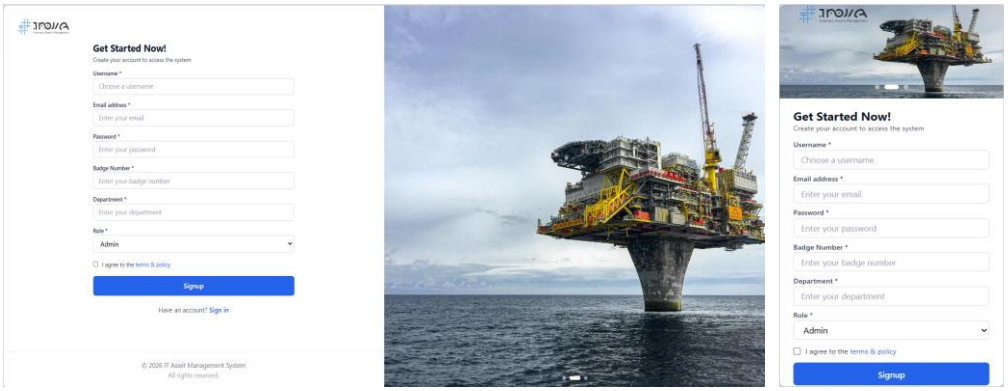
1. *Landing Page*

   Landing page merupakan halaman awal yang ditampilkan ketika pengguna pertama kali mengakses *website* maupun aplikasi. Halaman ini berfungsi sebagai pintu utama untuk memperkenalkan sistem kepada pengguna.

Gambar 4. 32 Implementasi *Landing Page*

2. *Register*

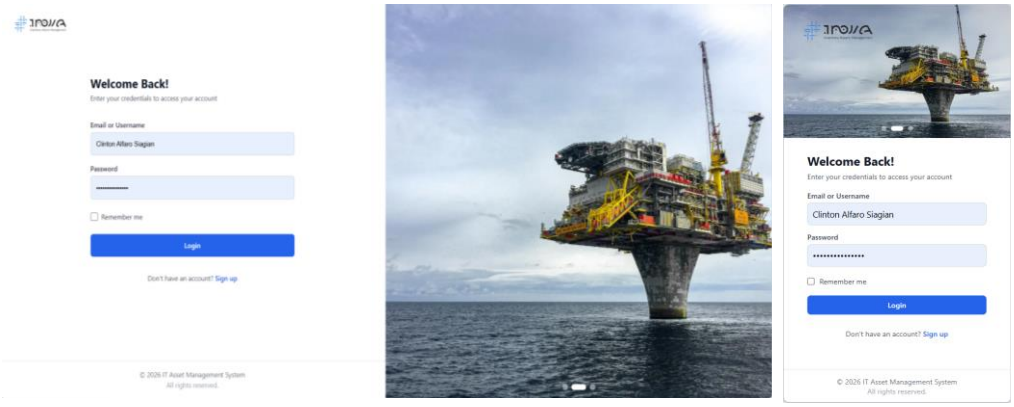
   Proses registrasi pada sistem *Inova* dilakukan oleh pengguna yang belum memiliki akun dengan mengisi form yang tersedia seperti *usersname*, *email*, *password*, *badge number*, *department*, dan *role*. Sistem nantinya akan melakukan validasi terhadap data yang dimasukkan, di mana jika *usersname*, *email*, atau *badge number* sudah pernah terdaftar, maka akan ditampilkan notifikasi gagal. Namun, apabila data yang dimasukkan valid dan belum pernah digunakan, maka sistem akan menyimpan data pengguna ke dalam *database* dan mengarahkan pengguna ke halaman *login* untuk melakukan autentikasi dan mengakses sistem.



Gambar 4. 33 Implementasi *Register*

3. *Login*

Proses *login* pada sistem *Inova* dapat dilakukan dengan memasukkan email atau *usersname* serta *password* yang telah terdaftar. Sistem akan melakukan validasi terhadap data yang dimasukkan dengan mencocokkannya dengan data yang ada di *database*, sehingga pengguna hanya dapat mengakses sistem apabila kombinasi *usersname/email* dan *password* sesuai. Jika pengguna memasukkan *password* yang salah, maka sistem akan menampilkan notifikasi kesalahan dan meminta pengguna untuk menginput ulang data dengan benar. Selain itu, apabila data akun yang dimasukkan belum pernah terdaftar di dalam sistem, maka akan ditampilkan *alert* bahwa akun tidak ditemukan, sehingga hanya pengguna yang valid yang dapat mengakses fitur sistem inventaris.



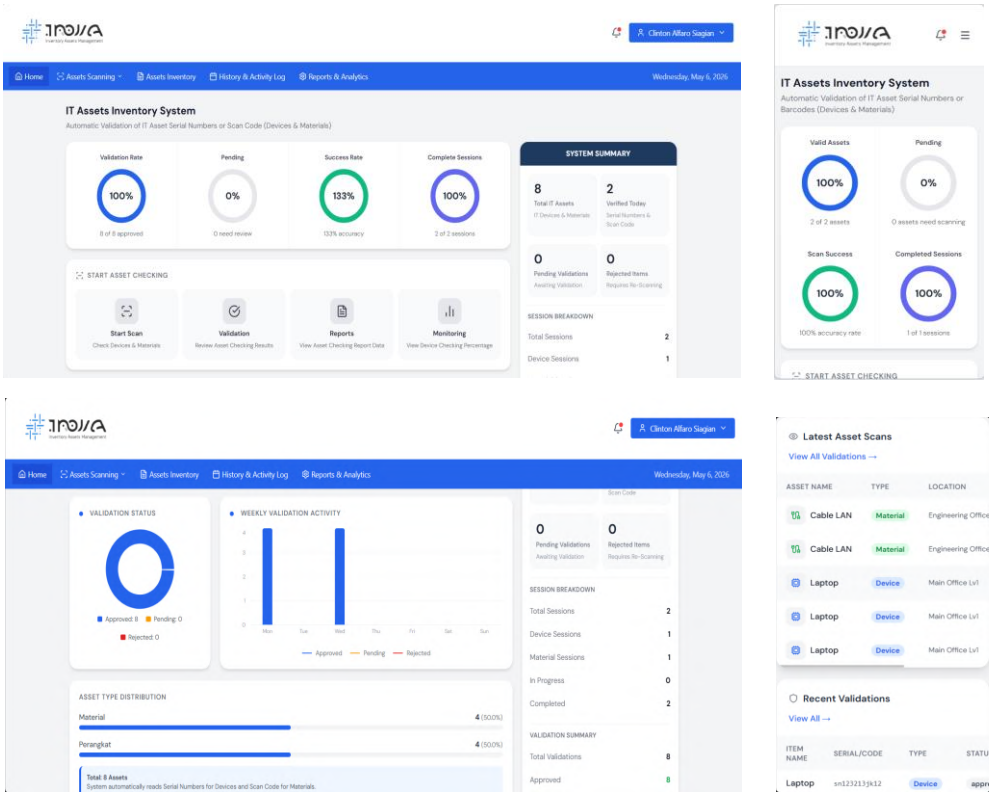
Gambar 4. 34 Implementasi *Login*

4. *Dashboard*

Pada halaman *dashboard* sistem *Inova*, pengguna akan melihat ringkasan informasi inventaris aset *IT* secara *real-time*. *Dashboard* menampilkan menu

utama seperti *Home*, *Assets Scanning*, *Assets Inventory*, *History & Activity Log*, *serta Reports & Analytics*. Selain itu, ditampilkan informasi sistem berupa *IT Assets Inventory System dengan fitur Automatic Validation of IT Asset Serial Numbers or Scan Code*. Pada *dashboard* juga ditampilkan statistik utama seperti jumlah aset valid, status pending, tingkat keberhasilan *scanning*, serta jumlah sesi pengecekan yang telah diselesaikan.

Sistem menyediakan tombol aksi seperti *Start Asset Checking*, *Start Scan*, *Checking Status*, *dan Checking Reports* untuk mempermudah proses pengecekan aset. Selain itu, terdapat grafik dan statistik seperti status validasi aset (valid, *pending*, *error*), aktivitas pengecekan harian, serta distribusi jenis aset. *Dashboard* juga menampilkan ringkasan aset seperti total aset *IT*, jumlah verifikasi hari ini, *pending inspection*, dan *error scanning*. Informasi lain seperti *session breakdown* dan *latest asset checking* juga ditampilkan untuk memantau aktivitas pengecekan. Seluruh data pada *dashboard* akan otomatis diperbarui berdasarkan aktivitas *scanning* dan *validasi*, sehingga membantu pengguna dalam melakukan *monitoring* inventaris secara lebih cepat, akurat, dan efisien.

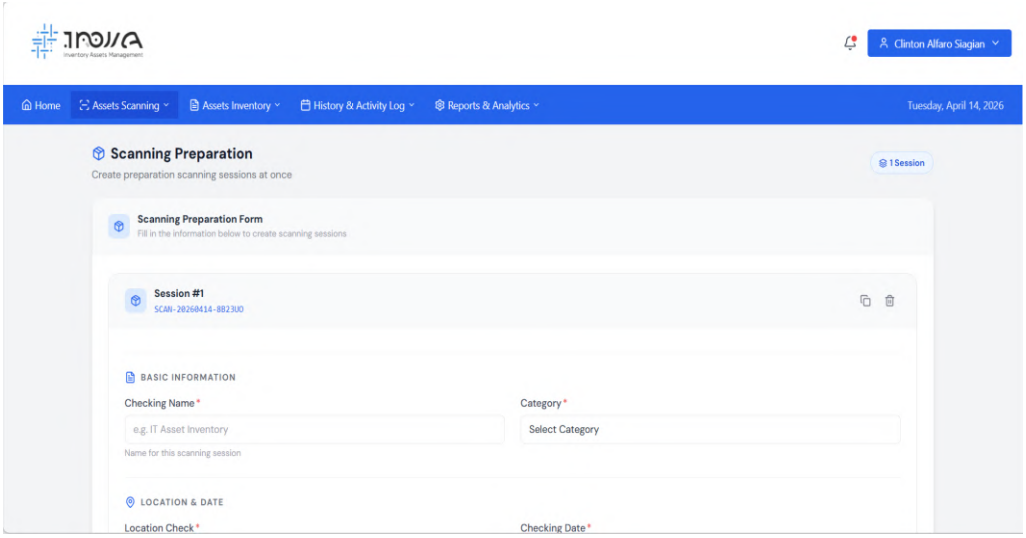


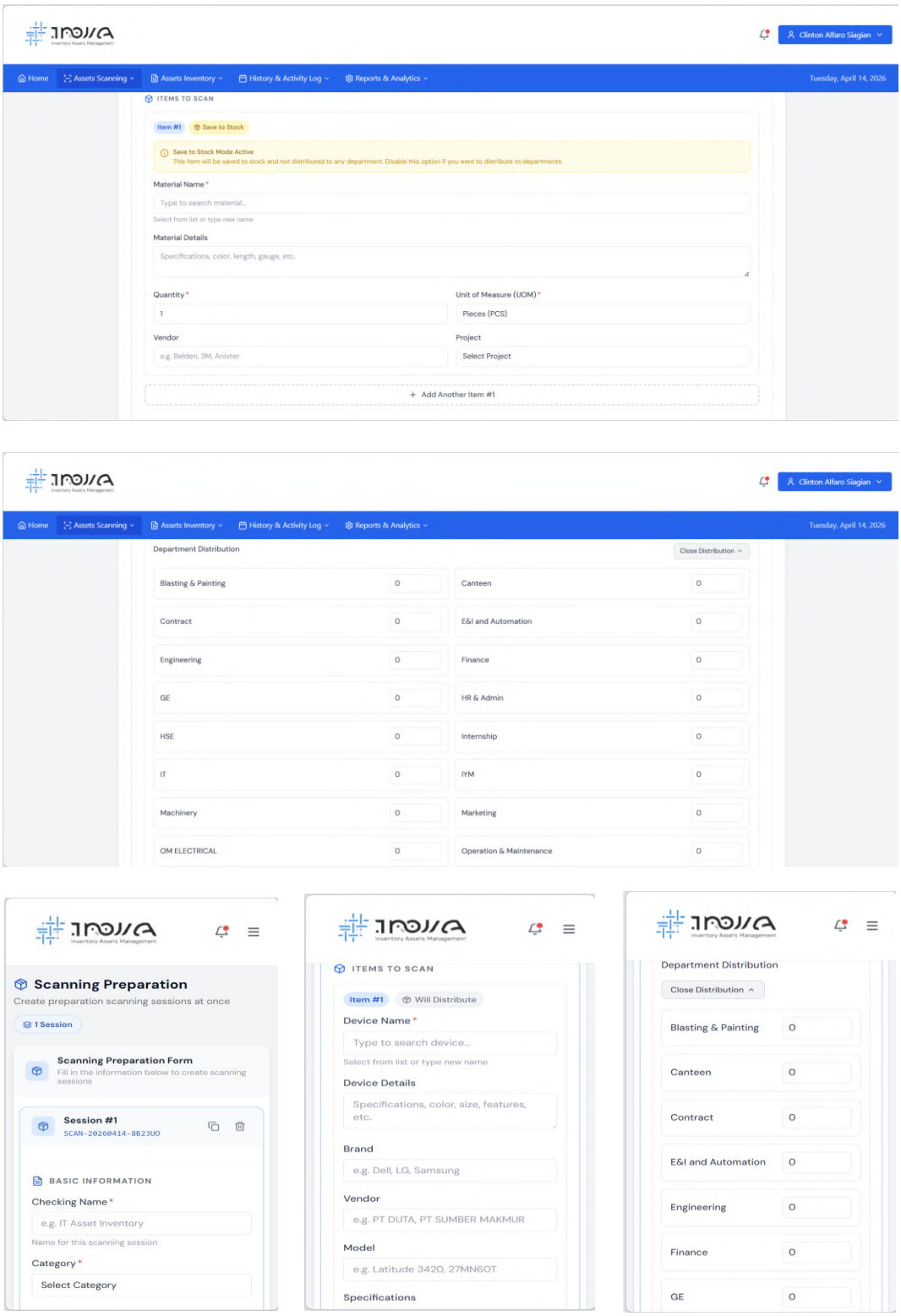
Gambar 4. 35 Implementasi *Dashboard*

5. *Create Preparation*

Sebelum melakukan proses *scanning* perangkat atau material, fitur ini hanya dapat diakses oleh *admin* dimana terlebih dahulu membuat *scanning preparation* melalui halaman *Create Preparation*. Pada halaman ini, *admin* mengisi *Basic Information* yang terdiri dari *checking name*, *category (devices* atau *materials)*, *location check*, serta *checking date*. Selanjutnya pada bagian *Items to Scan*, *admin* mengisi detail item sesuai kategori yang dipilih. Untuk kategori *devices*, *users* mengisi *device name*, *brand*, *vendor*, *model*, *specifications*, *quantity*, dan *project*. Sedangkan untuk kategori *materials*, *users* mengisi *material name*, *quantity*, *unit of measure (UOM)*, *vendor*, dan *project*. Selain itu, *admin* juga dapat menambahkan *remarks* sebagai informasi tambahan terkait sesi *scanning*.

Pada proses selanjutnya, *admin* memilih metode penyimpanan yaitu *save to stock* atau *distribute directly*. Jika memilih *save to stock*, maka item akan langsung disimpan ke stok tanpa distribusi ke *departement*. Sedangkan jika memilih *distribute directly*, *users* perlu mengatur *department distribution* sesuai jumlah *quantity*, kemudian menentukan *receiver* untuk setiap item pada bagian *receiver assignment*. Sistem akan menampilkan ringkasan distribusi untuk memastikan jumlah sesuai. Setelah seluruh data terisi dengan benar, *admin* dapat menekan tombol *Create* untuk menyimpan data sebagai *session scanning* yang akan digunakan pada proses *scanning* berikutnya.

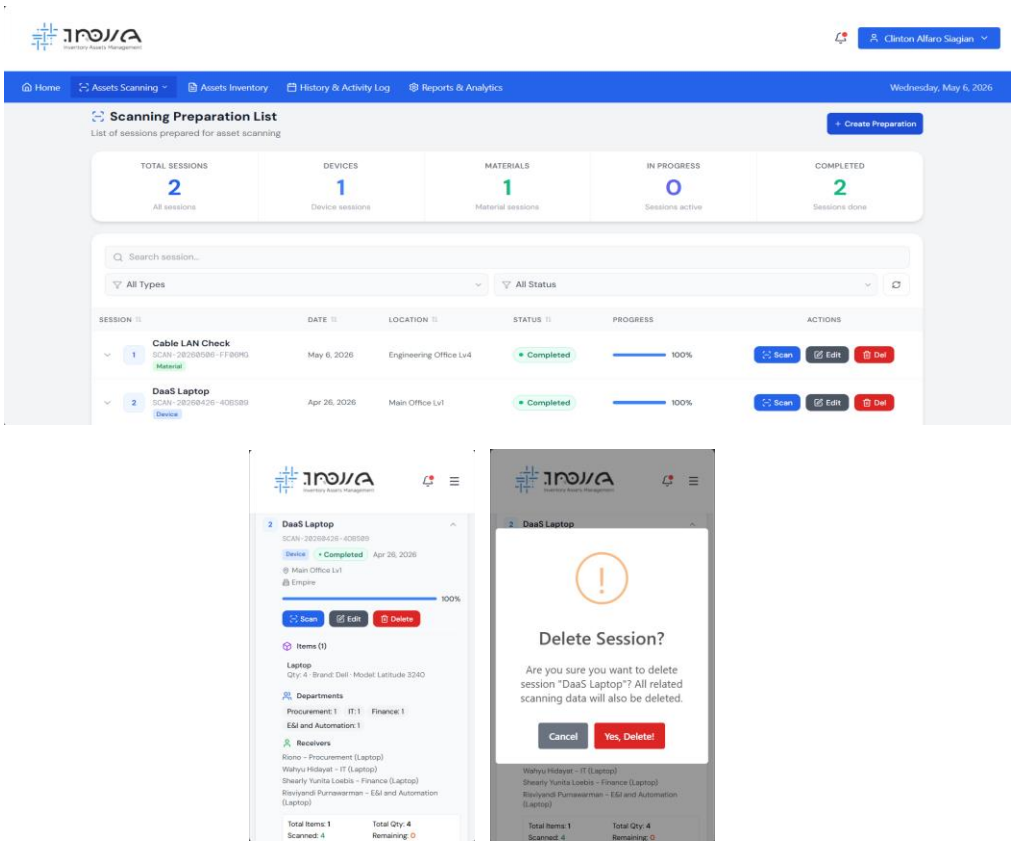




Gambar 4. 36 Implementasi *Create Preparation*

6. *Scanning Preparation List*

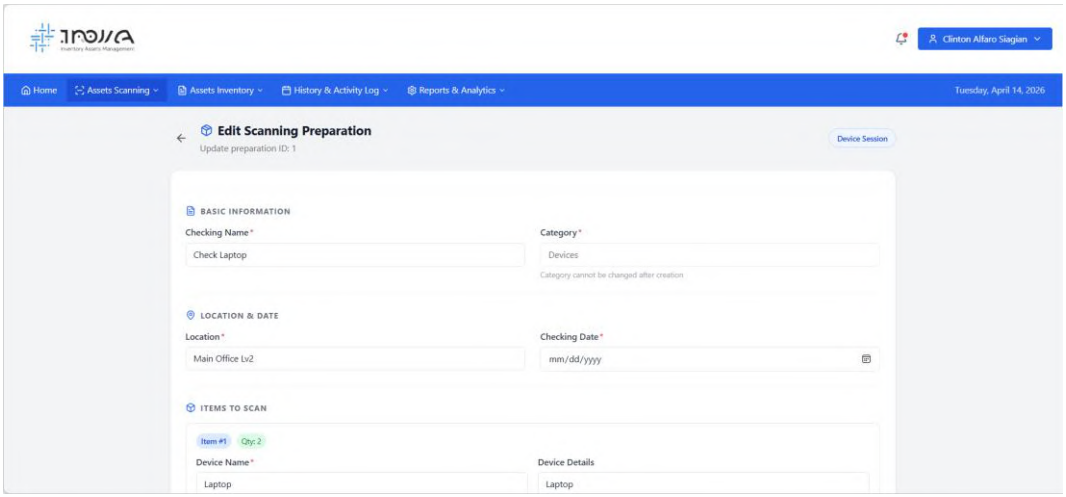
Setelah selesai membuat *scanning preparation*, *admin* dapat melihat data *scanning preapration* dihalaman *Scanning Preparation List* yang dimana menampilkan daftar seluruh sesi *scanning* yang telah dibuat. Pada halaman ini, *users* dapat melihat ringkasan data seperti total *sessions*, *devices sessions*, *materials sessions*, status *in progress*, dan *completed*. Setiap data *preparation* ditampilkan dalam bentuk list yang berisi informasi seperti nama sesi, kode *scanning*, kategori, tanggal, lokasi, *project*, serta status dan *progres scanning*. *Admin* dapat melihat detail lengkap dengan menekan tombol *dropdown*, yang menampilkan informasi item, jumlah, distribusi *departement*, serta *receiver assignment* atau data *stock* jika data *sessionnya* tidak mendistribusikan *department*. Selain itu, *admin* dapat melakukan aksi seperti *Scan* untuk memulai proses *scanning*, Edit untuk mengubah data *preparation*, dan *Delete* untuk menghapus data. Halaman ini juga dilengkapi fitur pencarian dan filter berdasarkan tipe dan status untuk memudahkan pengelolaan data.

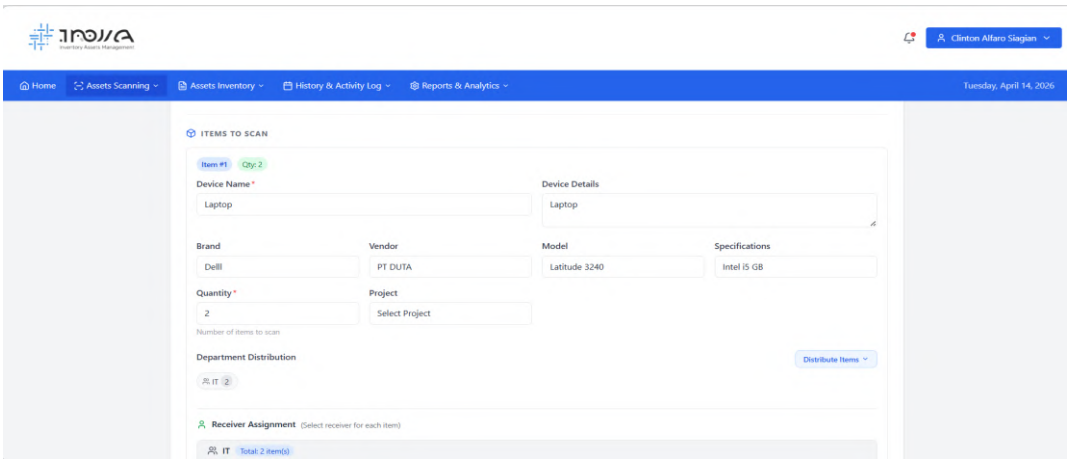


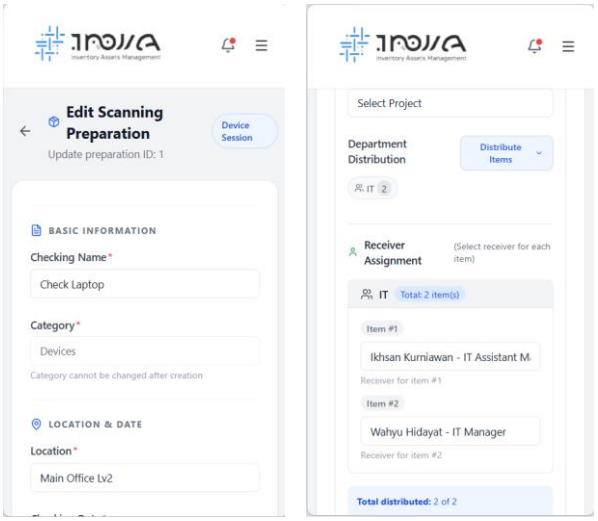
Gambar 4. 37 Implementasi *Scanning Preparation List*

7. Edit *Scanning Preparation*

Pada halaman *Edit Scanning Preparation users* dapat melakukan edit data *scanning preparation* dengan memilih data dari *Scanning Preparation List*. Pada halaman ini, *admin* memperbarui informasi pada bagian *Basic Information* seperti *checking name*, *location*, dan *checking date*, sedangkan *category* tidak dapat diubah. Selanjutnya pada bagian *Items to Scan*, *users* akan mengedit detail item sesuai kategori, seperti *device name*, *brand*, *vendor*, *model*, *specifications*, *quantity*, dan *project* untuk perangkat. *Admin* juga diminta untuk menyesuaikan metode penyimpanan, baik *save to stock* maupun *distribute items*, termasuk mengatur kembali *department distribution* dan *receiver assignment* sesuai kebutuhan. Setelah seluruh perubahan dilakukan, *admin* menekan tombol *Update Session* untuk menyimpan data terbaru, sehingga *session scanning* tetap terbarui dan sesuai dengan kondisi aktual sebelum proses *scanning* dilakukan.





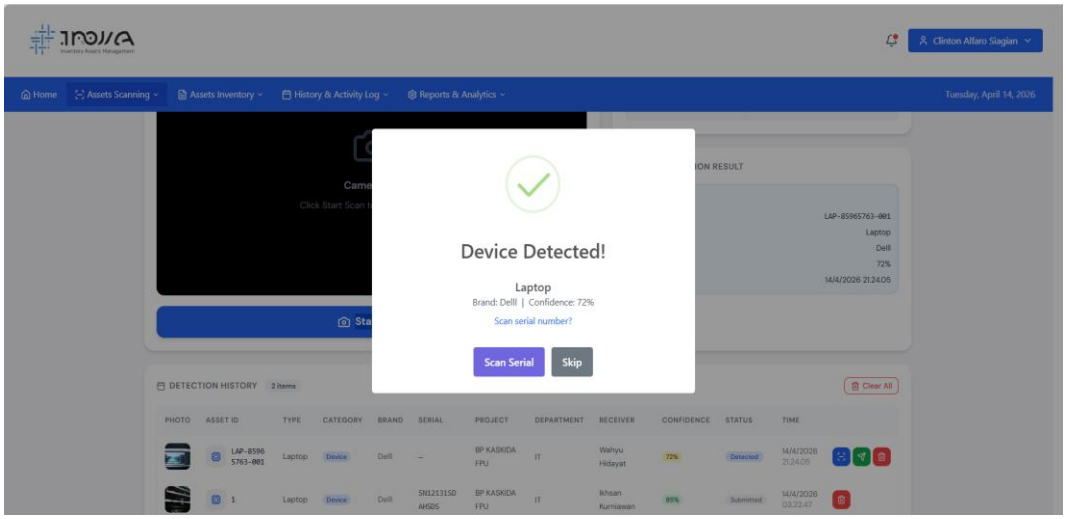


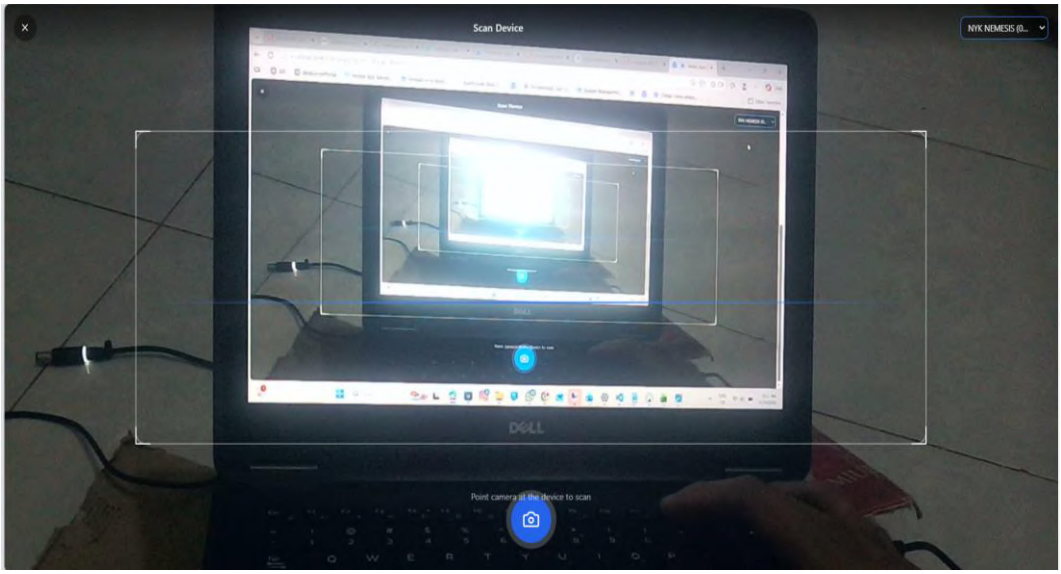
Gambar 4. 38 Implementasi Edit *Scanning Preparation*

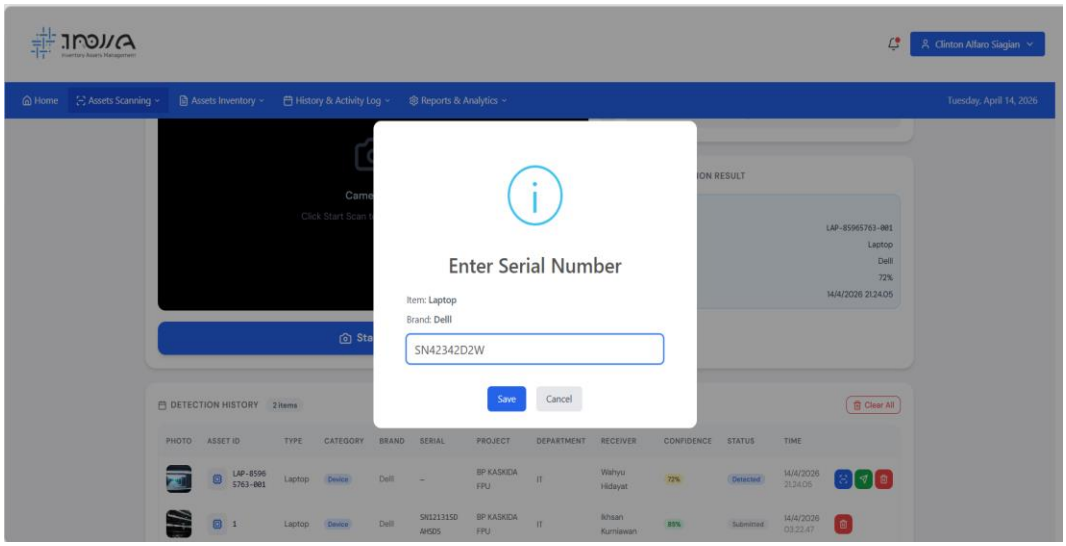
8. *Scanning Assets*

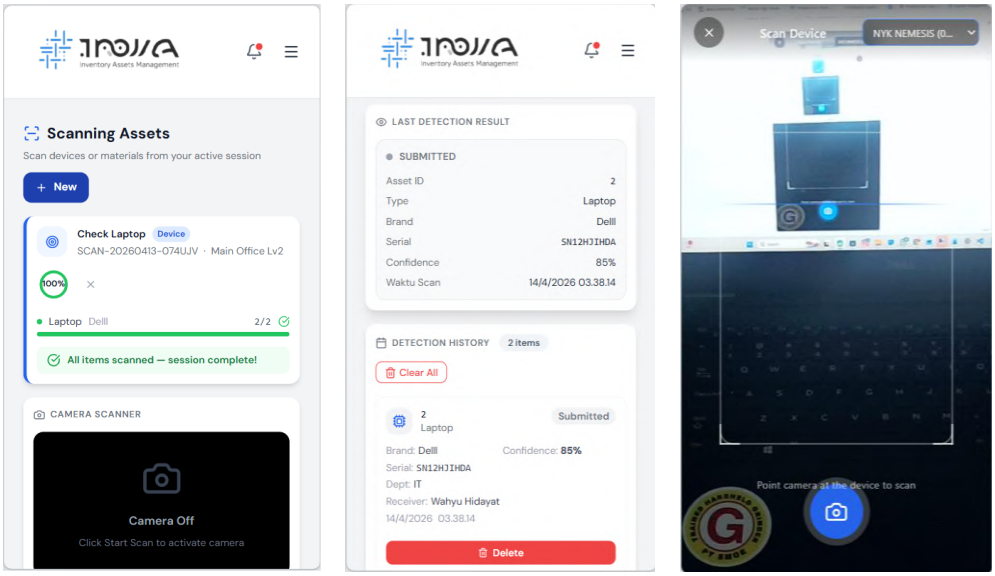
Setelah *scanning preparation* dibuat, *admin* dapat melakukan proses *scanning* pada halaman *Scanning Assets* dengan memilih *session* yang telah dibuat sebelumnya. Pada halaman ini, sistem akan menampilkan detail *session* seperti nama, lokasi, kategori, serta jumlah item yang harus di *scan*. *Admin* memulai proses dengan menekan tombol *Start Scan* untuk mengaktifkan kamera, kemudian mengarahkan kamera ke objek perangkat atau material. Sistem akan melakukan deteksi objek menggunakan model *AI*, dimana jika objek yang terdeteksi tidak sesuai dengan data pada *session* maka akan muncul notifikasi ketidaksesuaian, sedangkan jika sesuai maka data item akan ditampilkan dan disimpan ke dalam *detection history*.

Selanjutnya *users* melanjutkan proses dengan melakukan *scanning serial number* atau *scan code* melalui kamera, di mana sistem akan melakukan deteksi dan ekstraksi teks. Jika hasil tidak sesuai format atau sudah pernah digunakan, maka akan muncul notifikasi kesalahan, namun jika valid maka data akan disimpan ke dalam tabel hasil *scanning*. Selain itu, *admin* juga dapat melakukan *manual input* baik untuk objek maupun *serial number* atau *scan code* apabila tidak menggunakan kamera. Seluruh hasil *scanning* akan tercatat lengkap beserta informasi seperti tipe aset, *brand*, *project*, *departement*, *receiver*, *confidence*, dan waktu *scan*, sehingga proses validasi inventaris dapat dilakukan secara akurat dan terstruktur.


Clinton Alfaro Siagian
Home
Assets Scanning
Assets Inventory
History & Activity Log
Reports & Analytics
Tuesday, April 14, 2026
Device Detected!
Laptop
Brand: Dell | Confidence: 72%
Scan serial number?
Scan Serial
Skip
DETECTION HISTORY
Clear All


Scan Device
Point camera at the device to scan

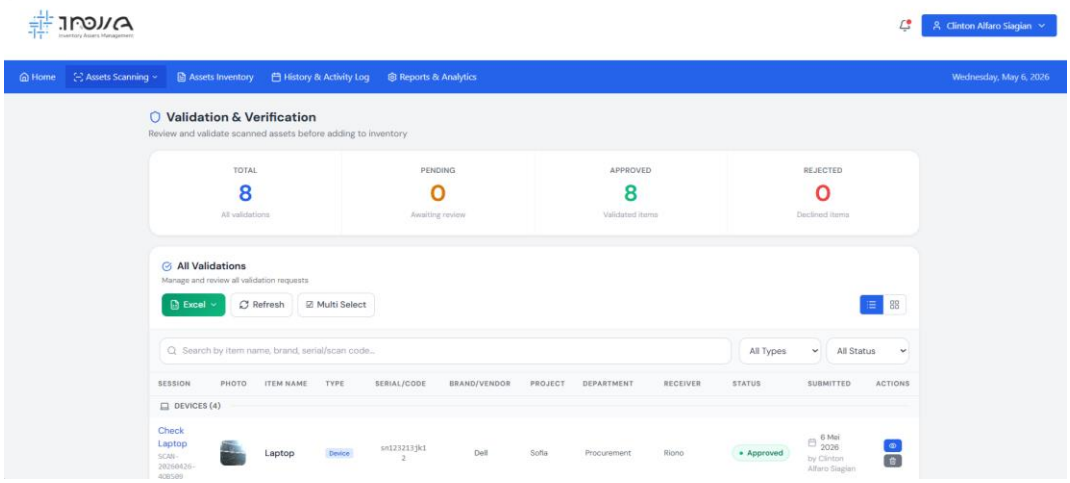

Clinton Alfaro Siagian
Home
Assets Scanning
Assets Inventory
History & Activity Log
Reports & Analytics
Tuesday, April 14, 2026
Enter Serial Number
Item: Laptop
Brand: Dell
SN42342D2W
Save
Cancel
DETECTION HISTORY
Clear All

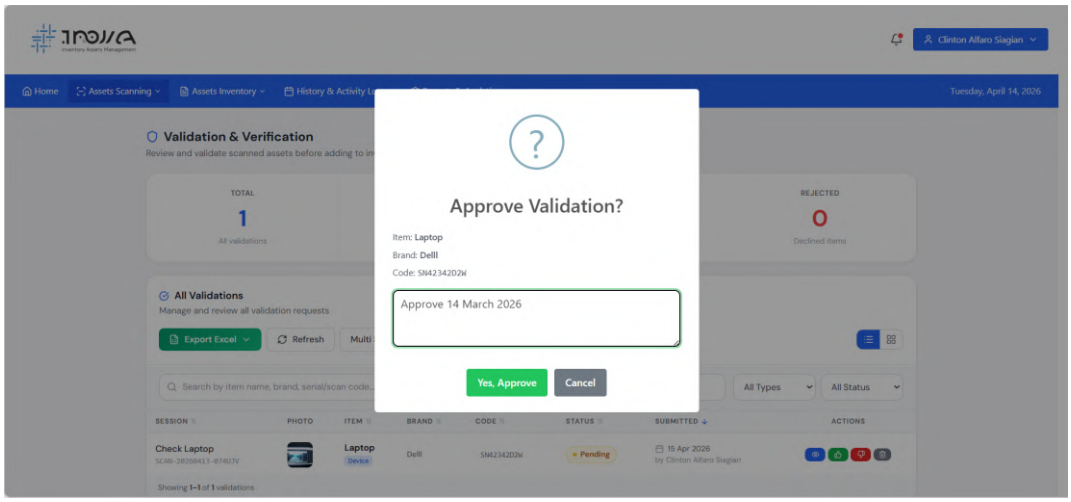


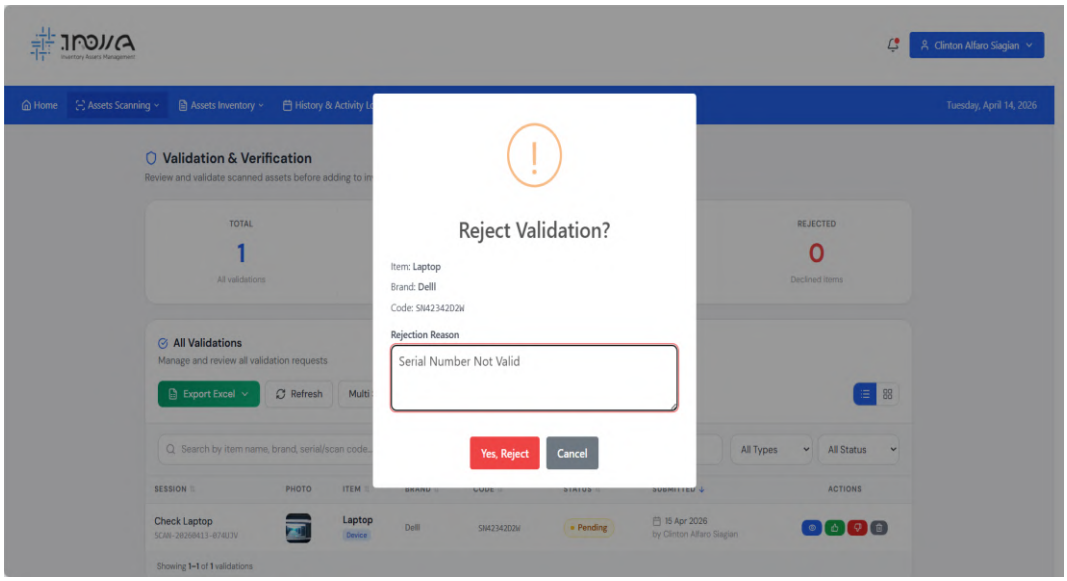
Gambar 4. 39 Implementasi *Scanning Assets*

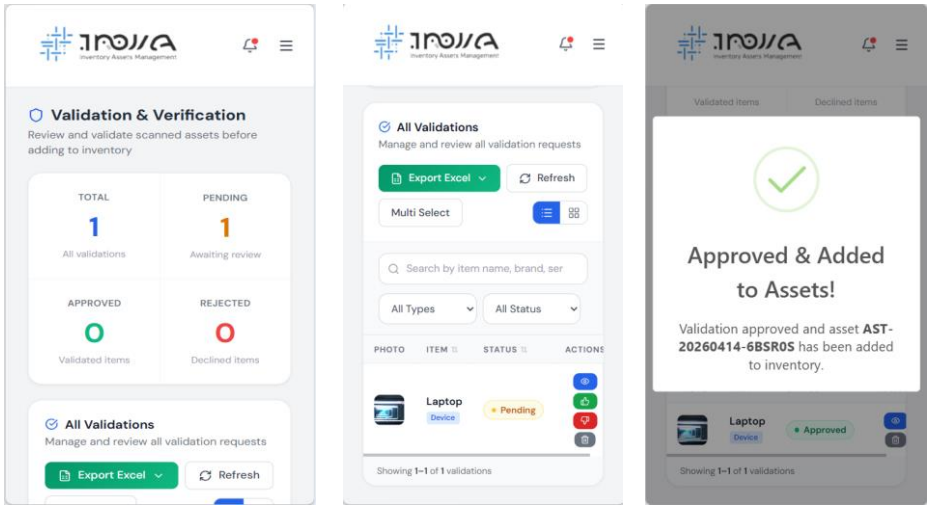
9. *Validation & Verification*

Hasil *scan* perangkat atau material akan muncul dihalaman *validation & verification*, disini *admin* dapat melihat seluruh data hasil *scanning* perangkat atau material beserta *serial number* atau *scan code* yang telah tersimpan. Halaman ini menampilkan daftar data dengan status seperti *pending*, *approved*, dan *rejected*, serta informasi detail seperti nama item, kategori, *brand*, *serial/scan code*, tanggal, dan pengguna yang melakukan *scanning*. *Admin* akan memilih setiap data untuk melihat detail hasil *scan* dan melakukan proses validasi dengan cara *approve* atau *reject*. Jika data dinyatakan valid (*approve*), maka sistem akan mengonfirmasi dan menyimpan data tersebut ke dalam *database* sebagai data final *assets* pada menu *Assets Inventory*. Namun jika data tidak sesuai (*reject*), maka sistem akan menandai data sebagai tidak valid dan mengarahkan *users* untuk melakukan proses *scanning* ulang. Selain itu, *admin* juga dapat menghapus data tertentu jika diperlukan. Seluruh proses ini memastikan bahwa data yang masuk ke dalam sistem inventaris telah melalui tahap verifikasi sehingga akurat dan sesuai dengan kondisi sebenarnya.


Clinton Alfaro Siagian
Home
Assets Scanning
Assets Inventory
History & Activity Log
Reports & Analytics
Wednesday, May 6, 2026
Validation & Verification
Review and validate scanned assets before adding to inventory
TOTAL
8
All validations
PENDING
0
Awaiting review
APPROVED
8
Validated items
REJECTED
0
Declined items
All Validations
Manage and review all validation requests
Excel
Refresh
Multi Select
Search by item name, brand, serial/scan code...
All Types
All Status
SESSION
PHOTO
ITEM NAME
TYPE
SERIAL/CODE
BRAND/VENDOR
PROJECT
DEPARTMENT
RECEIVER
STATUS
SUBMITTED
ACTIONS
DEVICES (4)
Check Laptop
Laptop
Device
Dell
Sofia
Procurement
Riono
Approved


Clinton Alfaro Siagian
Tuesday, April 14, 2026
Validation & Verification
TOTAL
1
All validations
REJECTED
0
Declined items
All Validations
Manage and review all validation requests
Export Excel
Refresh
Search by item name, brand, serial/scan code...
All Types
All Status
SESSION
PHOTO
ITEM
BRAND
CODE
STATUS
SUBMITTED
ACTIONS
Check Laptop
Laptop
Device
Dell
SN42342D2W
Pending
15 Apr 2026
by Clinton Alfaro Siagian
Showing 1–1 of 1 validations
Approve Validation?
Item: Laptop
Brand: Delll
Code: SN42342D2W
Approve 14 March 2026
Yes, Approve
Cancel

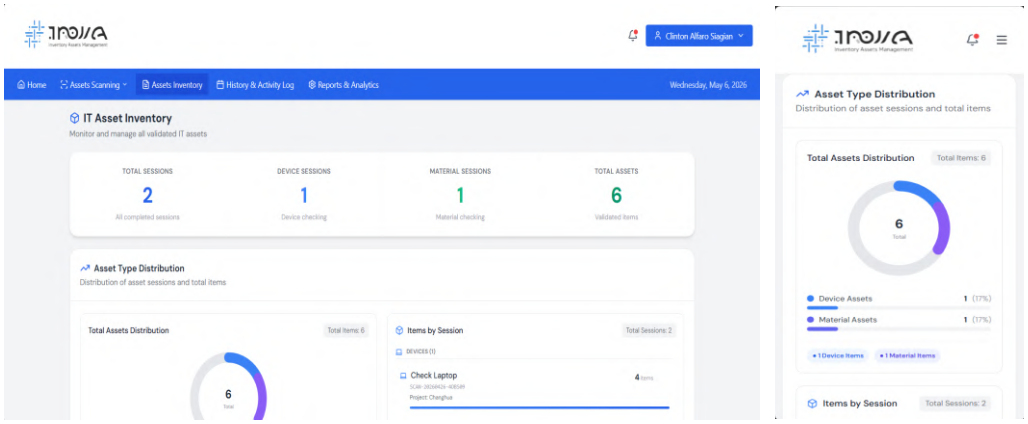

Clinton Alfaro Siagian
Tuesday, April 14, 2026
Validation & Verification
TOTAL
1
All validations
REJECTED
0
Declined items
All Validations
Manage and review all validation requests
Export Excel
Refresh
Search by item name, brand, serial/scan code...
All Types
All Status
SESSION
PHOTO
ITEM
ACTIONS
Check Laptop
Laptop
Device
Dell
SN42342D2W
Pending
15 Apr 2026
by Clinton Alfaro Siagian
Showing 1–1 of 1 validations
Reject Validation?
Item: Laptop
Brand: Delll
Code: SN42342D2W
Rejection Reason
Serial Number Not Valid
Yes, Reject
Cancel

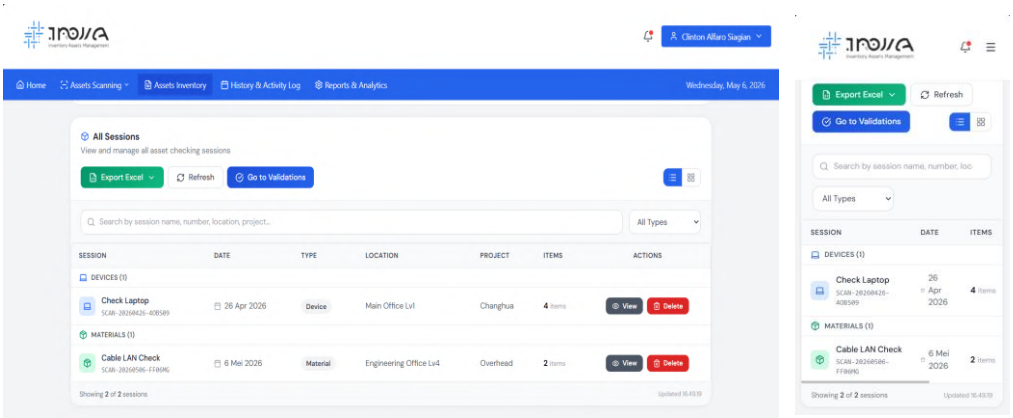


Gambar 4. 40 Implementasi *Validation & Verification*

10. *Assets Inventory*

Data hasil *scanning* dan validasi akan muncul sebagai data *Assets* dihalaman *Assets Inventory*, dimana *users* dapat melihat seluruh data aset yang telah melalui proses *scanning* dan validasi (*approve*) sehingga menjadi data akhir inventaris. Halaman ini menampilkan ringkasan seperti total *sessions*, *device sessions*, material *sessions*, serta total *assets* yang telah tervalidasi. Selain itu, ditampilkan daftar *session scanning* yang telah selesai, lengkap dengan informasi nama *session*, kode *scanning*, tanggal, lokasi, *project*, serta jumlah item. *Users* dapat melakukan pencarian data, melakukan *export* ke *Excel*, serta mengakses halaman validasi. Setiap *session* yang ditampilkan merupakan hasil dari proses *scanning* dan verifikasi sebelumnya, sehingga hanya data yang telah disetujui yang akan muncul pada halaman ini sebagai data inventaris resmi.

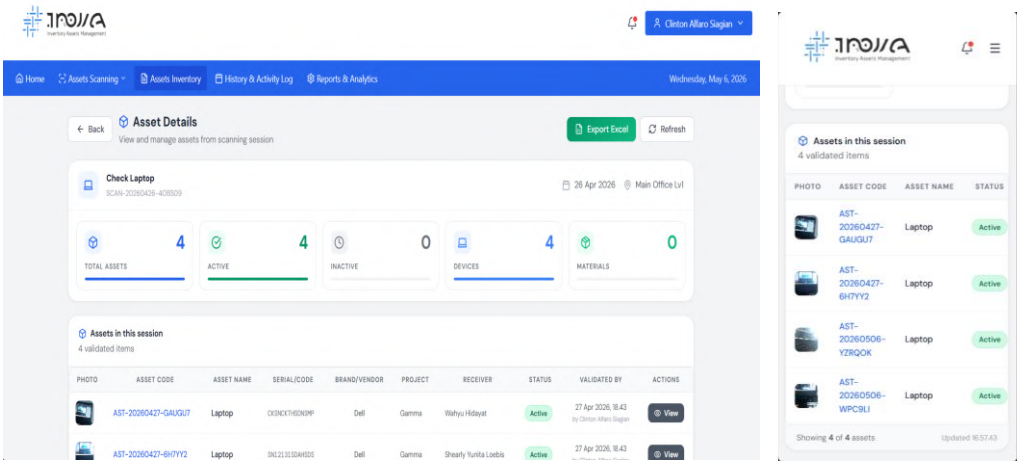




Gambar 4. 41 Implementasi *Assets Inventory*

11. *Assets* Details

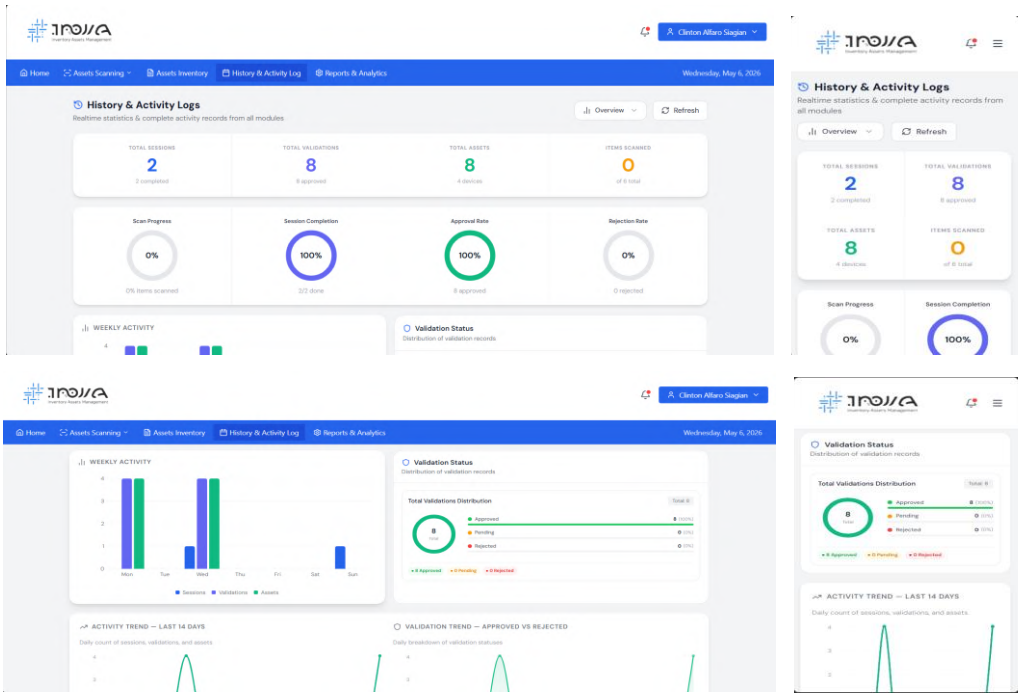
Pada halaman *Asset Details*, *users* dapat melihat rincian aset dari setiap *session* yang dipilih pada halaman *Assets Inventory*. Halaman ini menampilkan informasi seperti total aset, status aset (*active*, *inactive*), serta pembagian antara *devices* dan *materials*. Selain itu, ditampilkan daftar detail aset yang telah tervalidasi, meliputi *asset code*, nama aset, tipe, *serial number* atau *scan code*, status, tanggal validasi, serta pengguna yang melakukan validasi. *Users* juga dapat melihat detail aset lebih lanjut melalui aksi *view*. Seluruh data pada halaman ini merupakan hasil akhir dari proses *scanning* dan validasi yang telah dilakukan dan telah disimpan ke dalam *database* sebagai data inventaris, sehingga dapat digunakan untuk *monitoring* dan pengelolaan aset secara terstruktur dan akurat.



Gambar 4. 42 Implementasi *Assets Details*

12. *History & Logs*

Pada halaman *History & Activity Log*, *users* dapat memantau seluruh aktivitas sistem secara *real-time* yang mencakup proses *scanning*, validasi, hingga data aset yang telah dibuat. Halaman ini menampilkan ringkasan seperti total *session*, jumlah item yang di *scan*, jumlah validasi (*approved/rejected*), serta jumlah aset yang berhasil dibuat. Selain itu, ditampilkan juga grafik aktivitas mingguan, status *session* (*completed/in progress*), serta tingkat keberhasilan validasi seperti *approval rate* dan *rejection rate*. *Users* dapat melakukan filter atau pencarian berdasarkan jenis data *seperti scanning*, *validation*, *assets*, maupun *activity logs* untuk melihat detail aktivitas secara lebih spesifik. Seluruh data yang ditampilkan akan diperbarui secara otomatis dari *database*.

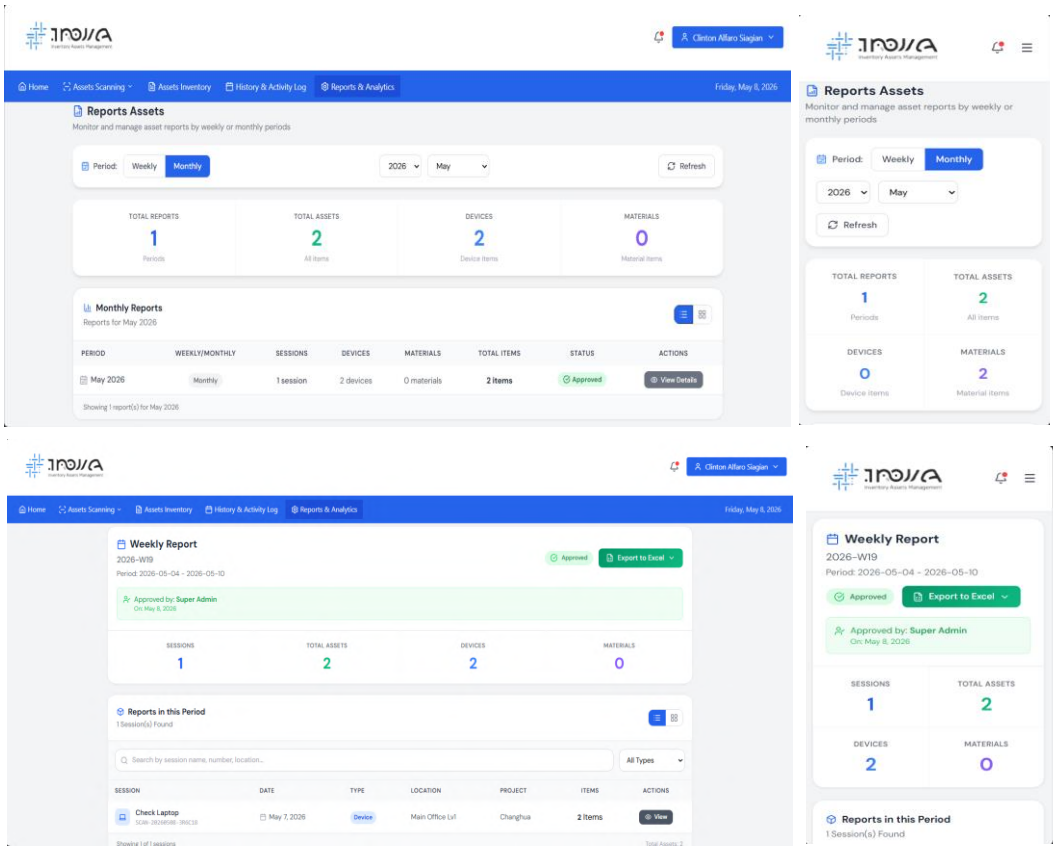


Gambar 4. 43 Implementasi *History & Activity Logs*

13. *Reports & Analytics*

Data *Assets* yang sudah di *scan* dan divalidasi sebelumnya akan muncul juga sebagai data laporan pengecekan inventaris pada halaman *Reports & Analytics*. Disini *admin* dapat melihat laporan hasil pengecekan inventaris berdasarkan data *scanning* dan validasi yang telah dilakukan. Sistem menampilkan ringkasan seperti *total reports*, *total assets*, *success rate*, serta perbandingan

devices dan materials. *Admin* disini dapat melihat dan mencari data laporan berdasarkan periode tertentu seperti *weekly* dan *monthly*, serta melihat status verifikasi laporan pengecekan yang telah dilakukan oleh *superadmin*, serta membuka detail laporan dengan tombol *view*. Selain itu, laporan dapat diekspor ke format *Excel* melalui tombol *Export Excel*. Jika belum ada data validasi, maka sistem akan menampilkan bahwa laporan belum tersedia karena laporan hanya terbentuk setelah proses validasi selesai.

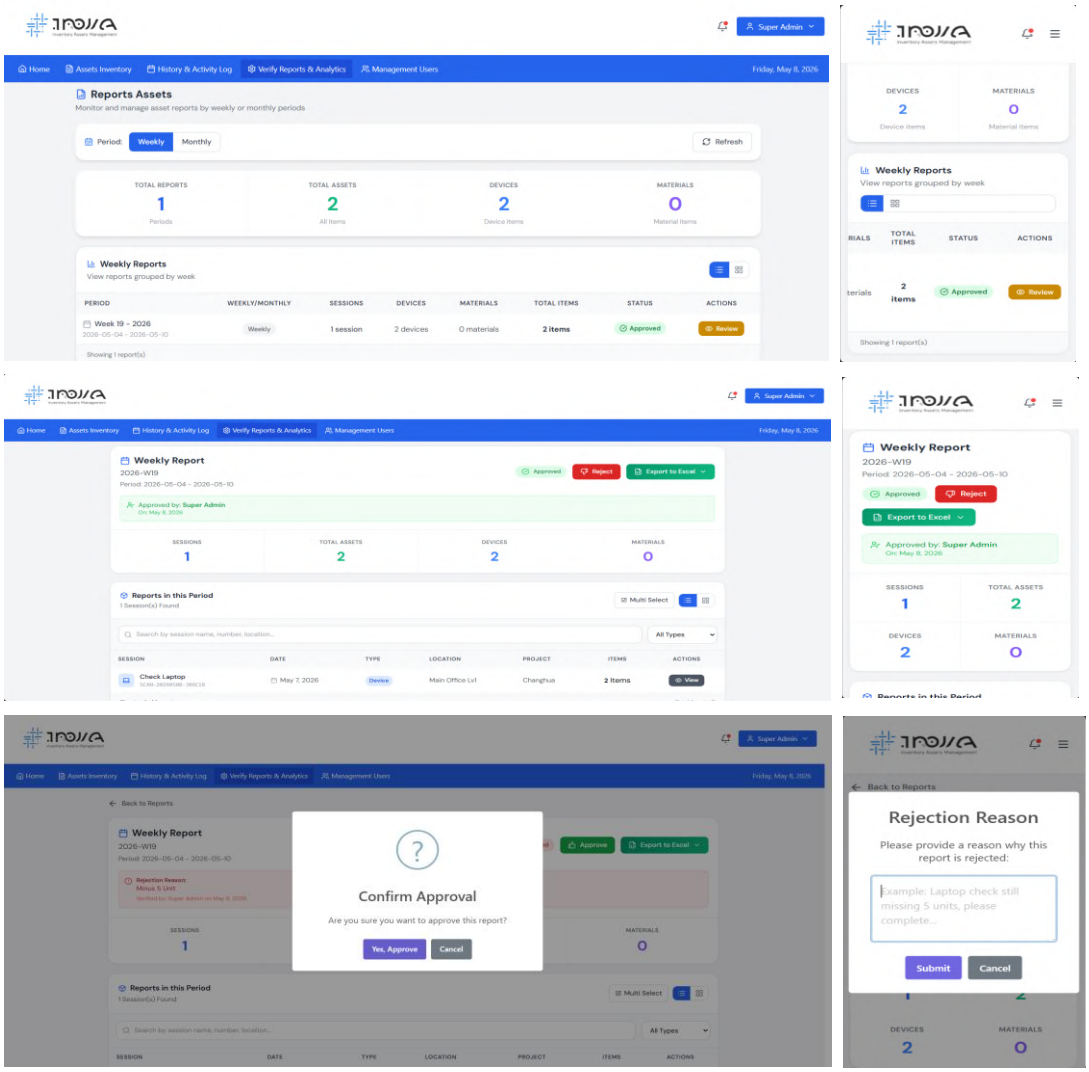


Gambar 4. 44 Implementasi *Report & Analytics*

14. *Verify Reports & Analytics*

Pada halaman *Verify Reports & Analytics*, fitur ini hanya dapat diakses oleh pengguna dengan *role Superadmin* dimana memiliki hak akses untuk melakukan *review* dan verifikasi laporan pengecekan inventaris aset yang telah dilakukan oleh seluruh pengguna sistem yaitu *Admin*. *Superadmin* disini dapat melihat data lengkap laporan berdasarkan periode tertentu dalam *weekly* dan *monthly* beserta informasi total *reports*, *assets*, *devices*, *materials*, serta status laporan pengecekan. Disini *superadmin* juga dapat melihat detail laporan

pengecekan dan memberikan status *approval* atau melakukan *reject* pada laporan. Jika laporan belum sesuai target atau data belum lengkap, maka laporan dapat di *reject* untuk dilakukan pengecekan kembali oleh *admin*. Namun jika laporan sudah sesuai, *Superadmin* dapat memberikan *approval* sehingga status verifikasi akan ditampilkan pada halaman *Reports & Analytics Admin*. Selain itu, laporan juga dapat diekspor ke format *Excel* untuk kebutuhan dokumentasi.

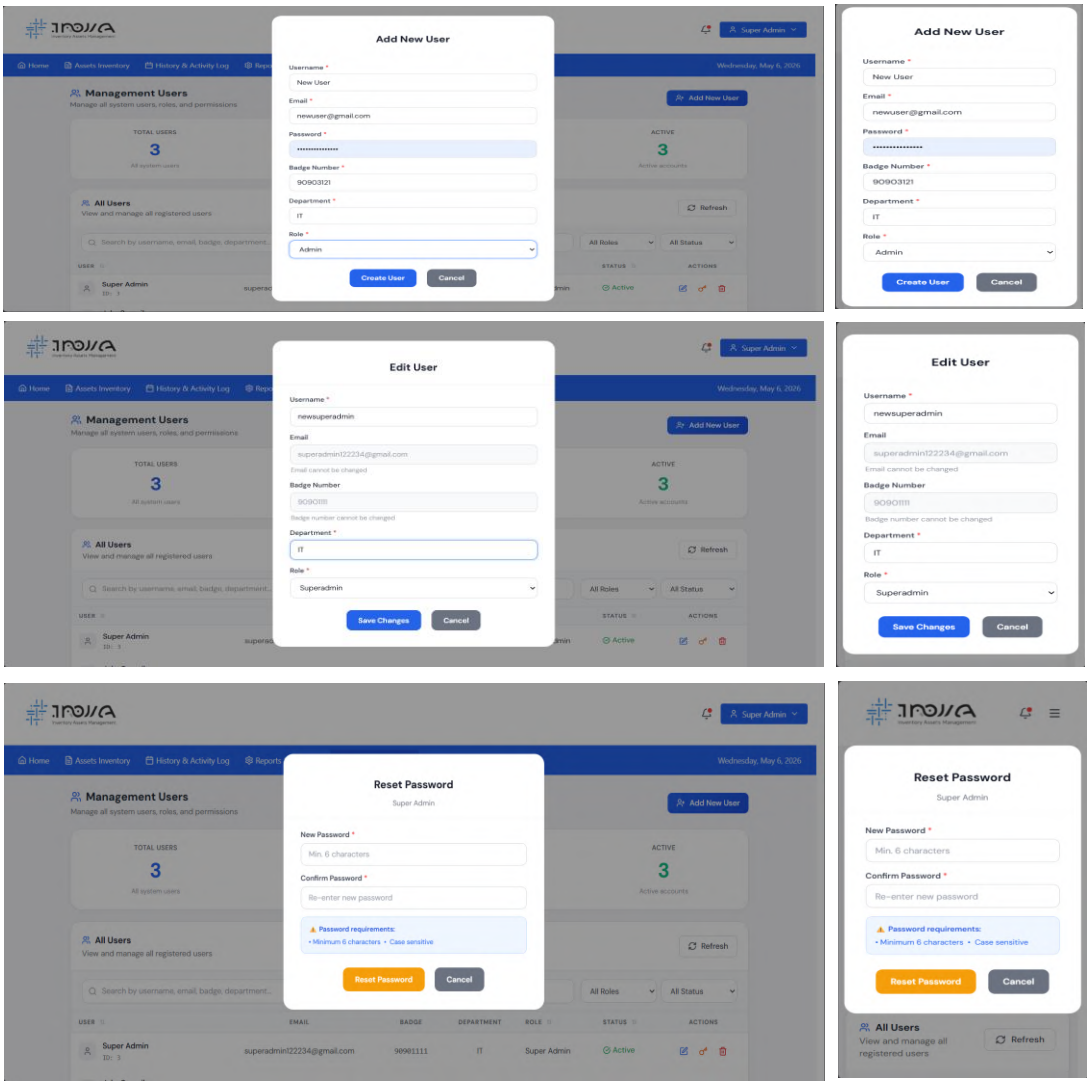


Gambar 4. 45 Implementasi *Verify Reports & Analytics*

15. *Management Users*

Pada halaman *Management Users*, fitur ini hanya dapat diakses oleh pengguna dengan *role Superadmin* dimana memiliki hak akses penuh dalam manajemen hak akses pengguna sistem. Pada halamn ini *Superadmin* dapat melihat seluruh daftar pengguna yang terdaftar lengkap dalam sistem dengan

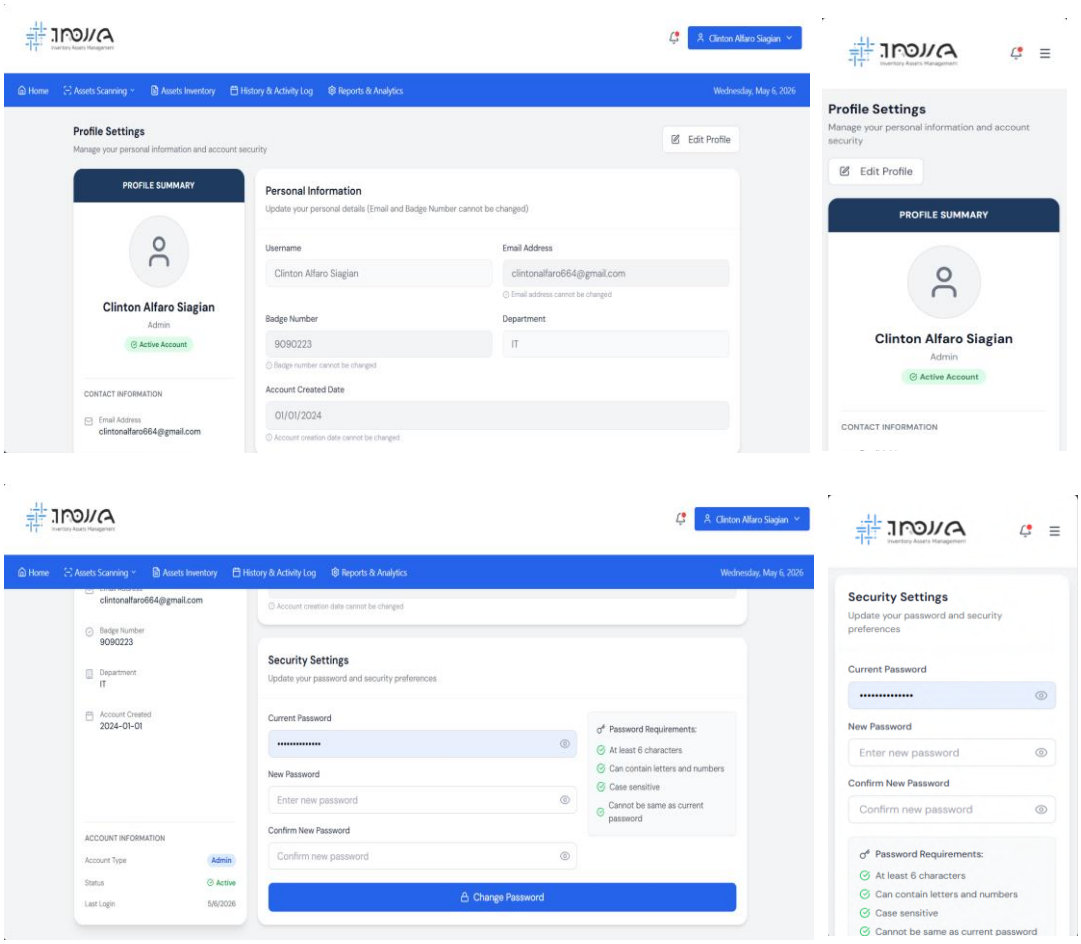
informasi seperti *username*, *email*, *badge number*, *department*, dan *role* pengguna. *Superadmin* disini dapat menambahkan pengguna baru dengan mengisi *form* yang tersedia, dimana sistem akan melakukan validasi untuk memastikan data yang di *create* tidak duplikasi data. *Superadmin* juga dapat melakukan edit data pengguna untuk memperbarui informasi pengguna serta dapat mengubah *role* dari pengguna tersebut. Jika pengguna mengalami kendala akses, *superadmin* dapat melakukan reset *password* sehingga pengguna dapat kembali mengakses sistem dengan *password* baru. Selain itu, *superadmin* juga dapat menghapus data pengguna dari sistem apabila sudah tidak diperlukan. Seluruh proses pengelolaan ini dirancang untuk memastikan data pengguna tetap terorganisir, aman, dan sesuai dengan kebutuhan operasional sistem.



Gambar 4. 46 Implementasi *Management Users*

16. *Profile*

Pada halaman *Profile Settings*, *users* dapat melihat dan mengelola informasi akun yang dimiliki, seperti *username*, *email*, *badge number*, dan *department* yang ditampilkan pada bagian *Profile Summary* dan *Personal Information*. Selain itu, *users* juga dapat melakukan perubahan *password* melalui fitur *Security Settings* dengan mengisi *current password*, *new password*, dan *confirm new password*. Sistem nantinya akan melakukan validasi terhadap input *password*, seperti memastikan panjang minimal karakter, kesesuaian konfirmasi *password*, serta tidak sama dengan *password* sebelumnya. Jika data yang dimasukkan valid, maka sistem akan menyimpan perubahan *password* ke *database* dan menampilkan notifikasi berhasil. Namun, jika tidak sesuai, *users* akan diminta untuk memperbaiki kembali input yang diberikan.



Gambar 4. 47 Implementasi *Profile*

4.2 Pengujian *Users* Acceptance Testing (UAT)

4.2.1 Pengujian Perangkat Lunak *(Functional Testing)*

Pada tahap ini dilakukan pengujian fungsional untuk memastikan bahwa seluruh fitur dalam sistem *INOVA* dapat berjalan sesuai dengan kebutuhan dan skenario bisnis yang telah ditetapkan sebelumnya. Pengujian dilakukan dengan menggunakan metode *blackbox*, yaitu metode yang berfokus pada pemeriksaan *output* berdasarkan *input* yang diberikan tanpa melihat struktur internal kode program. Dalam penelitian ini, terdapat 14 fungsionalitas utama yang menjadi objek pengujian, yang mencakup keseluruhan proses dalam sistem inventaris aset *IT*. Berikut hasil uji fungsionalitas dengan metode *blackbox*:

1. *Testing Scenario Use Case Register*

   Pengujian *Scenario Use Case Register* dilakukan berdasarkan *Use Case Scenario* (Tabel 3.3). *Use case scenario* mendefinisikan proses pengguna dalam melakukan registrasi akun ke dalam sistem *Inova*, namun pengujian dilakukan secara lebih rinci untuk memvalidasi setiap input dan respon sistem terhadap data yang dimasukkan pengguna. Hal ini bertujuan untuk memastikan proses registrasi berjalan sesuai dengan kebutuhan sistem.

Tabel 4. 2 *Blackbox Testing Register*

Test Case ID	***Scenario***	**Luaran yang Diharapkan**	**Luaran yang Dihasilkan**	**Hasil**	
				B	**G**
TC01	Data registrasi terisi lengkap dan valid *(username, email, badge number, department, role, pasword,* dan konfirmasi *password* sesuai.	Registrasi berhasil divalidasi dan menampilkan pesan "Registrasi berhasil" dan beralih halaman *login*.	Registrasi berhasil divalidasi dan beralih halaman *login*.	✓	
TC02	Data registrasi terisi dengan lengkap dan valid serta belum pernah terdaftar sebelumnya.	Registrasi berhasil divalidasi dan menyimpan data, kemudian menampilkan	Registrasi berhasil divalidasi dan menyimpan data, kemudian menampilkan	✓	

		pesan "Registrasi berhasil" dan mengarahkan kehalaman *login.*	pesan "Registrasi berhasil" dan mengarahkan kehalaman *login.*		
TC03	Data registrasi tidak diisi dengan lengkap (terdapat *field* yang kosong).	Registrasi tidak berhasil dan menampilkan pesan "Data *form* wajib diisi dengan lengkap".	Registrasi tidak berhasil dan menampilkan pesan "Data *form* wajib diisi dengan lengkap".	✓	
TC04	*Email* yang di *input* sudah terdaftar pada sistem.	Registrasi tidak berhasil dan menampilkan pesan "*Email* sudah digunakan".	Registrasi tidak berhasil dan menampilkan pesan "*Email* sudah digunakan".	✓	
TC05	*Username* yang di *input* sudah terdaftar pada sistem.	Registrasi tidak berhasil dan menampilkan pesan "U*sername* sudah digunakan".	Registrasi tidak berhasil dan menampilkan pesan "*username* sudah digunakan".	✓	
TC06	*Badge Number* yang di *input* sudah digunakan pengguna lain dan terdaftar pada sistem.	Registrasi tidak berhasil dan menampilkan pesan "B*adge number* sudah digunakan, gagal registrasi".	Registrasi tidak berhasil dan menampilkan pesan "B*adge number* sudah digunakan, gagal registrasi".	✓	
TC07	*Password* dan Konfirmasi *password* tidak sesuai.	Registrasi tidak berhasil dan sistem menampilkan pesan "Data *Password* tidak sesuai".	Registrasi tidak berhasil dan sistem menampilkan pesan "Data *Password* tidak sesuai".	✓	
TC08	Format *email* tidak sesuai ketentuan (tidak mengandung "@" atau *domain* yang lengkap seperti *.com*).	Registrasi tidak berhasil dan menampilkan pesan "Format email tidak valid".	Registrasi tidak berhasil dan menampilkan pesan "Format email tidak valid".	✓	

TC09	Seluruh data *form* registrasi tidak diisi.	Registrasi tidak berhasil dan menampilkan pesan "Seluruh data wajib diisi".	Registrasi tidak berhasil dan menampilkan pesan "Seluruh data wajib diisi".	✓	
TC10	Proses registrasi dilakukan saat koneksi internet tidak tersedia.	Registrasi tidak berhasil dan menampilkan pesan "Registrasi gagal, periksa koneksi internet".	Registrasi tidak berhasil dan menampilkan pesan "Registrasi gagal, periksa koneksi internet".	✓	

Hasil Analisis:

1. *Persentase test case* yang lolos = $\frac{10}{10} = 100\%$
2. *Persentase test case* yang gagal = $\frac{0}{10} = 0\%$
3. *Persentase test case* yang telah diuji = $\frac{10}{10} = 100\%$
4. *Persentase test case* yang belum diuji = $\frac{0}{10} = 0\%$

Jumlah total *test case* dalam pengujian = 10

2. *Testing Scenario Use Case Login*

Pengujian *Scenario Use Case Login* dilakukan berdasarkan *Use Case Scenario Login* (Tabel 3.4). *Use case scenario* mendefinisikan proses pengguna dalam melakukan *login* ke dalam sistem *Inova*, namun pengujian dilakukan secara lebih rinci untuk memvalidasi setiap kondisi input serta respon sistem terhadap proses autentikasi pengguna. Hal ini bertujuan untuk memastikan mekanisme *login* berjalan sesuai dengan kebutuhan sistem dan skenario bisnis yang telah dirancang.

Tabel 4. 3 *Blackbox Testing Login*

Test Case ID	***Scenario***	**Luaran yang Diharapkan**	**Luaran yang Dihasilkan**	**Hasil**	
				B	**G**

TC01	*Email* atau *Username* dan *password* valid serta terdaftar dalam sistem.	*Login* berhasil divalidasi dan memunculkan pesan "*Login* berhasil" dan beralih kehalaman *dashboard* sesuai *role* pengguna.	*Login* berhasil divalidasi dan beralih kehalaman *dashboard* sesuai *role* pengguna.	✓	
TC02	*Email* dan *Password* valid serta terdaftar dalam sistem.	*Login* berhasil dan menampilkan pesan *"Login* berhasil" dan mendapatkan akses kesistem dan beralih kehalaman *dashboard*.	*Login* berhasil dan menampilkan pesan *"Login* berhasil" dan mendapatkan akses kesistem dan beralih kehalaman *dashboard*.	✓	
TC03	*Username* dan *Password* valid serta terdaftar dalam sistem.	*Login* berhasil dan menampilkan pesan *"Login* berhasil" dan mendapatkan akses kesistem dan beralih kehalaman *dashboard*.	*Login* berhasil dan menampilkan pesan *"Login* berhasil" dan mendapatkan akses kesistem dan beralih kehalaman *dashboard*.	✓	
TC04	*Email* dan *Password* tidak sesuai dengan akun yang terdaftar.	*Login* tidak berhasil dan menampilkan pesan *"Email* atau *Password* salah".	*Login* tidak berhasil dan menampilkan pesan *"Email* atau *Password* salah".	✓	
TC05	*Email* atau *Username* tidak terdaftar dalam sistem.	*Login* tidak berhasil dan menampilkan pesan "Akun tidak ditemukan".	*Login* tidak berhasil dan menampilkan pesan "Akun tidak ditemukan".	✓	
TC06	*Email* atau *username* valid tetapi *password* kosong.	*Login* tidak berhasil dan menampilkan pesan "*Password* harus diisi".	*Login* tidak berhasil dan menampilkan pesan "*Password* harus diisi".	✓	

TC07	*Email* atau *username* kosong tetapi *password* diisi.	*Login* tidak berhasil dan menampilkan pesan "*Email* atau *Username* harus diisi".	*Login* tidak berhasil dan menampilkan pesan "*Email* atau *Username* harus diisi".	✓	
TC08	*Email* atau *username* dan password tidak diisi.	*Login* tidak berhasil dan menampilkan pesan "Seluruh data wajib diisi".	*Login* tidak berhasil dan menampilkan pesan "Seluruh data wajib diisi".	✓	
TC09	Format *email* tidak sesuai ketentuan (tidak mengandung "@gmail.com").	*Login* tidak berhasil dan menampilkan pesan "Format email tidak valid".	*Login* tidak berhasil dan menampilkan pesan "Format email tidak valid".	✓	
TC10	Proses *login* dilakukan saat koneksi internet tidak tersedia.	*Login* tidak berhasil dan menampilkan pesan "*Login* gagal, periksa koneksi internet".	*Login* tidak berhasil dan menampilkan pesan "*Login* gagal, periksa koneksi internet".	✓	

Hasil Analisis:

1. *Persentase test case* yang lolos = 10/10 = 100%
2. *Persentase test case* yang gagal = 0/10 = 0%
3. *Persentase test case* yang telah diuji = 10/10 = 100%
4. *Persentase test case* yang belum diuji = 0/10 = 0%

Jumlah total *test case* dalam pengujian = 10

3. *Testing Use Case Scenario Dashboard Admin & Superadmin*

Pengujian *Scenario Use Case Dashboard* dilakukan *Scenario Use Case Scenario Dashboard* (Tabel 3.5). *Use case scenario* mendefinisikan proses pengguna yang telah berhasil *login* dan mengakses halaman *dashboard* untuk melihat seluruh informasi inventaris secara *real-time* sesuai *role* pengguna, yaitu *Admin* dan *Superadmin*. Pengujian dilakukan secara rinci untuk memvalidasi tampilan ringkasan data aset, aktivitas *scanning*, status validasi, laporan pengecekan terbaru, serta grafik *monitoring* inventaris. Hal

ini bertujuan untuk memastikan halaman *dashboard* dapat menampilkan informasi sesuai kebutuhan sistem dan hak akses masing-masing pengguna.

Tabel 4. 4 *Blackbox Testing Dashboard Admin & Superadmin*

Test Case ID	***Scenario***	**Luaran yang Diharapkan**	**Luaran yang Dihasilkan**	**Hasil**	
				B	**G**
TC01	*Admin/Superadmin* berhasil *login* dan membuka halaman *dashboard*.	Halaman *dashboard Admin/Superadmin* tampil beserta ringkasan data inventaris.	Halaman *dashboard Admin/Superadmin* tampil beserta ringkasan data inventaris.	✓	
TC02	Data aset tersedia saat *dashboard Admin/Superadmin* diakses.	Sistem menampilkan ringkasan jumlah seluruh aset pada *dashboard.*	Sistem menampilkan ringkasan jumlah seluruh aset pada *dashboard.*	✓	
TC03	Data aktivitas *scanning* tersedia saat *dashboard Admin/Superadmin* diakses.	Sistem menampilkan aktivitas pengecekan atau *scanning* terbaru pada *dashboard.*	Sistem menampilkan aktivitas pengecekan atau *scanning* terbaru pada *dashboard.*	✓	
TC04	Data status validasi hasil *scanning* tersedia saat *dashboard Admin/Superadmin* diakses.	Sisitem menampilkan status validasi hasil *scanning (approve/reject)* pada *dashboard.*	Sisitem menampilkan status validasi hasil *scanning (approve/reject)* pada *dashboard.*	✓	
TC05	Data laporan pengecekan terbaru tersedia saat *dashboard Admin/Superadmin* diakses.	Sistem menampilkan data laporan pengecekan terbaru pada *dashboard.*	Sistem menampilkan data laporan pengecekan terbaru pada *dashboard.*	✓	
TC06	Data *monitoring* inventaris tersedia saat *dashboard*	Sistem menampilkan grafik *monitoring*	Sistem menampilkan grafik *monitoring*	✓	

	Admin/Superadmin diakses.	inventaris secara *real-time* pada *dashboard*.	inventaris secara *real-time* pada *dashboard*.		
TC07	*Dashboard Admin/Superadmin* diakses saat data aset belum tersedia.	Sistem menampilkan pesan "Data aset belum tersedia, lakukan pengecekan terlebih dahulu".	Sistem menampilkan Pesan "Data aset belum tersedia, lakukan pengecekan terlebih dahulu".		
TC08	*Dashboard Admin/Superadmin* diakses saat aktivitas *scanning* belum tersedia.	Sistem menampilkan pesan "Belum ada aktivitas pengecekan" ditampilkan.	Sistem menampilkan pesan "Belum ada aktivitas pengecekan" ditampilkan.	✓	
TC09	*Dashboard Admin/Superadmin* diakses saat data validasi belum tersedia.	Sistem menampilkan pesan "Belum ada data yang divalidasi".	Sistem menampilkan pesan "Belum ada data yang divalidasi".	✓	
TC10	*Dashboard Admin/Superadmin* diakses saat data laporan pengecekan belum tersedia.	Sistem menampilkan pesan "Belum ada pengecekan terbaru".	Sistem menampilkan pesan "Belum ada pengecekan terbaru".	✓	
TC11	*Dashboard Admin/Superadmin* diakses saat grafik gagal dimual.	Sistem menampilkan pesan "Data grafik tidak tersedia".	Sistem menampilkan pesan "Data grafik tidak tersedia".	✓	

Hasil Analisis:

1. *Persentase test case* yang lolos = $\frac{11}{11} = 100\%$
2. *Persentase test case* yang gagal = $\frac{0}{11} = 0\%$
3. *Persentase test case* yang telah diuji = $\frac{11}{11} = 100\%$
4. *Persentase test case* yang belum diuji = $\frac{0}{11} = 0\%$

Jumlah total *test case* dalam pengujian = 11

4. *Testing Use Case Scenario Scanning Preparation*

Pengujian *Scenario Use Case Create Scanning Preparation* dilakukan berdasarkan *Use Case Scenario Scanning Preparation* (Tabel 3.6). *Use case scenario* mendefinisikan proses *Admin* dalam membuat data *scanning preparation* dengan mengisi informasi pengecekan, memilih kategori perangkat atau material, serta menginput detail item yang akan dilakukan pengecekan. Pengujian dilakukan secara lebih rinci untuk memvalidasi setiap input dan respon sistem terhadap data yang dimasukkan. Hal ini bertujuan untuk memastikan proses pembuatan *scanning preparation* berjalan sesuai dengan kebutuhan sistem dan skenario bisnis yang telah dirancang.

Tabel 4. 5 *Blackbox Testing Create Scanning Preparation*

Test Case ID	***Scenario***	**Luaran yang Diharapkan**	**Luaran yang Dihasilkan**	**Hasil**	
				B	**G**
TC01	*Admin* mengakses menu "*Assets Scanning"* dan memilih *"Create Scanning Preparation"*.	Sistem menampilkan halaman *form create scanning preparation*.	Sistem menampilkan halaman *form create scanning preparation*.	✓	
TC02	Data *form basic information (checking name, category, loaction check, checking date)* diisi lengkap	Sistem tidak menampilkan pesan dan dapat dilanjutkan ke pengisian *items preparation*.	Sistem tidak menampilkan pesan dan dapat dilanjutkan ke pengisian *items preparation*.	✓	
TC03	Data *form basic information (checking name, category, loaction check, checking date)* tidak diisi dengan lengkap atau kosong.	Sistem menampilkan pesan "Semua data wajib diisi" dan diminta melakukan pengisian ulang data.	Sistem menampilkan pesan "Semua data wajib diisi" dan diminta melakukan pengisian ulang data.	✓	
TC04	*Admin* memilih data kategori *devices*.	Sistem menampilkan *form items to scan (Device name, Device*	Sistem menampilkan *form items to scan (Device name, Device*	✓	

		details, Brand, Vendor, Model, *Spesifications, Quantity, Project).*	*details, Brand, Vendor,* Model, *Spesifications, Quantity, Project).*		
TC05	*Admin* tidak memilih kategori *devices*	Sistem menampilkan pesan "Pilih kategori terlebih dahulu" dan proses tidak dapat dilanjutkan.	Sistem menampilkan pesan "Pilih kategori terlebih dahulu" dan proses tidak dapat dilanjutkan.	✓	
TC06	Detail item kategori *devices* diisi lengkap *(device name, brand, vendor,* model, *specifications, quantity, project).*	Sistem tidak menampilkan pesan dan melanjutkan proses serta ditampilkan pada daftar item.	Sistem tidak menampilkan pesan dan melanjutkan proses serta ditampilkan pada daftar item.	✓	
TC07	Detail item kategori *devices* tidak diisi dengan lengkap *(device name, brand, vendor,* model, *specifications, quantity, project).*	Sistem menampilkan pesan "Seluruh data *items device* wajib diisi dengan lengkap".	Sistem menampilkan pesan "Seluruh data wajib diisi dengan lengkap".	✓	
TC08	*Admin* memilih data kategori *materials*.	Sistem menampilkan *form items to scan (material name, quantity, unit of measure (UOM), vendor, project).*	Sistem menampilkan *form items to scan (material name, quantity, unit of measure (UOM), vendor, project).*	✓	
TC09	*Admin* tidak memilih kategori *materials*	Sistem menampilkan pesan "Pilih kategori terlebih dahulu" dan	Sistem menampilkan pesan "Pilih kategori terlebih dahulu" dan	✓	

		proses tidak dapat dilanjutkan.	proses tidak dapat dilanjutkan.		
TC10	Detail item kategori *materials* diisi lengkap *(material name, quantity, unit of measure (UOM), vendor, project).*	Sistem tidak menampilkan pesan dan melanjutkan proses serta ditampilkan pada daftar item.	Sistem tidak menampilkan pesan dan melanjutkan proses serta ditampilkan pada daftar item.	✓	
TC11	Detail item kategori *materials* tidak diisi dengan lengkap *(material name, quantity, unit of measure (UOM), vendor, project).*	Sistem menampilkan pesan "Seluruh data item *material* wajib diisi dengan lengkap".	Sistem menampilkan pesan "Seluruh data item *material* wajib diisi dengan lengkap".	✓	
TC12	Data item diisi tidak valid (*quantity* tidak sesuai atau kosong).	Sistem menampilkan pesan "Seluruh data wajib diisi, silahkan perbaiki data".	Sistem menampilkan pesan "Seluruh data wajib diisi, silahkan perbaiki data".	✓	
TC13	Seluruh data *basic information* dan *items* detail diisi dengan benar.	Data disimpan sementara dan dapat dilanjutkan ke tahap pengaturan metode penyimpanan.	Data disimpan sementara dan dapat dilanjutkan ke tahap pengaturan metode penyimpanan.	✓	

Hasil Analisis:

1. *Persentase test case* yang lolos = $\frac{13}{13} = 100\%$
2. *Persentase test case* yang gagal = $\frac{0}{13} = 0\%$
3. *Persentase test case* yang telah diuji = $\frac{13}{13} = 100\%$
4. *Persentase test case* yang belum diuji = $\frac{0}{13} = 0\%$

Jumlah total *test case* dalam pengujian = 13

5. *Testing Use Case Scenario* Pengaturan Metode Penyimpanan

Pengujian *Scenario Use Case* Pengaturan Metode Penyimpanan dilakukan berdasarkan *Use Case Scenario* Pengaturan Metode Penyimpanan (Tabel 3.7). *Use case scenario* mendefinisikan proses *Admin* dalam menentukan metode penyimpanan hasil *scanning*, yaitu *save to stock* atau *distribute directly*, serta mengisi jumlah item sesuai dengan batas *quantity* yang telah ditentukan hingga data disimpan sebagai *session scanning*. Pengujian dilakukan secara rinci untuk memvalidasi setiap proses mulai dari pemilihan metode, pengisian jumlah item, hingga validasi dan penyimpanan data. Hal ini bertujuan untuk memastikan proses pengaturan metode penyimpanan berjalan sesuai dengan kebutuhan sistem dan skenario bisnis yang telah dirancang.

Tabel 4. 6 *Blackbox Testing* Pengaturan Metode Penyimpanan

Test Case ID	***Scenario***	**Luaran yang Diharapkan**	**Luaran yang Dihasilkan**	**Hasil**	
				B	**G**
TC01	*Admin* berada pada tahap pengaturan metode penyimpanan setelah mengisi *form basic information* dan *items to scan*.	Sistem menampilkan pengaturan metode penyimpanan.	Sistem menampilkan pengaturan metode penyimpanan.	✓	
TC02	*Admin* berada pada tahap pilihan metode penyimpanan.	Sistem menampilkan opsi pemilihan metode penyimpanan berupa *save to stock* dan *distribute directly*.	Sistem menampilkan opsi pemilihan metode penyimpanan berupa *save to stock* dan *distribute directly*.	✓	
TC03	*Admin* tidak memilih metode penyimpanan	Sistem menolak proses menampilkan pesan "Pilih metode penyimpanan".	Sistem menolak proses menampilkan pesan "Pilih metode penyimpanan".	✓	

TC04	*Admin* mengisi jumlah item sesuai *quantity*.	Sistem menerima input jumlah item dan menampilkan jumlah item sesuai batas *quantity*.	Sistem menerima input jumlah item dan menampilkan jumlah item sesuai batas *quantity*.	✓	
TC05	*Admin* tidak mengisi jumlah item tidak sesuai atau kurang dari *quantity*.	Sistem menolak input dan menampilkan pesan "Jumlah item wajib diisi".	Sistem menolak input dan menampilkan pesan "Jumlah item wajib diisi".	✓	
TC06	*Admin* mengisi jumlah item melebihi batas *quantity*.	Sistem menolak input dan menampilkan pesan "Jumlah item melebihi batas *quantity*".	Sistem menolak input dan menampilkan pesan "Jumlah item melebihi batas *quantity*".	✓	
TC07	*Admin* memilih metode penyimpanan *Save to stock*.	Sistem menampilkan dan menyimpan pilihan metode *save to stock*.	Sistem menampilkan dan menyimpan pilihan metode *save to stock*.	✓	
TC08	*Admin* memilih metode penyimpanan *Distribute directly*.	Sistem menampilkan dan menyimpan pilihan metode *distribute directly*.	Sistem menampilkan dan menyimpan pilihan metode *distribute directly*.	✓	
TC09	*Admin* melakukan *create* data setelah memilih metode dan mengisi jumlah.	Sistem menampilkan pesan "Berhasil membuat *scanning preparation"*.	Sistem menampilkan pesan "Berhasil membuat *scanning preparation"*.	✓	
TC10	Seluruh data *preparation* dan metode penyimpanan valid.	Sistem menyimpan data sebagai *session scanning* dan menampilkan "Berhasil membuat *scanning preparation"*.	Sistem menyimpan data sebagai *session scanning* dan menampilkan "Berhasil membuat *scanning preparation"*.	✓	

TC11	Seluruh data *preparation* dan metode penyimpanan tidak valid.	Sistem menampilkan pesan "Data tidak valid, periksa data kembali".	Sistem menampilkan pesan "Data tidak valid, periksa data kembali".	✓	

Hasil Analisis:

1. *Persentase test case* yang lolos = $\frac{11}{11} = 100\%$
2. *Persentase test case* yang gagal = $\frac{0}{11} = 0\%$
3. *Persentase test case* yang telah diuji = $\frac{11}{11} = 100\%$
4. *Persentase test case* yang belum diuji = $\frac{0}{11} = 0\%$

Jumlah total *test case* dalam pengujian = 11

6. *Testing Use Case Scenario Scanning* Perangkat atau Material

Pengujian *Scenario Use Case Scanning* Perangkat atau Material dilakukan berdasarkan *Use Case Scenario Scanning* Perangkat atau Material (Tabel 3.8). *Use case scenario* ini mendefinisikan proses *Admin* dalam melakukan pemindaian perangkat atau material menggunakan kamera atau input manual berdasarkan *session scanning* yang telah dibuat sebelumnya. Pengujian dilakukan secara rinci untuk memvalidasi setiap proses mulai dari pemilihan *session scanning*, aktivasi kamera, proses deteksi objek, hingga hasil deteksi yang ditampilkan. Hal ini bertujuan untuk memastikan proses *scanning* perangkat atau material berjalan sesuai dengan kebutuhan sistem dan skenario bisnis yang telah dirancang.

Tabel 4. 7 *Blackbox Testing Scanning* Perangkat atau Material

Test Case ID	***Scenario***	**Luaran yang Diharapkan**	**Luaran yang Dihasilkan**	**Hasil**	
				B	**G**
TC01	*Admin* berada pada halaman *Scanning* dan melihat *session scanning* yang tersedia.	Sistem menampilkan halaman *scanning* beserta daftar *session scanning* yang aktif dan *pending* atau *on progres* atau *completed*.	Sistem menampilkan halaman *scanning* beserta daftar *session scanning* yang aktif dan *pending* atau *on progres*.	✓	

TC02	*Admin* memilih salah satu *session scanning*.	Sistem menampilkan status *progress bar session scanning* aktif dan siap digunakan untuk proses *scanning*.	Sistem menampilkan status *progress bar session scanning* aktif dan siap digunakan untuk proses *scanning*.	✓	
TC03	*Admin* memilih *session scanning* dengan status *pending*.	Sistem menampilkan status *progress bar session scanning pending* aktif dan siap digunakan untuk proses *scanning*.	Sistem menampilkan status *progress bar session scanning pending* aktif dan siap digunakan untuk proses *scanning*.	✓	
TC04	*Admin* memilih *session scanning* dengan status *on progress*.	Sistem menampilkan status *progress bar session scanning on progress* aktif dan menampilkan *progress scanning* sebelumnya dan siap digunakan untuk proses *scanning*.	Sistem menampilkan status *progress bar session scanning on progress* aktif dan menampilkan *progress scanning* sebelumnya dan siap digunakan untuk proses *scanning*.	✓	
TC05	*Admin* memilih *session scanning* dengan status *completed*.	Sistem menampilkan status *progress bar session scanning completed* aktif dan hanya dapat dilihat dan tidak dapat digunakan *process scanning* kembali kecuali ada perbaikan data.	Sistem menampilkan status *progress bar session scanning completed* aktif dan hanya dapat dilihat dan tidak dapat digunakan *process scanning* kembali kecuali ada perbaikan data.	✓	
TC06	Kamera berhasil diaktifkan saat memulai *scanning*.	Kamera aktif dengan menampilkan nama	Kamera aktif dengan menampilkan nama	✓	

		kamera yang aktif dan siap digunakan untuk mendeteksi objek.	kamera yang aktif dan siap digunakan untuk mendeteksi objek.		
TC07	Kamera tidak tersedia atau tidak aktif saat proses *scanning*.	Kamera tidak aktif dan proses tidak dapat dilanjutkan dan menampilkan pesan "Kamera tidak tersedia, silakan periksa perangkat".	Kamera tidak aktif dan proses tidak dapat dilanjutkan dan menampilkan pesan "Kamera tidak tersedia, silakan periksa perangkat".	✓	
TC08	*Admin* mengarahkan kamera ke objek perangkat atau material dengan jelas.	Objek berhasil ditangkap oleh kamera dan melakukan deteksi.	Objek berhasil ditangkap oleh kamera dan melakukan deteksi.	✓	
TC09	*Admin* mengarahkan kamera ke objek perangkat atau material dengan kondisi kamera buram.	Objek tidak berhasil ditangkap dan menampilkan pesan "Objek tidak terbaca, kamera terlalu buram, silahkan ulangi lagi".	Objek tidak berhasil ditangkap dan menampilkan pesan "Objek tidak terbaca, kamera terlalu buram, silahkan ulangi lagi".	✓	
TC10	*Admin* mengarahkan kamera ke objek perangkat atau material dengan jelas dan sistem berhasil mendeteksi objek perangkat atau material.	Sistem menampilkan pesan sukses "Objek dikenali sesuai kategori *(devices/materials)*" dengan nama objeknya.	Sistem menampilkan pesan sukses "Objek dikenali sesuai kategori *(devices/materials)* dengan nama objeknya.	✓	
TC11	*Admin* mengarahkan kamera ke objek perangkat atau material dengan jelas dan sistem gagal	Sistem menampilkan pesan gagal "Gagal mendeteksi objek, coba kembali".	Sistem menampilkan pesan gagal "Gagal mendeteksi objek, coba kembali".	✓	

	mendeteksi objek perangkat atau material.				
TC12	Objek yang terdeteksi sesuai dengan *session scanning*.	Sistem menampilkan hasil deteksi dilayar dan memunculkan pesan sukses "Objek dikenali sesuai kategori *(devices/materials)*" dengan nama objeknya.	Sistem menampilkan hasil deteksi dilayar dan memunculkan pesan sukses "Objek dikenali sesuai kategori *(devices/materials)*" dengan nama objeknya.	✓	
TC13	Objek yang terdeteksi tidak sesuai dengan *session scanning*.	Sistem menampilkan nama objek yang dideteksi dan muncul pesan "Objek tidak sesuai dengan *session scanning*, silahkan ulangi lagi" serta proses tidak dapat dilanjutkan.	Sistem menampilkan nama objek yang dideteksi dan muncul pesan "Objek tidak sesuai dengan *session scanning*, silahkan ulangi lagi" serta proses tidak dapat dilanjutkan.	✓	
TC14	Objek yang terdeteksi berhasil ditampilkan pada layar.	Sistem menampilkan informasi objek dan siap dilanjutkan ke proses *scanning serial number* atau *scan code*.	Sistem menampilkan informasi objek dan siap dilanjutkan ke proses *scanning serial number* atau *scan code*.	✓	
TC15	*Admin* melanjutkan ke proses *scanning serial number* atau *scan code* setelah objek terdeteksi.	Proses berpindah ke tahap *scanning serial number* atau *scan code.*	Proses berpindah ke tahap *scanning serial number* atau *scan code.*	✓	

Hasil Analisis:

1. *Persentase test case* yang lolos = $\frac{15}{15} = 100\%$
2. *Persentase test case* yang gagal = $\frac{0}{15} = 0\%$
3. *Persentase test case* yang telah diuji = $\frac{15}{15} = 100\%$

4. *Persentase test case* yang belum diuji = $\frac{0}{15} = 0\%$

Jumlah total *test case* dalam pengujian = 15

7. *Testing Use Case Scenario Scanning* & Ekstraksi *Serial Number* atau *Scan Code*

Pengujian *Scenario Use Case Scanning Serial Number* atau *Scan Code* dilakukan berdasarkan *Use Case Scenario Scanning &* Ekstraksi *Serial Number* atau *Scan Code* (Tabel 3.9). *Use case scenario* mendefinisikan proses *Admin* dalam melakukan *scanning serial number* atau *scan code* dilakukan setelah objek perangkat atau material berhasil dideteksi, kemudian sistem melakukan ekstraksi teks menggunakan *OCR* dan menampilkan hasilnya untuk proses validasi. Pengujian dilakukan secara rinci untuk memvalidasi setiap proses mulai dari deteksi area *serial number* atau *scan code,* proses ekstraksi teks, hingga pengecekan hasil dan penanganan kondisi *error* seperti data tidak terbaca, duplikat, atau format tidak valid. Hal ini bertujuan untuk memastikan seluruh proses berjalan sesuai dengan kebutuhan sistem dan skenario bisnis yang telah dirancang.

Tabel 4. 8 *Blackbox Testing Scanning &* Ekstraksi *Serial Number* atau *Scan Code*

Test Case ID	***Scenario***	**Luaran yang Diharapkan**	**Luaran yang Dihasilkan**	**Hasil**	
				B	**G**
TC01	*Admin* berada pada proses *scanning serial number* atau *scan code* setelah objek (perangkat atau material) terdeteksi.	Sistem menampilkan tombol *scan* untuk *scan serial number* atau *scan code* atau *skip.*	Sistem menampilkan tombol *scan* untuk *scan serial number* atau *scan code* atau *skip.*	✓	
TC02	*Admin* berada pada proses *scanning serial number* atau *scan code* setelah objek (perangkat atau material) terdeteksi dan	Sistem menampilkan hasil deteksi objek tanpa teks *serial number* atau *scan code.*	Sistem menampilkan hasil deteksi objek tanpa teks *serial number* atau *scan code.*	✓	

	memilih tombol skip *scan serial number*.				
TC03	Kamera berhasil diaktifkan saat memulai *scanning*.	Kamera aktif dengan menampilkan nama kamera yang aktif dan siap digunakan untuk mendeteksi *serial number* atau *scan code*.	Kamera aktif dengan menampilkan nama kamera yang aktif dan siap digunakan untuk mendeteksi *serial number* atau *scan code*.	✓	
TC04	Kamera tidak tersedia atau tidak aktif saat proses *scanning*.	Kamera tidak aktif dan proses tidak dapat dilanjutkan dan menampilkan pesan "Kamera tidak tersedia, silakan periksa perangkat".	Kamera tidak aktif dan proses tidak dapat dilanjutkan dan menampilkan pesan "Kamera tidak tersedia, silakan periksa perangkat".	✓	
TC05	*Admin* mengarahkan kamera ke *serial number* atau *scan code* dengan jelas.	Area *serial number* atau *scan code* berhasil ditangkap oleh kamera dan melakukan deteksi.	Area *serial number* atau *scan code* berhasil ditangkap oleh kamera dan melakukan deteksi.	✓	
TC06	*Admin* mengarahkan kamera ke *serial number* atau *scan code* dengan kondisi kamera buram.	Area *serial number* atau *scan code* tidak berhasil ditangkap dan menampilkan pesan "Data tidak terbaca, kamera terlalu buram, silahkan ulangi *scan*".	Area *serial number* atau *scan code* tidak berhasil ditangkap dan menampilkan pesan "Data tidak terbaca, kamera terlalu buram, silahkan ulangi *scan*".	✓	
TC07	*Admin* mengarahkan kamera ke *serial number* atau *scan code* dengan jelas dan sistem berhasil	Sistem menampilkan pesan sukses "*Serial Number* terdeteksi".	Sistem menampilkan pesan sukses "*Serial Number* terdeteksi".	✓	

	mendeteksi *serial number* atau *scan code*.				
TC08	*Admin* mengarahkan kamera ke *serial number* atau *scan code* dengan jelas dan sistem gagal mendeteksi *serial number* atau *scan code*.	Sistem menampilkan pesan gagal "Gagal mendeteksi *serial number* atau *scan code*, coba kembali".	Sistem menampilkan pesan gagal "Gagal mendeteksi *serial number* atau *scan code*, coba kembali".	✓	
TC09	*Admin* mengarahkan kamera ke *serial number* atau *scan code* dengan jelas dan sistem berhasil melakukan ekstraksi teks.	Sistem menampilkan pesan sukses "*Serial number* atau *Scan code* terdeteksi" dan memunculkan teks *serial number* atau *scan code*.	Sistem menampilkan pesan sukses "*Serial number* atau *Scan code* terdeteksi" dan memunculkan teks *serial number* atau *scan code*.	✓	
TC10	Hasil ekstraksi teks *serial number* atau *scan code* hasil *scan* jelas dan sesuai.	Hasil *serial number* atau *scan code* ditampilkan dan dapat dilanjutkan ke proses validasi.	Hasil *serial number* atau *scan code* ditampilkan dan dapat dilanjutkan ke proses validasi.	✓	
TC11	Hasil ekstraksi teks *serial number* atau *scan code* hasil *scan* tidak jelas atau tidak akurat.	Hasil *serial number* atau *scan code* tidak ditampilkan dan diminta melakukan *scan ulang* atau input manual.	Hasil *serial number* atau *scan code* tidak ditampilkan dan diminta melakukan *scan ulang* atau input manual.	✓	
TC12	Hasil data dan ekstraksi teks *serial number* atau *scan code* terdeteksi data duplikat.	Sistem menampilkan pesan "Data sudah terdaftar (duplikat)".	Sistem menampilkan pesan "Data sudah terdaftar (duplikat)".	✓	
TC13	Format *serial number* atau *scan code* hasil *scan* sesuai standar.	Sistem menampilkan pesan sukses "*Serial number* atau *Scan code*	Sistem menampilkan pesan sukses "*Serial number* atau *Scan code*	✓	

		terdeteksi" dan memunculkan teks *serial number* atau *scan code*.	terdeteksi" dan memunculkan teks *serial number* atau *scan code*.		
TC14	Format *serial number* atau *scan code* hasil *scan* tidak sesuai standar.	Sistem menampilkan pesan gagal "Format *serial number* atau *scan code* tidak valid".	Sistem menampilkan pesan gagal "Format *serial number* atau *scan code* tidak valid".	✓	
TC15	*Admin* melakukan input manual *serial number* atau *scan code* sesuai format.	Sistem menampilkan pesan sukses "*Serial number* atau *Scan code* terdeteksi" dan memunculkan teks *serial number* atau *scan code*.	Sistem menampilkan pesan sukses "*Serial number* atau *Scan code* terdeteksi" dan memunculkan teks *serial number* atau *scan code*.	✓	
TC16	*Admin* melakukan input manual *serial number* atau *scan code* tidak sesuai dengan format.	Sistem menampilkan pesan gagal "Format *serial number* atau *scan code* tidak valid".	Sistem menampilkan pesan gagal "Format *serial number* atau *scan code* tidak valid".	✓	
TC17	*Admin* melakukan pengiriman data hasil *scan* secara *single*.	Sistem menampilkan pesan sukses "Data berhasil terkirim untuk divalidasi" dan beralih kehalaman validasi dan verifikasi data.	Sistem menampilkan pesan sukses "Data berhasil terkirim untuk divalidasi" dan beralih kehalaman validasi dan verifikasi data.	✓	
TC18	*Admin* melakukan pengiriman data hasil *scan* secara *submit all*.	Sistem menampilkan pesan sukses "Seluruh data berhasil terkirim untuk divalidasi" dan beralih kehalaman validasi dan verifikasi data.	Sistem menampilkan pesan sukses "Seluruh data berhasil terkirim untuk divalidasi" dan beralih kehalaman validasi dan verifikasi data.	✓	

TC19	*Admin* menghapus data hasil *scan* secara *single*.	Sistem menampilkan pesan sukses "Data berhasil dihapus" dan data hasil *scan* berhasil dihapus dari daftar *scanning*.	Sistem menampilkan pesan sukses "Data berhasil dihapus" dan data hasil *scan* berhasil dihapus dari daftar *scanning*.	✓	
TC20	*Admin* menghapus seluruh data hasil *scan (delete all).*	Sistem menampilkan pesan sukses "Seluruh data berhasil dihapus" dan seluruh data hasil *scan* berhasil dihapus dari daftar *scanning*.	Sistem menampilkan pesan sukses "Seluruh data berhasil dihapus" dan seluruh data hasil *scan* berhasil dihapus dari daftar *scanning*.	✓	

Hasil Analisis:

1. *Persentase test case* yang lolos = $\frac{20}{20} = 100\%$
2. *Persentase test case* yang gagal = $\frac{0}{20} = 0\%$
3. *Persentase test case* yang telah diuji = $\frac{20}{20} = 100\%$
4. *Persentase test case* yang belum diuji = $\frac{0}{20} = 0\%$

Jumlah total *test case* dalam pengujian = 20

8. *Testing Use Case Scenario* Validasi & Verifikasi Data Hasil *Scan*

Pengujian *Scenario Use Case* Validasi & Verifikasi Data Hasil *Scan* dilakukan berdasarkan *Use Case Scenario* Validasi & Verifikasi Data Hasil *Scan* (Tabel 3.10). *Use case scenario* mendefinisikan proses *Admin* dalam validasi data hasil *scanning* perangkat atau material dengan memberikan status *approve* atau *reject* sebelum data disimpan sebagai data final *assets*. Pengujian dilakukan secara rinci untuk memvalidasi setiap proses mulai dari penampilan data hasil *scan*, pengecekan kesesuaian data, pemilihan aksi validasi, hingga proses penyimpanan data ke *database*. Hal ini bertujuan untuk memastikan proses validasi dan verifikasi berjalan sesuai dengan kebutuhan sistem dan skenario bisnis yang telah dirancang.

Tabel 4. 9 *Blackbox Testing* Validasi & Verifikasi Hasil *Scan*

Test Case ID	***Scenario***	**Luaran yang Diharapkan**	**Luaran yang Dihasilkan**	**Hasil**	
				B	**G**
TC01	*Admin* berada pada halaman *Validation & Verification* setelah proses *scanning* selesai.	Sistem menampilkan halaman *Validation & Verification* beserta data hasil *scan*.	Sistem menampilkan halaman *Validation & Verification* beserta data hasil *scan*.	✓	
TC02	Data hasil *scan* tersedia berupa data perangkat atau material beserta informasi *serial number* atau *scan code*.	Sistem menampilkan data hasil *scan* secara lengkap sesuai dengan hasil *scanning*.	Sistem menampilkan data hasil *scan* secara lengkap sesuai dengan hasil *scanning*.	✓	
TC03	Data hasil *scan* tidak tersedia termasuk data perangkat atau material beserta informasi *serial number* atau *scan code*.	Sistem menampilkan halaman *Validation & Verification* kosong dan meminta untuk melakukan pengecekan terlebih dahulu.	Sistem menampilkan halaman *Validation & Verification* kosong dan meminta untuk melakukan pengecekan terlebih dahulu.	✓	
TC04	Data hasil *scan* perangkat atau material beserta informasi *serial number* atau *scan code* lengkap dan sesuai.	Data ditampilkan dan dilakukan proses validasi.	Data ditampilkan dan dilakukan proses validasi.	✓	
TC05	Data hasil *scan* perangkat atau material beserta informasi *serial number* atau *scan code* tidak lengkap atau tidak sesuai.	Data ditampilkan dengan tidak lengkap dan sistem menampilkan pesan "Data tidak lengkap atau tidak valid" dan meminta pengecekan ulang.	Data ditampilkan dengan tidak lengkap dan sistem menampilkan pesan "Data tidak lengkap atau tidak valid" dan meminta pengecekan ulang.	✓	

TC06	*Admin* memeriksa kesesuaian data hasil *scan* perangkat atau material beserta *serial number* atau *scan code*.	Sistem menampilkan data dan status hasil deteksi *(session, photo, item, brand, code, status, submitted)* untuk divalidasi.	Sistem menampilkan data dan status hasil deteksi *(session, photo, item, brand, code, status, submitted)* untuk divalidasi.	✓	
TC07	*Admin* memilih aksi *approve* pada data hasil *scan* perangkat atau material beserta *serial number* atau *scan code* yang valid.	Sistem memproses validasi dan menampilkan pesan "Data berhasil di *approve*".	Sistem memproses validasi dan menampilkan pesan "Data berhasil di *approve*".	✓	
TC08	*Admin* memilih aksi *approve* pada data hasil *scan* perangkat atau material beserta *serial number* atau *scan code* yang valid.	Sistem menghasilkan kode unik hasil validasi untuk perangkat atau material.	Sistem menghasilkan kode unik hasil validasi untuk perangkat atau material.	✓	
TC09	*Admin* memilih aksi *approve* pada data hasil *scan* perangkat atau material beserta *serial number* atau *scan code* yang valid.	Data disimpan ke *database* sebagai data final *assets* dan ditampilkan pada halaman *assets*.	Data disimpan ke *database* sebagai data final *assets* dan ditampilkan pada halaman *assets*.	✓	
TC10	*Admin* memilih aksi *reject* pada data hasil *scan* perangkat atau material beserta *serial number* atau *scan code* yang tidak sesuai.	Data tidak disimpan sebagai *assets* dan diminta melakukan *scan* ulang atau *re-scan*.	Data tidak disimpan sebagai *assets* dan diminta melakukan *scan* ulang atau *re-scan*.	✓	

TC11	Proses validasi data hasil *scan* perangkat atau material beserta *serial number* atau *scan code* berhasil dilakukan *(approve).*	Sistem menampilkan notifikasi sukses “Data berhasil divalidasi dan disimpan”.	Sistem menampilkan notifikasi sukses “Data berhasil divalidasi dan disimpan”.	✓	
TC12	Proses validasi data hasil *scan* perangkat atau material beserta *serial number* atau *scan code* gagal dilakukan.	Sistem menampilkan pesan “Data tidak valid, periksa kembali”.	Sistem menampilkan pesan “Data tidak valid, periksa kembali”.	✓	
TC13	*Admin* memilih aksi view pada salah satu data hasil *scan* perangkat atau material beserta *serial number* atau *scan code*.	Sistem menampilkan detail informasi data hasil *scan* dalam bentuk modal *(popup)* secara lengkap.	Sistem menampilkan detail informasi data hasil *scan* dalam bentuk modal *(popup)* secara lengkap.	✓	
TC14	*Admin* menghapus data hasil validasi atau hasil *scan* perangkat atau material beserta *serial number* atau *scan code* tertentu.	Sistem menampilkan pesan “Data berhasil dihapus” dan data hilang dari daftar validasi.	Sistem menampilkan pesan “Data berhasil dihapus” dan data hilang dari daftar validasi.	✓	
TC15	*Admin* menghapus data hasil validasi atau hasil *scan* perangkat atau material beserta *serial number* atau *scan code* tertentu. tetapi terjadi kegagalan sistem.	Sistem menampilkan pesan “Gagal menghapus data, coba kembali”.	Sistem menampilkan pesan “Gagal menghapus data, coba kembali”.	✓	

Hasil Analisis:

1. *Persentase test case* yang lolos = $\frac{15}{15} = 100\%$
2. *Persentase test case* yang gagal = $\frac{0}{15} = 0\%$
3. *Persentase test case* yang telah diuji = $\frac{15}{15} = 100\%$
4. *Persentase test case* yang belum diuji = $\frac{0}{15} = 0\%$

Jumlah total *test case* dalam pengujian = 15

9. *Testing Use Case Scenario* Melihat Data *Assets*

Pengujian *Scenario Use Case* Melihat Data *Assets* dilakukan berdasarkan *Use Case Scenario* Melihat Data *Assets* (Tabel 3.11). *Use case scenario* mendefinisikan proses *Admin* dalam melihat data final *assets* (perangkat atau material) hasil *scanning* dan validasi yang telah tersimpan dalam sistem. Pengujian dilakukan secara rinci untuk memvalidasi setiap proses mulai dari akses menu *Assets Inventory*, pengambilan data aset dari *database*, penampilan daftar aset, hingga fitur pencarian dan filter data. Hal ini bertujuan untuk memastikan proses melihat data assets berjalan sesuai dengan kebutuhan sistem dan skenario bisnis yang telah dirancang.

Tabel 4. 10 *Blackbox Testing* Melihat Data *Assets*

Test Case ID	***Scenario***	**Luaran yang Diharapkan**	**Luaran yang Dihasilkan**	**Hasil**	
				B	**G**
TC01	*Admin* mengakses halaman *Assets Inventory*.	Sistem menampilkan halaman *Assets Inventory* beserta data *assets* dan struktur data aset berdasarkan hasil *scan* dan validasi.	Sistem menampilkan halaman *Assets Inventory* beserta data *assets* dan struktur data aset berdasarkan hasil *scan* dan validasi.	✓	
TC02	Sistem mengambil data aset berupa data perangkat atau material beserta informasi *serial number*	Sistem berhasil mengambil data aset dan menampilkannya dalam daftar.	Sistem berhasil mengambil data aset dan menampilkannya dalam daftar.	✓	

	atau *scan code* dari *database*.				
TC03	Sistem gagal mengambil data aset berupa data perangkat atau material beserta informasi *serial number* atau *scan code* dari *database*.	Sistem menampilkan pesan “Gagal memuat data aset, periksa koneksi”.	Sistem menampilkan pesan “Gagal memuat data aset, periksa koneksi”.	✓	
TC04	Data *assets* tersedia dari *database* berupa data perangkat atau material beserta informasi *serial number* atau *scan code*.	Sistem menampilkan data *assets* secara lengkap dan menampilkannya dalam daftar list *assets* berdasarkan *session preparation*.	Sistem menampilkan data *assets* secara lengkap dan menampilkannya dalam daftar list *assets* berdasarkan *session preparation*.	✓	
TC05	Data *assets* belum tersedia dari *database* dan data perangkat atau material beserta informasi *serial number* atau *scan code* tidak muncul.	Sistem menampilkan pesan “Data aset belum tersedia”.	Sistem menampilkan pesan “Data aset belum tersedia”.	✓	
TC06	Data detail *assets* ditampilkan berupa foto, kode aset, nama aset, kategori, *serial number/scan code*, lokasi, status, dan tanggal.	Sistem menampilkan Informasi detail aset ditampilkan secara lengkap sesuai data yang tersimpan pada sistem yaitu foto, kode aset, nama aset, kategori, *serial number/scan*	Sistem menampilkan Informasi detail aset ditampilkan secara lengkap sesuai data yang tersimpan pada sistem yaitu foto, kode aset, nama aset, kategori, *serial number/scan*	✓	

		*code,7*lokasi, status, dan tanggal.	*code,* lokasi, status, dan tanggal.		
TC07	*Admin* memilih salah satu data *session assets* untuk melihat detail *assets (view assets)*.	Sistem menampilkan detail seluruh list item-item *assets* berdasarkan *session* yang dipilih.	Sistem menampilkan detail seluruh list item-item *assets* berdasarkan *session* yang dipilih.	✓	
TC08	*Admin* melakukan pencarian data *assets* berdasarkan kata kunci tertentu.	Sistem menampilkan data *assets* sesuai hasil pencarian.	Sistem menampilkan data *assets* sesuai hasil pencarian.	✓	
TC09	*Admin* melakukan pencarian data *assets* berdasarkan kata kunci tertentu tetapi data tidak ditemukan.	Sistem menampilkan halaman *assets* dimana Data tidak ditemukan sesuai kriteria.	Sistem menampilkan halaman *assets* dimana Data tidak ditemukan sesuai kriteria.	✓	
TC10	*Admin* melakukan *filter* data *assets* berdasarkan kategori (perangkat atau material) atau *session preparation* tertentu.	Sistem menampilkan data *assets* sesuai filter yang dipilih.	Sistem menampilkan data *assets* sesuai filter yang dipilih.	✓	
TC11	*Admin* melakukan *filter* data *assets* berdasarkan kategori (perangkat atau material) atau *session preparation* tertentu tetapi data tidak ditemukan.	Sistem menampilkan halaman *assests* dimana Data tidak ditemukan sesuai kriteria.	Sistem menampilkan halaman *assests* dimana Data tidak ditemukan sesuai kriteria.	✓	
TC12	*Admin* melakukan *filter* data *assets* berdasarkan *checking number session preparation* tertentu.	Sistem menampilkan data *assets* sesuai *checking number session preparaion*.	Sistem menampilkan data *assets* sesuai *checking number session preparaion*.	✓	

TC13	*Admin* melakukan *filter* data *assets* berdasarkan *checking number session preparation* tertentu tetapi tidak ditemukan.	Sistem menampilkan Data tidak ditemukan sesuai *checking number*.	Sistem menampilkan Data tidak ditemukan sesuai *checking number*.	✓	

Hasil Analisis:

1. *Persentase test case* yang lolos = $\frac{13}{13} = 100\%$
2. *Persentase test case* yang gagal = $\frac{0}{13} = 0\%$
3. *Persentase test case* yang telah diuji = $\frac{13}{13} = 100\%$
4. *Persentase test case* yang belum diuji = $\frac{0}{13} = 0\%$

Jumlah total *test case* dalam pengujian = 13

10. *Testing Use Case Scenario* Laporan Pengecekan Inventaris

Pengujian *Scenario Use Case* Laporan Pengecekan Inventaris dilakukan berdasarkan *Use Case Scenario* Laporan Pengecekan Inventaris (Tabel 3.12). *Use case scenario* mendefinisikan proses *Admin* dan *Superadmin* dalam melihat serta mengunduh laporan hasil pengecekan inventaris perangkat atau material berdasarkan periode tertentu dalam format *excel*. Pengujian dilakukan secara rinci untuk memvalidasi setiap proses mulai dari akses menu *Report & Analytics*, pemilihan periode laporan (harian/mingguan), pengambilan data pengecekan dari *database*, hingga proses pengunduhan laporan dalam format *excel*. Hal ini bertujuan untuk memastikan proses laporan pengecekan inventaris berjalan sesuai dengan kebutuhan sistem dan skenario bisnis yang telah dirancang.

Tabel 4. 11 *Blackbox Testing* Laporan Pengecekan Inventaris

Test Case ID	***Scenario***	**Luaran yang Diharapkan**	**Luaran yang Dihasilkan**	**Hasil**	
				B	**G**

TC01	*Admin/Superadmin* mengakses halaman *Dashboard* dan memilih menu *Reports & Analytics*.	Sistem menampilkan halaman *Reports & Analytics* dan berisikan data laporan pengecekan inventaris seluruh data aset berdasarkan hasil *scan* dan validasi.	Sistem menampilkan halaman *Reports & Analytics* dan berisikan data laporan pengecekan inventaris seluruh data aset berdasarkan hasil *scan* dan validasi.	✓	
TC02	Sistem mengambil data laporan pengecekan inventaris aset berupa data perangkat atau material beserta informasi *serial number* atau *scan code* dari *database*.	Sistem berhasil mengambil data laporan pengecekan inventaris aset dan menampilkannya dalam daftar list laporan berdasarkan periode.	Sistem berhasil mengambil data laporan pengecekan inventaris aset dan menampilkannya dalam daftar list laporan berdasarkan periode.	✓	
TC03	Sistem gagal mengambil laporan pengecekan inventaris aset berupa data perangkat atau material beserta informasi *serial number* atau *scan code* dari *database*.	Sistem menampilkan pesan "Gagal memuat data laporan pengecekan aset, periksa koneksi".	Sistem menampilkan pesan "Gagal memuat data laporan pengecekan aset, periksa koneksi".	✓	
TC04	Data laporan pengecekan inventaris aset berhasil diakses.	Sistem menampilkan opsi periode laporan pengecekan berupa harian dan mingguan.	Sistem menampilkan opsi periode laporan pengecekan berupa harian dan mingguan.	✓	
TC05	*Admin/Superadmin* memilih salah satu periode laporan pengecekan inventaris.	Sistem menerima input periode dan memproses permintaan pengambilan data laporan.	Sistem menerima input periode dan memproses permintaan pengambilan data laporan.	✓	

TC06	Data laporan pengecekan aset perangkat atau material beserta informasi *serial number* atau *scan code* tersedia sesuai periode yang dipilih.	Sistem berhasil mengambil data pengecekan perangkat atau material dari *database* dan menampilkan laporan sesuai periode.	Sistem berhasil mengambil data pengecekan perangkat atau material dari *database* dan menampilkan laporan sesuai periode.	✓	
TC07	Data laporan pengecekan aset perangkat atau material beserta informasi *serial number* atau *scan code* tidak tersedia sesuai periode yang dipilih.	Sistem menampilkan halaman *Report & Analytics* dengan Tidak ada data laporan pada periode ini.	Sistem menampilkan halaman *Report & Analytics* dengan Tidak ada data laporan pada periode ini.	✓	
TC08	*Admin/Superadmin* memilih salah satu laporan dari daftar periode yang dipilih.	Sistem menampilkan detail laporan pengecekan inventaris yang berisi data perangkat atau material secara lengkap.	Sistem menampilkan detail laporan pengecekan inventaris yang berisi data perangkat atau material secara lengkap.	✓	
TC09	Data detail laporan pengecekan inventaris tidak tersedia atau gagal dimuat.	Sistem menampilkan halaman *Report & Analytics* dengan informasi Data detail laporan tidak tersedia.	Sistem menampilkan halaman *Report & Analytics* dengan informasi Data detail laporan tidak tersedia.	✓	
TC10	Sistem melakukan *generate* laporan pengecekan inventaris.	Sistem menghasilkan file laporan pengecekan inventaris aset perangkat atau material berdasarkan periode yang dipilih.	Sistem menghasilkan file laporan pengecekan inventaris aset perangkat atau material berdasarkan periode yang dipilih.	✓	

TC11	*Admin/Superadmin* melakukan proses unduh laporan pengecekan inventaris dalam format *excel*.	Sistem memproses permintaan unduh dan menghasilkan file laporan dalam format *excel*.	Sistem memproses permintaan unduh dan menghasilkan file laporan dalam format *excel*.	✓	
TC12	Proses unduh laporan pengecekan inventaris berhasil dilakukan.	Sistem menampilkan file laporan pengecekan inventaris dalam format *excel*.	Sistem menampilkan file laporan pengecekan inventaris dalam format *excel*.	✓	
TC13	Proses unduh laporan pengecekan inventaris gagal akibat koneksi bermasalah.	Sistem menampilkan pesan "Gagal mengunduh laporan, cek koneksi".	Sistem menampilkan pesan "Gagal mengunduh laporan, cek koneksi".	✓	

Hasil Analisis:

1. *Persentase test case* yang lolos = $\frac{13}{13} = 100\%$
2. *Persentase test case* yang gagal = $\frac{0}{13} = 0\%$
3. *Persentase test case* yang telah diuji = $\frac{13}{13} = 100\%$
4. *Persentase test case* yang belum diuji = $\frac{0}{13} = 0\%$

Jumlah total *test case* dalam pengujian = 13

11. *Testing Use Case Scenario Monitoring Real-time*

Pengujian *Scenario Use Case Monitoring Real-Time* dilakukan berdasarkan *Use Case Scenario Monitoring Real-Time* (Tabel 3.13). *Use case scenario* ini mendefinisikan proses *Superadmin* dalam memantau aktivitas pengecekan inventaris perangkat atau material secara *real-time*. Pengujian dilakukan untuk memvalidasi proses pengambilan data inventaris, penampilan ringkasan aset, status validasi *(approve/reject)*, grafik *monitoring*, serta aktivitas *scanning* dan validasi terbaru dengan benar sesuai kebutuhan sistem.

Tabel 4. 12 *Blackbox Testing Monitoring Real-time*

Test Case ID	***Scenario***	**Luaran yang Diharapkan**	**Luaran yang Dihasilkan**	**Hasil**	
				B	**G**
TC01	*Superadmin* mengakses dan berada pada halaman *Dashboard*.	Sistem menampilkan halaman *Dashboard* dan berisikan data *monitoring* seluruh data aset berdasarkan hasil *scan* dan validasi secara *real-time*.	Sistem menampilkan halaman *Dashboard* dan berisikan data *monitoring* seluruh data aset berdasarkan hasil *scan* dan validasi secara *real-time*.	✓	
TC02	Sistem mengambil seluruh data *monitoring* inventaris aset dari *database*.	Sistem berhasil mengambil data inventaris aset perangkat atau material dari *database*.	Sistem berhasil mengambil data inventaris aset perangkat atau material dari *database*.	✓	
TC03	Sistem gagal mengambil seluruh data *monitoring* inventaris aset dari *database*.	Sistem menampilkan pesan "Gagal memuat data, periksa koneksi jaringan Anda".	Sistem menampilkan pesan "Gagal memuat data, periksa koneksi jaringan Anda".	✓	
TC04	Data *monitoring* inventaris aset tersedia dalam *database*.	Sistem menampilkan ringkasan jumlah seluruh aset perangkat atau material.	Sistem menampilkan ringkasan jumlah seluruh aset perangkat atau material.	✓	
TC05	Data *monitoring* inventaris aset tidak tersedia dalam *database*.	Sistem menampilkan halaman *Dashboard* dengan belum ada data perangkat atau material pada sistem.	Sistem menampilkan halaman *Dashboard* dengan belum ada data perangkat atau material pada sistem.	✓	

TC06	Data *monitoring* status validasi hasil *scanning* aset tersedia.	Sistem menampilkan status validasi *(approve/reject)* pada *dashboard*.	Sistem menampilkan status validasi *(approve/reject)* pada *dashboard*.	✓	
TC07	Data *monitoring* status validasi hasil *scanning* aset tidak tersedia.	Sistem menampilkan halaman *Dashboard* dengan informasi bahwa belum ada data validasi yang tersedia.	Sistem menampilkan halaman *Dashboard* dengan informasi bahwa belum ada data validasi yang tersedia.	✓	
TC08	Data grafik *monitoring* inventaris aset tersedia.	Sistem menampilkan grafik *monitoring* inventaris aset secara *real-time*.	Sistem menampilkan grafik *monitoring* inventaris aset secara *real-time*.	✓	
TC09	Data grafik *monitoring* inventaris aset gagal dimuat.	Sistem menampilkan Data grafik kosong dan akan muncul setelah terdapat aset.	Sistem menampilkan Data grafik kosong dan akan muncul setelah terdapat aset.	✓	
TC10	Data aktivitas *scanning* aset dan validasi terbaru tersedia.	Sistem menampilkan aktivitas *scanning* dan validasi terbaru secara *real-time.*	Sistem menampilkan aktivitas *scanning* dan validasi terbaru secara *real-time.*	✓	
TC11	Data aktivitas *scanning* aset dan validasi tidak tersedia.	Sistem menampilkan halaman *Dashboard* dengan belum ada aktivitas pengecekan.	Sistem menampilkan halaman *Dashboard* dengan belum ada aktivitas pengecekan.	✓	
TC12	*Superadmin* melakukan filter data berdasarkan periode, status, atau kategori.	Sistem menampilkan data *monitoring* sesuai filter yang dipilih.	Sistem menampilkan data *monitoring* sesuai filter yang dipilih.	✓	

TC13	*Superadmin* melakukan filter data berdasarkan periode, status, atau kategori tetapi hasil filter tidak ditemukan.	Sistem menampilkan pesan "Tidak ditemukan data sesuai kriteria pencarian".	Sistem menampilkan pesan "Tidak ditemukan data sesuai kriteria pencarian".	✓	
TC14	Sistem melakukan pembaruan data secara *real-time*.	Sistem memperbarui data *monitoring* inventaris secara otomatis tanpa *refresh* halaman.	Sistem memperbarui data *monitoring* inventaris secara otomatis tanpa *refresh* halaman.	✓	

Hasil Analisis:

1. *Persentase test case* yang lolos = $\frac{14}{14} = 100\%$
2. *Persentase test case* yang gagal = $\frac{0}{14} = 0\%$
3. *Persentase test case* yang telah diuji = $\frac{14}{14} = 100\%$
4. *Persentase test case* yang belum diuji = $\frac{0}{14} = 0\%$

Jumlah total *test case* dalam pengujian = 14

12. *Testing Use Case Scenario* Verifikasi Laporan Pengecekan

Pengujian *Scenario Use Case* Verifikasi Laporan Pengecekan dilakukan berdasarkan *Use Case Scenario* Verifikasi Laporan Pengecekan (Tabel 3.14). Pengujian ini bertujuan untuk memastikan proses verifikasi laporan inventaris oleh *Superadmin* berjalan dengan benar, mulai dari melihat daftar laporan, memeriksa detail laporan, hingga memberikan persetujuan atau revisi sesuai kebutuhan sistem.

Tabel 4. 13 *Blackbox Testing* Verifikasi Laporan Pengecekan

Test Case ID	***Scenario***	**Luaran yang Diharapkan**	**Luaran yang Dihasilkan**	**Hasil**	
				B	**G**

TC01	*Superadmin* berada pada *Dashboard* dan mengakses menu *Report & Analytics*	Sistem menampilkan halaman *Report & Analytics* beserta daftar seluruh laporan pengecekan inventaris aset.	Sistem menampilkan halaman *Report & Analytics* beserta daftar seluruh laporan pengecekan inventaris aset.	✓	
TC02	Sistem mengambil seluruh data laporan pengecekan inventaris aset dari *database*.	Sistem berhasil mengambil data laporan pengcekan inventaris aset perangkat atau material dari *database*.	Sistem berhasil mengambil data laporan pengcekan inventaris aset perangkat atau material dari *database*.	✓	
TC03	Sistem gagal mengambil seluruh data laporan pengecekan inventaris aset dari *database*.	Sistem menampilkan pesan "Gagal memuat data laporan, periksa koneksi jaringan Anda".	Sistem menampilkan pesan "Gagal memuat data laporan, periksa koneksi jaringan Anda".	✓	
TC04	Data laporan pengecekan inventaris aset tersedia dalam *database*.	Sistem menampilkan seluruh laporan pengecekan inventaris aset perangkat atau material yang dapat dipilih.	Sistem menampilkan seluruh laporan pengecekan inventaris aset perangkat atau material yang dapat dipilih.	✓	
TC05	Data laporan pengecekan inventaris aset tersedia dalam *database*.	Sistem menampilkan halaman *Dashboard* dengan belum ada data laporan pengecekan inventaris aset perangkat atau material pada sistem.	Sistem menampilkan halaman *Dashboard* dengan belum ada data laporan pengecekan inventaris aset perangkat atau material pada sistem.	✓	
TC06	Data laporan tidak lengkap (ada data	Sistem menampilkan pesan "Data perangkat	Sistem menampilkan pesan "Data perangkat	✓	

	perangkat/material yang belum lengkap).	atau material belum lengkap".	atau material belum lengkap".		
TC07	*Superadmin* memeriksa isi laporan pengecekan inventaris aset.	Sistem menampilkan detail data laporan untuk dilakukan verifikasi.	Sistem menampilkan detail data laporan untuk dilakukan verifikasi.	✓	
TC08	*Superadmin* memberikan persetujuan *(approve)* pada laporan pengecekan inventaris aset.	Sistem menyimpan status laporan menjadi disetujui dan menampilkan pesan "Verifikasi berhasil".	Sistem menyimpan status laporan menjadi disetujui dan menampilkan pesan "Verifikasi berhasil".	✓	
TC09	*Superadmin* melakukan revisi laporan pengecekan inventaris aset.	Sistem menampilkan status laporan menjadi revisi dan mengembalikan ke *admin* untuk pengecekan ulang.	Sistem menampilkan status laporan menjadi revisi dan mengembalikan ke *admin* untuk pengecekan ulang.	✓	
TC10	*Superadmin* membatalkan proses verifikasi laporan pengecekan inventaris aset.	Sistem tidak menyimpan perubahan dan kembali ke halaman sebelumnya.	Sistem tidak menyimpan perubahan dan kembali ke halaman sebelumnya.	✓	
TC11	Proses verifikasi laporan pengecekan inventaris aset berhasil dilakukan.	Sistem menampilkan notifikasi "Verifikasi laporan berhasil".	Sistem menampilkan notifikasi "Verifikasi laporan berhasil".	✓	
TC12	Proses verifikasi laporan pengecekan inventaris aset gagal dilakukan.	Sistem menampilkan pesan "Verifikasi gagal, coba kembali".	Sistem menampilkan pesan "Verifikasi gagal, coba kembali".	✓	

Hasil Analisis:

1. *Persentase test case* yang lolos = $\frac{12}{12} = 100\%$
2. *Persentase test case* yang gagal = $\frac{0}{12} = 0\%$
3. *Persentase test case* yang telah diuji = $\frac{12}{12} = 100\%$

4. *Persentase test case* yang belum diuji = $\frac{0}{12} = 0\%$

Jumlah total *test case* dalam pengujian = 12

13. *Testing Use Case Scenario* Manajemen Hak Akses Pengguna

Pengujian *Scenario Use Case* Manajemen Hak Akses Pengguna dilakukan berdasarkan *Use Case Scenario* Manajemen Hak Akses Pengguna (Tabel 3.15). *Use case scenario* mendefinisikan proses *Superadmin* dalam mengelola data pengguna sistem seperti melihat daftar pengguna, melihat detail pengguna, mengubah *role* pengguna, serta melakukan reset *password*. Pengujian dilakukan secara lebih rinci untuk memastikan setiap proses manajemen hak akses berjalan sesuai dengan kebutuhan sistem dan skenario bisnis yang telah dirancang.

Tabel 4. 14 *Blackbox Testing* Manajemen Hak Akses Pengguna

Test Case ID	***Scenario***	**Luaran yang Diharapkan**	**Luaran yang Dihasilkan**	**Hasil**	
				B	**G**
TC01	*Superadmin* berada pada *Dashboard* dan mengakses menu *Users Management*	Sistem menampilkan halaman *Users Management* beserta daftar seluruh sistem.	Sistem menampilkan halaman *Users Management* beserta daftar seluruh sistem.	✓	
TC02	Sistem mengambil seluruh data seluruh pengguna sistem dari *database*.	Sistem berhasil mengambil data seluruh pengguna sistem dari *database*.	Sistem berhasil mengambil data laporan seluruh pengguna sistem dari *database*.	✓	
TC03	Sistem gagal mengambil seluruh pengguna sistem dari *database*.	Sistem menampilkan pesan "Gagal memuat data pengguna, periksa koneksi jaringan Anda".	Sistem menampilkan pesan "Gagal memuat data pengguna, periksa koneksi jaringan Anda".	✓	
TC04	Data pengguna sistem tersedia di *database*.	Sistem menampilkan halaman *Users Management* dengan	Sistem menampilkan halaman *Users Management* dengan	✓	

		data seluruh pengguna sistem.	data seluruh pengguna sistem.		
TC05	Data pengguna sistem tidak tersedia di *database*.	Sistem menampilkan halaman *Users Management* dengan tidak ada data pengguna.	Sistem menampilkan halaman *Users Management* dengan tidak ada data pengguna.	✓	
TC06	*Superadmin* memilih data salah satu pengguna sistem.	Sistem menampilkan detail data pengguna.	Sistem menampilkan detail data pengguna.	✓	
TC07	Sistem gagal menampilkan detail data pengguna sistem.	Sistem menampilkan pesan "Gagal memuat data pengguna".	Sistem menampilkan pesan "Gagal memuat data pengguna".	✓	
TC08	*Superadmin* memilih mengubah *role* pengguna dengan menggunakan data valid.	Sistem menampilkan *form* untuk merubah *role* pengguna.	Sistem menampilkan *form* untuk merubah *role* pengguna.	✓	
TC09	*Superadmin* melakukan perubahan pada *role* pengguna dengan mengganti *role* pada *field role* pengguna.	Sistem menyimpan perubahan *role* dan menampilkan pesan sukses "Berhasil merubah *role* data pengguna".	Sistem menyimpan perubahan *role* dan menampilkan pesan sukses "Berhasil merubah *role* data pengguna".	✓	
TC10	*Superadmin* tidak melakukan perubahan pada *role* pengguna dan membatalkan aksi.	Sistem tidak menyimpan perubahan dan kembali ke halaman sebelumnya.	Sistem tidak menyimpan perubahan dan kembali ke halaman sebelumnya.	✓	
TC11	Superadmin melakukan perubahan data *role* pengguna tanpa koneksi internet.	Sistem menampilkan pesan "Gagal menyimpan perubahan *role*, periksa koneksi".	Sistem menampilkan pesan "Gagal menyimpan perubahan *role*, periksa koneksi".	✓	

TC12	*Superadmin* memilih reset *password* pengguna dengan menggunakan data valid.	Sistem menampilkan *form* untuk mereset *password* pengguna.	Sistem menampilkan *form* untuk mereset *password* pengguna.	✓	
TC13	*Superadmin* melakukan reset *password* pengguna dengan mengisi *field password* pengguna dengan *password* baru.	Sistem menyimpan data reset *password* pengguna dan menampilkan pesan sukses "Berhasil reset *password* data pengguna".	Sistem menyimpan data reset *password* pengguna dan menampilkan pesan sukses "Berhasil reset *password* data pengguna".	✓	
TC14	*Superadmin* tidak melakukan reset *password* pengguna dan membatalkan aksi.	Sistem tidak menyimpan perubahan dan kembali ke halaman sebelumnya.	Sistem tidak menyimpan perubahan dan kembali ke halaman sebelumnya.	✓	
TC15	Superadmin melakukan reset *password* pengguna tanpa koneksi internet.	Sistem menampilkan pesan "Gagal reset *password*, periksa koneksi".	Sistem menampilkan pesan "Gagal reset *password*, periksa koneksi".	✓	

Hasil Analisis:

1. *Persentase test case* yang lolos = $\frac{15}{15} = 100\%$
2. *Persentase test case* yang gagal = $\frac{0}{15} = 0\%$
3. *Persentase test case* yang telah diuji = $\frac{15}{15} = 100\%$
4. *Persentase test case* yang belum diuji = $\frac{0}{15} = 0\%$

Jumlah total *test case* dalam pengujian = 15

14. *Testing Use Case Scenario Logout*

Pengujian Scenario *Use Case Logout* dilakukan berdasarkan *Use Case Scenario Logout* (Tabel 3.16). *Use case scenario* ini mendefinisikan proses

pengguna dalam keluar dari sistem setelah menyelesaikan aktivitas. Pengujian dilakukan untuk memvalidasi proses *logout* mulai dari akses menu *profile*, konfirmasi *logout*, hingga penghapusan *session* dan pengalihan ke halaman *login*, sehingga memastikan keamanan sistem dan mencegah akses tanpa autentikasi.

Tabel 4. 15 *Blackbox Testing Logout*

Test Case ID	***Scenario***	**Luaran yang Diharapkan**	**Luaran yang Dihasilkan**	**Hasil**	
				B	**G**
TC01	*Admin/Superadmin* mengakses menu *Profile* sistem.	Sistem menampilkan halaman *Profile* pengguna.	Sistem menampilkan halaman *Profile* pengguna.	✓	
TC02	*Admin/Superadmin* memilih opsi *Logout* pada menu *profile.*	Sistem menampilkan modal dalam bentuk *pop up* untuk konfirmasi *logout* dengan pesan "Apakah anda yakin melakukan *Logout?*".	Sistem menampilkan modal dalam bentuk *pop up* untuk konfirmasi *logout* dengan pesan "Apakah anda yakin melakukan *Logout?*".	✓	
TC03	*Admin/Superadmin* melihat tampilan konfirmasi *Logout*.	Sistem menampilkan pilihan "*Yes, Logout*" dan "*Cancel*" untuk konfirmasi *logout*.	Sistem menampilkan pilihan "*Yes, Logout*" dan "*Cancel*" untuk konfirmasi *logout*.	✓	
TC04	*Admin/Superadmin* menekan tombol *"Yes, Logout"*.	Sistem menghapus *session login* pengguna dan mengarahkan ke halaman *login*.	Sistem menghapus *session login* pengguna dan mengarahkan ke halaman *login*.	✓	
TC05	*Admin/Superadmin* berhasil melakukan *Logout* dari sistem.	Sistem menampilkan halaman *login* dan pengguna harus *login* kembali untuk mengakses sistem.	Sistem menampilkan halaman *login* dan pengguna harus *login* kembali untuk mengakses sistem.	✓	

TC06	*Admin/Superadmin* menekan tombol *"Cancel"*.	Sistem membatalkan proses *logout* dan tetap mempertahankan *session* aktif.	Sistem membatalkan proses *logout* dan tetap mempertahankan *session* aktif.	✓	
TC07	*Admin/Superadmin* menekan tombol *"Cancel"* dan kembali kesistem.	Sistem menutup modal *pop up* konfirmasi *logout* dan tetap berada pada halaman sebelumnya.	Sistem menutup modal *pop up* konfirmasi *logout* dan tetap berada pada halaman sebelumnya.	✓	
TC08	*Admin/Superadmin* memilih opsi *logout* menggunakan tombol *logout* pada *dropdown profile*.	Sistem menampilkan modal dalam bentuk *pop up* untuk konfirmasi *logout* dengan pesan "Apakah anda yakin melakukan *Logout?*".	Sistem menampilkan modal dalam bentuk *pop up* untuk konfirmasi *logout* dengan pesan "Apakah anda yakin melakukan *Logout?*".	✓	
TC09	*Admin/Superadmin* menekan tombol *"Yes, Logout"* pada modal *pop up* pada *dropdown profile*.	Sistem menghapus *session login* pengguna dan mengarahkan ke halaman *login*.	Sistem menghapus *session login* pengguna dan mengarahkan ke halaman *login*.	✓	
TC10	*Admin/Superadmin* menekan tombol *"Cancel"* pada modal *pop up* pada *dropdown profile*.	Sistem membatalkan proses *logout* dan tetap mempertahankan *session* aktif.	Sistem membatalkan proses *logout* dan tetap mempertahankan *session* aktif.	✓	
TC11	*Admin/Superadmin* mengakses halaman setelah *logout*.	Sistem melakukan *redirect* kehalaman *login* kembali sebelum mengakses halaman.	Sistem melakukan *redirect* kehalaman *login* kembali sebelum mengakses halaman.	✓	
TC12	*Admin/Superadmin* mencoba mengakses	Sistem menolak akses karena *session* sudah berakhir dan melakukan	Sistem menolak akses karena *session* sudah berakhir dan melakukan	✓	

	halaman menggunakan *session* lama setelah *logout*.	*redirect* kehalaman *login*.	*redirect* kehalaman *login*.		
TC13	*Admin/Superadmin* melakukan proses *logout* saat koneksi tidak stabil.	Sistem tetap menghapus *session* lokal dan mengarahkan ke halaman *login* atau menampilkan pesan *error* jika gagal.	Sistem tetap menghapus *session* lokal dan mengarahkan ke halaman *login* atau menampilkan pesan *error* jika gagal.	✓	

Hasil Analisis:

1. *Persentase test case* yang lolos = $\frac{13}{13} = 100\%$
2. *Persentase test case* yang gagal = $\frac{0}{13} = 0\%$
3. *Persentase test case* yang telah diuji = $\frac{13}{13} = 100\%$
4. *Persentase test case* yang belum diuji = $\frac{0}{13} = 0\%$

Jumlah total *test case* dalam pengujian = 13

4.2.2 Pengujian Perangkat Lunak *(Non-Functional Testing)*

Tahap pengujian non-fungsional dilakukan untuk memastikan sistem berjalan dengan baik dari sisi performa, keamanan, dan keandalan. Pengujian ini tidak berfokus pada fitur, tetapi pada kualitas sistem seperti kecepatan respon, keamanan data, serta kestabilan sistem saat digunakan oleh banyak pengguna.

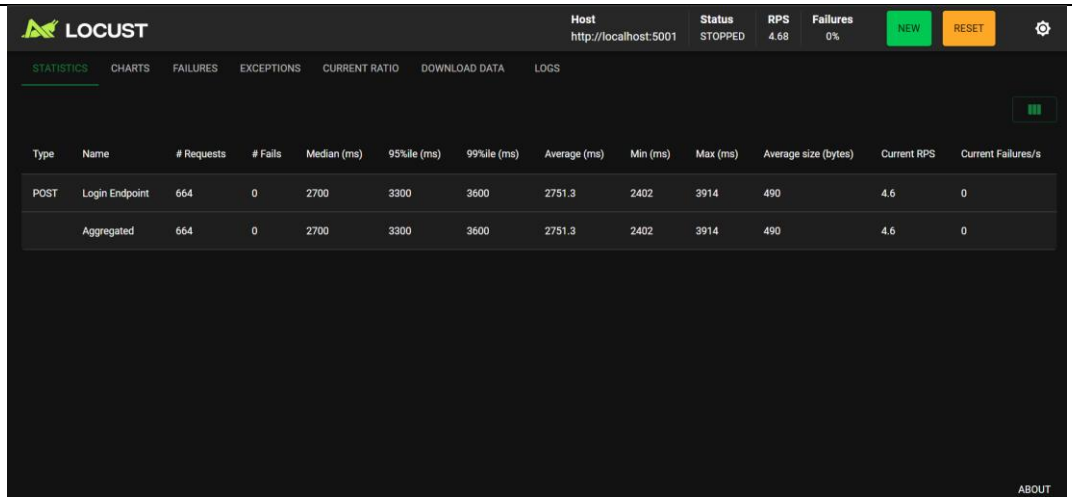
1. *Performance Testing*

   Pada tahap ini dilakukan pengujian *non-fungsional* untuk memastikan bahwa sistem memenuhi kebutuhan *non-fungsional* pada aspek performa sistem. Pengujian ini difokuskan pada kemampuan sistem dalam menangani banyak pengguna secara bersamaan, menjaga waktu respons tetap cepat, serta memastikan sistem tetap stabil saat menerima beban akses yang meningkat. Pengujian ini disusun berdasarkan kebutuhan *non-fungsional* NF-01 (Tabel 3.2), yaitu sistem harus memiliki performa yang baik, mampu menangani banyak pengguna secara bersamaan, tanpa mengalami penurunan performa.

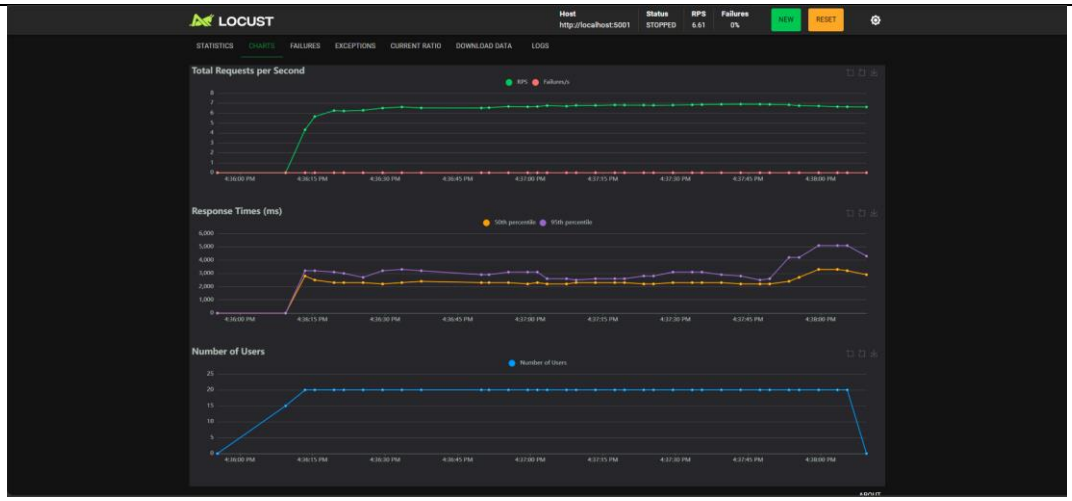
Tabel 4. 16 *Performance Testing*

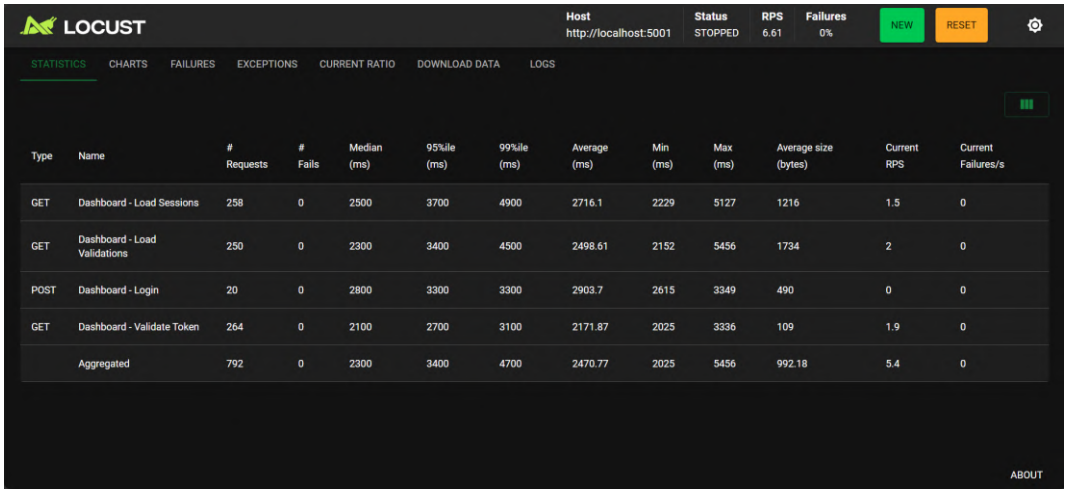
Komponen	**Rincian**
Tes Case ID	TC01-NF01
Type	*Performance Testing*
Judul	Sistem harus memiliki performa yang baik, mampu menangani banyak pengguna secara bersamaan, serta mampu memberikan waktu respons yang cepat dan konsisten pada setiap permintaan pengguna tanpa mengalami penurunan performa yang signifikan.
Tujuan	Memastikan sistem dapat beroperasi dengan baik di bawah berbagai beban kerja, mampu melayani banyak pengguna secara bersamaan, mampu memberikan waktu respons yang cepat dan konsisten pada setiap permintaan pengguna tanpa mengalami penurunan performa yang signifikan.
Lingkungan	Server pengujian menggunakan spesifikasi *hardware* dan *software* yang menyerupai lingkungan implementasi sistem, meliputi *framework* aplikasi, model *AI* yaitu *YOLO* dan *OCR*), serta basis data *PostgreSQL*. Seluruh komponen dikonfigurasi secara terintegrasi, sementara pengujian dilakukan melalui jaringan lokal/*intranet* dengan simulasi akses multi pengguna secara bersamaan
Alat	*Locust*
Metode	1. Simulasi aktivitas pengguna pada fitur utama sistem seperti *login*, membuka *dashboard*, membuat *scanning preparation*, melakukan *scanning aset*, ekstraksi *serial number/scan code*, validasi hasil *scan*, melihat data *assets* dan *reset password*. 2. Uji *load testing* dilakukan secara bertahap dengan beban 10 *user*, 20 *user*, dan 30 *user* bersamaan. 3. Mencatat *response time*, jumlah *request* berhasil, jumlah *request* gagal, dan *error rate*.
Ekspektasi	Sistem mampu merespons permintaan pengguna secara stabil dengan waktu respons yang cepat dan konsisten, tanpa mengalami penurunan kinerja, serta tidak terjadi *crash* dengan tingkat *error* yang rendah selama pengujian berlangsung.
Hasil Uji	

ID	LOT01	
Nama Pengujian	*Load Test – Endpoint Login*	
Deskripsi Pengujian	Pengujian dilakukan pada *endpoint login* untuk mengukur waktu respons, jumlah *request* yang diproses, dan kestabilan sistem saat menerima banyak permintaan *login* secara bersamaan.	
Method	*POST*	
Skenario Uji	Total *Users*	20
	Ramp Up	5
	Time (Durasi)	2 Menit
Response Time (ms)	*Min (ms)*	2402
	Average (ms)	2751.3 *ms*
	Max (ms)	3914 *ms*
	95%ile *(ms)*	3300 *ms*
Requests	Total R*equests*	664
	Success Requests	664
	Failed Requests	0
Error Rate (%)	0%	
Chart Pengujian		



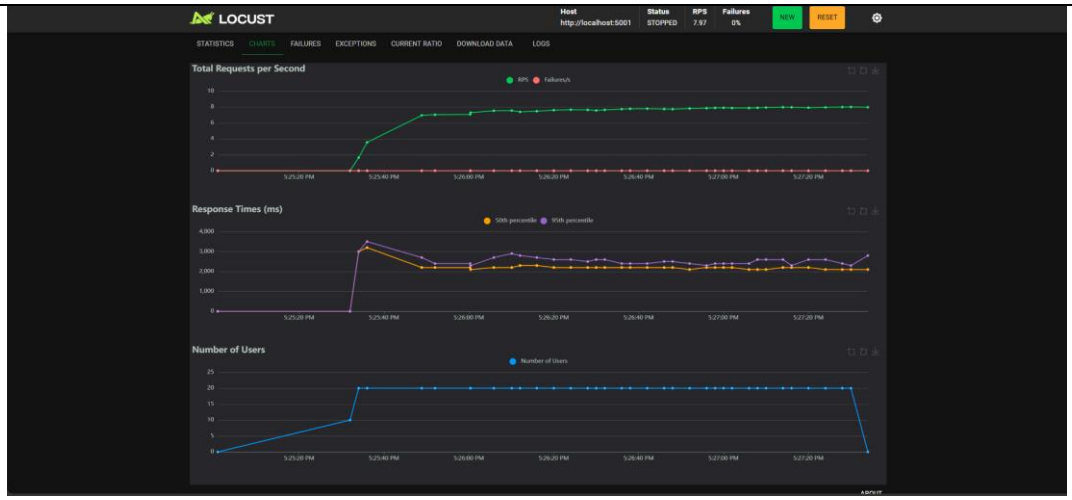
Hasil Uji		
ID	LOT02	
Nama Pengujian	*Load Test – Endpoint Dashboard*	
Deskripsi Pengujian	Pengujian dilakukan pada pembukaan *Dashboard* untuk mengukur waktu respons, jumlah *request*, dan kestabilan sistem saat banyak pengguna mengakses secara bersamaan, dengan simulasi *login*, validasi token, serta pengambilan data *scanning preparation* dan validasi.	
Method	*GET/POST*	
Skenario Uji	Total *Users*	20
	Ramp Up	5
	Time (Durasi)	2 Menit
Response Time (ms)	*Min (ms)*	2025 *ms*
	Average (ms)	2470.77
	Max (ms)	5456 *ms*
	95%ile *(ms)*	3400
Requests	Total *Requests*	792
	Success Requests	792
	Failed Requests	0
Error Rate (%)	0%	

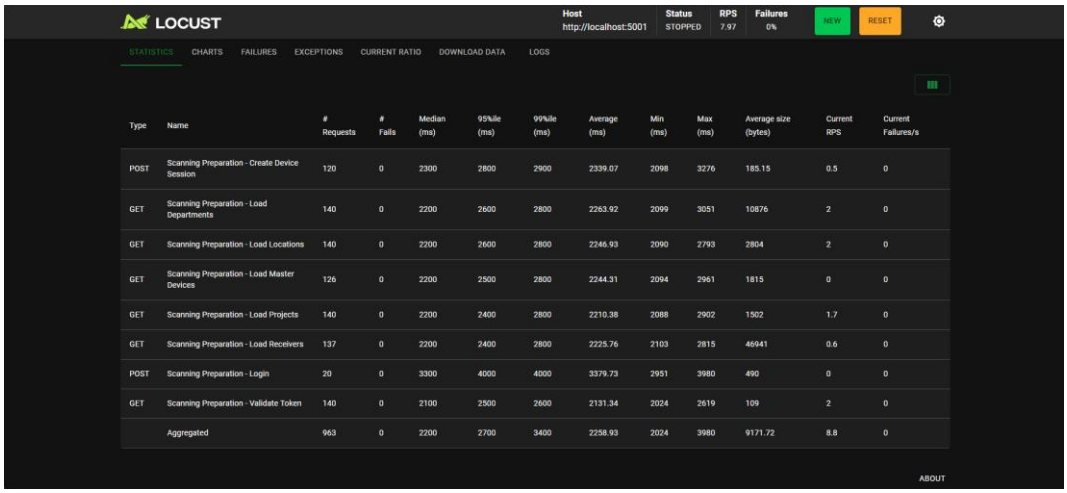
Chart Pengujian	



Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	Dashboard - Load Sessions	258	0	2500	3700	4900	2716.1	2229	5127	1216	1.5	0
GET	Dashboard - Load Validations	250	0	2300	3400	4500	2498.61	2152	5456	1734	2	0
POST	Dashboard - Login	20	0	2800	3300	3300	2903.7	2615	3349	490	0	0
GET	Dashboard - Validate Token	264	0	2100	2700	3100	2171.87	2025	3336	109	1.9	0
	Aggregated	792	0	2300	3400	4700	2470.77	2025	5456	992.18	5.4	0

Hasil Uji		
ID	LOT03	
Nama Pengujian	*Load Test – Endpoint Scanning Preparation*	
Deskripsi Pengujian	Pengujian dilakukan pada proses pembuatan *scanning preparation* untuk mengukur waktu respons, jumlah *request*, dan kestabilan sistem saat banyak pengguna membuat *session scanning* secara bersamaan. Pengujian mensimulasikan proses *login*, pengambilan data *master* pendukung *form*, serta pengiriman data *scanning preparation* baru ke sistem.	
Method	*GET/POST*	
Skenario Uji	Total *Users*	20
	Ramp Up	5

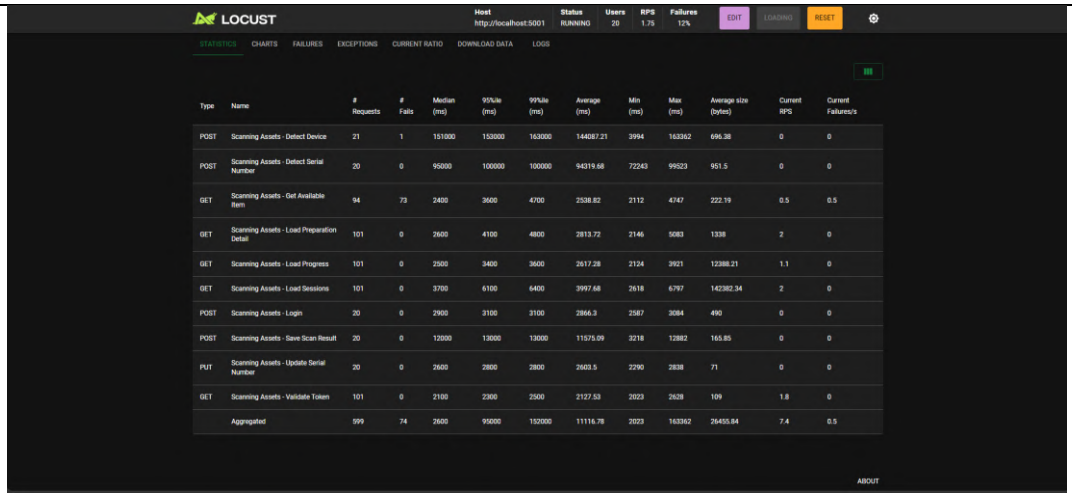
	Time (Durasi)	2 Menit
Response Time (ms)	*Min (ms)*	2024 *ms*
	Average (ms)	2258.93
	Max (ms)	3980 *ms*
	95%ile *(ms)*	2700
Requests	Total *Requests*	963
	Success Requests	963
	Failed Requests	0
Error Rate (%)	0%	
Chart Pengujian		



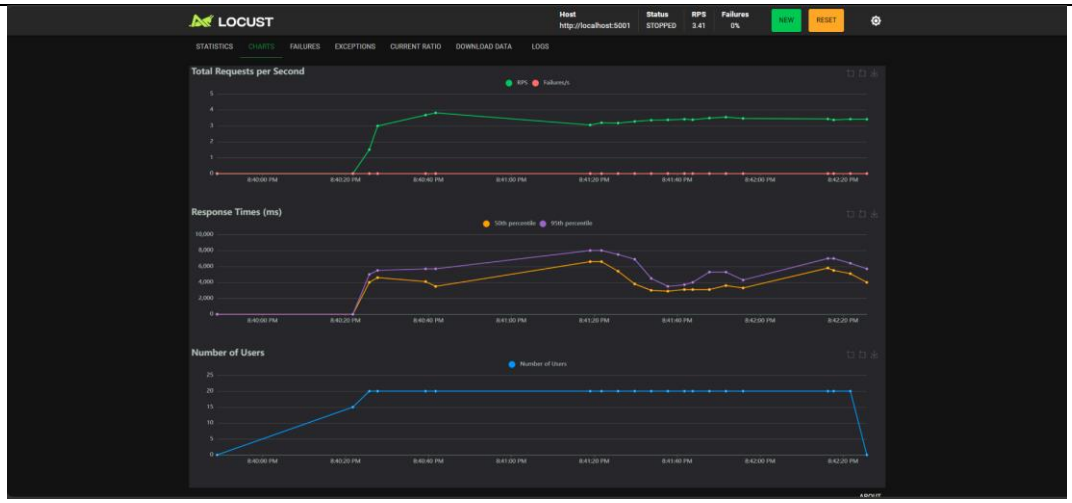
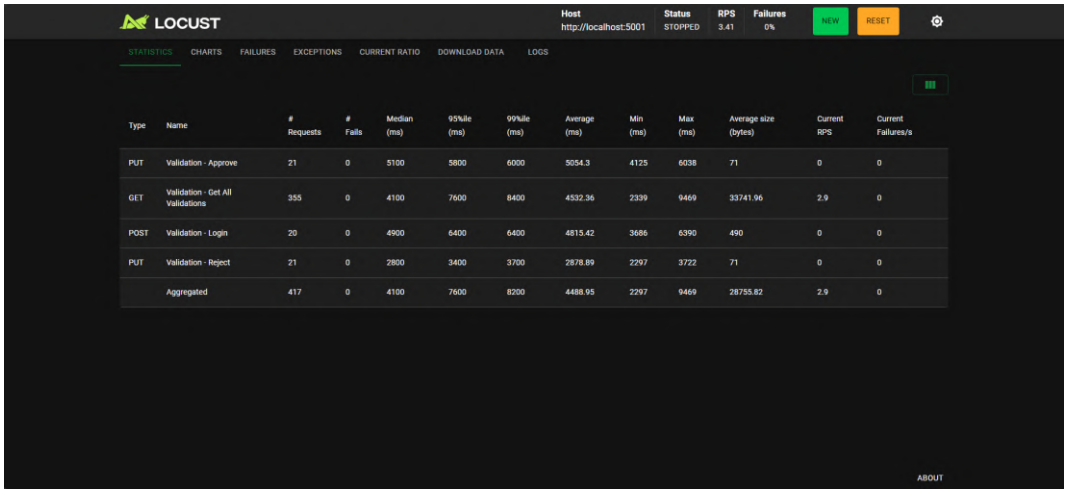
Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
POST	Scanning Preparation - Create Device Session	120	0	2300	2800	2900	2339.07	2098	3276	185.15	0.5	0
GET	Scanning Preparation - Load Departments	140	0	2200	2600	2800	2263.92	2099	3051	10876	2	0
GET	Scanning Preparation - Load Locations	140	0	2200	2600	2800	2246.93	2090	2793	2804	2	0
GET	Scanning Preparation - Load Master Devices	126	0	2200	2500	2800	2244.31	2094	2961	1815	0	0
GET	Scanning Preparation - Load Projects	140	0	2200	2400	2800	2210.38	2088	2902	1502	1.7	0
GET	Scanning Preparation - Load Receivers	137	0	2200	2400	2800	2225.76	2103	2815	46941	0.6	0
POST	Scanning Preparation - Login	20	0	3300	4000	4000	3379.73	2951	3980	490	0	0
GET	Scanning Preparation - Validate Token	140	0	2100	2500	2600	2131.34	2024	2619	109	2	0
	Aggregated	963	0	2200	2700	3400	2258.93	2024	3980	9171.72	8.8	0

Hasil Uji	
ID	LOT04

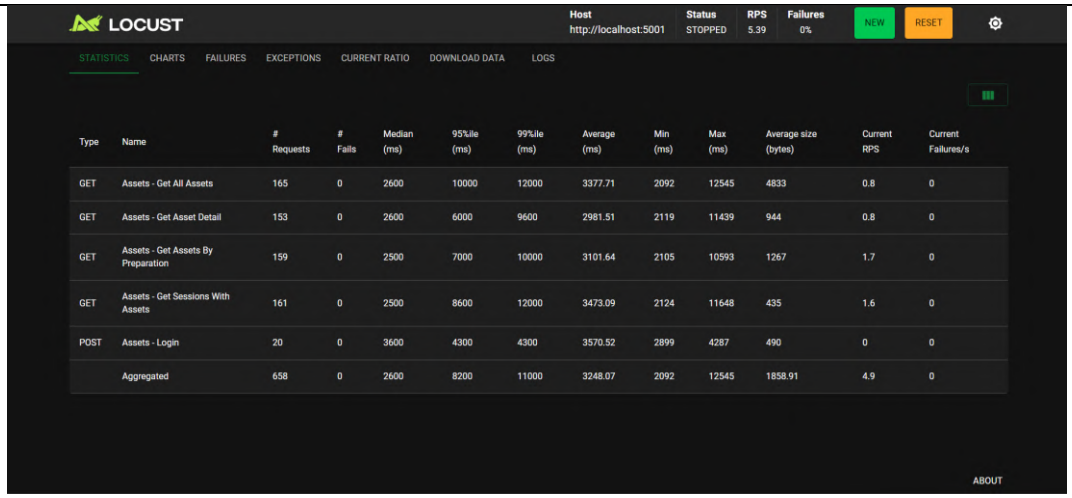
Nama Pengujian	*Load Test – Endpoint Scanning Assets* & Ekstraksi *Serial Number/Scan Code*	
Deskripsi Pengujian	Pengujian dilakukan pada proses *scanning assets* untuk mengukur waktu respons, jumlah *request*, dan kestabilan sistem saat banyak pengguna melakukan *scanning* perangkat atau material secara bersamaan. Pengujian mensimulasikan proses *login*, validasi token, pengambilan *session scanning*, pembukaan detail *preparation*, pengecekan *progress*, deteksi *assets* & Ekstraksi *Serial Number/Scan Code* melalui kamera, serta penyimpanan hasil *scan* ke sistem.	
Method	*GET/POST*	
Skenario Uji	Total *Users*	20
	Ramp Up	5
	Time (Durasi)	2-3 Menit
Response Time (ms)	*Min (ms)*	2023 *ms*
	Average (ms)	11116.78
	Max (ms)	163362 *ms*
	95%ile *(ms)*	95000
Requests	Total *Requests*	599
	Success Requests	525
	Failed Requests	74
Error Rate (%)	12%	
Chart Pengujian		



Hasil Uji		
ID	LOT05	
Nama Pengujian	*Load Test – Endpoint* Validasi Hasil *Scan*	
Deskripsi Pengujian	Pengujian dilakukan pada proses validasi dan verifikasi hasil *scanning assets* untuk mengukur waktu respons, jumlah *request*, serta kestabilan sistem saat banyak pengguna melakukan proses *approve* dan *reject* secara bersamaan. Pengujian mensimulasikan proses *login*, validasi *token*, pengambilan data *validation*, pemilihan item validasi, serta proses *approval* dan *rejection* secara *concurrent* oleh *multiple users*.	
Method	*GET/POST/PUT*	
Skenario Uji	Total *Users*	20
	Ramp Up	5
	Time (Durasi)	2-3 Menit
Response Time (ms)	*Min (ms)*	2297 *ms*
	Average (ms)	4488.95
	Max (ms)	9469 *ms*
	95%ile *(ms)*	7600
Requests	Total *Requests*	417
	Success Requests	417
	Failed Requests	0

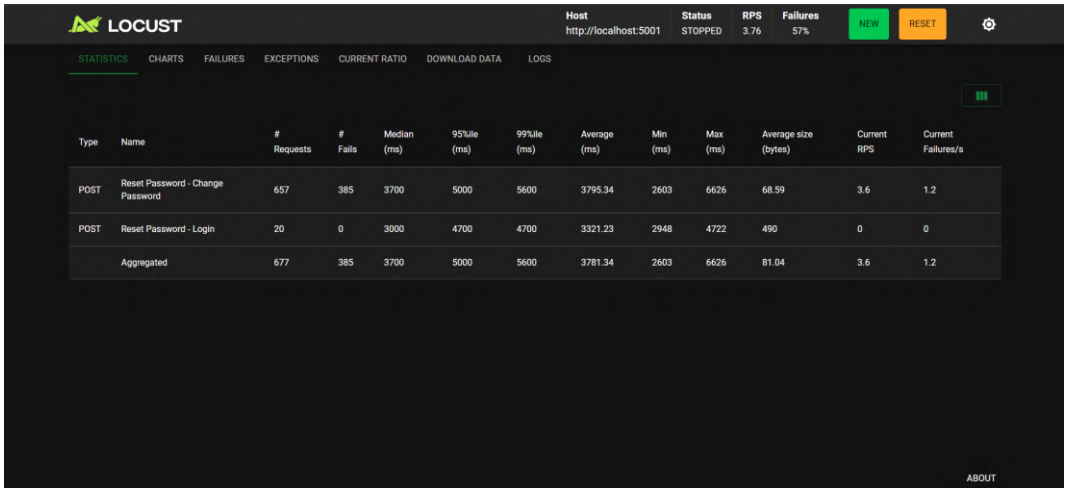
Error Rate (%)	0%
Chart Pengujian	 
Hasil Uji	
ID	LOT06
Nama Pengujian	*Load Test – Endpoint* Melihat Data *Assets*
Deskripsi Pengujian	Pengujian dilakukan pada proses melihat data *assets* untuk mengukur waktu respons, jumlah *request*, dan kestabilan sistem saat banyak pengguna mengakses daftar aset dan detail aset secara bersamaan. Pengujian mensimulasikan proses *login*, pengambilan daftar *session* yang memiliki *assets*, pembukaan detail *assets* berdasarkan *session*, serta pembukaan detail satu asset untuk melihat informasi lengkap aset.

Method	*GET/POST/PUT*	
Skenario Uji	Total *Users*	20
	Ramp Up	5
	Time (Durasi)	2-3 Menit
Response Time (ms)	*Min (ms)*	2092 *ms*
	Average (ms)	3248.07
	Max (ms)	12545 *ms*
	95%ile *(ms)*	8200
Requests	Total *Requests*	658
	Success Requests	658
	Failed Requests	0
Error Rate (%)	0%	
Chart Pengujian		



Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	Assets - Get All Assets	165	0	2600	10000	12000	3377.71	2092	12545	4833	0.8	0
GET	Assets - Get Asset Detail	153	0	2600	6000	9600	2981.51	2119	11439	944	0.8	0
GET	Assets - Get Assets By Preparation	159	0	2500	7000	10000	3101.64	2105	10593	1267	1.7	0
GET	Assets - Get Sessions With Assets	161	0	2500	8600	12000	3473.09	2124	11648	435	1.6	0
POST	Assets - Login	20	0	3600	4300	4300	3570.52	2899	4287	490	0	0
	Aggregated	658	0	2600	8200	11000	3248.07	2092	12545	1858.91	4.9	0

Hasil Uji		
ID	LOT07	
Nama Pengujian	*Load Test – Endpoint Reset/Change Passsword*	
Deskripsi Pengujian	Pengujian dilakukan pada proses perubahan *password* untuk mengukur waktu respons, jumlah *request*, dan kestabilan sistem saat banyak pengguna melakukannya secara bersamaan, dengan simulasi *login*, validasi *token*, serta pengiriman permintaan perubahan *password* melalui *endpoint backend* sebagai representasi *reset password*.	
Method	*GET/POST/PUT*	
Skenario Uji	Total *Users*	20
	Ramp Up	5
	Time (Durasi)	2-3 Menit
Response Time (ms)	*Min (ms)*	2603 *ms*
	Average (ms)	3781.34
	Max (ms)	6626 *ms*
	95%ile *(ms)*	5000
Requests	Total *Requests*	677
	Success Requests	292
	Failed Requests	385

Error Rate (%)	57%
Chart Pengujian	 

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
POST	Reset Password - Change Password	657	385	3700	5000	5600	3795.34	2603	6626	68.59	3.6	1.2
POST	Reset Password - Login	20	0	3000	4700	4700	3321.23	2948	4722	490	0	0
	Aggregated	677	385	3700	5000	5600	3781.34	2603	6626	81.04	3.6	1.2

2. *Security Testing*

Pada tahap ini dilakukan pengujian *non-fungsional* untuk memastikan bahwa sistem memenuhi kebutuhan *non-fungsional* pada aspek keamanan sistem. Pengujian ini difokuskan pada kemampuan sistem dalam melindungi data sensitif melalui mekanisme autentikasi pengguna, enkripsi data, serta pencegahan terhadap akses tidak sah dan berbagai serangan umum seperti *SQL Injection*, *Cross-Site Scripting (XSS)*, dan *brute force attack*. Pengujian ini disusun berdasarkan kebutuhan *non-fungsional* NF-02, yaitu sistem harus memiliki mekanisme keamanan yang memadai untuk menjaga kerahasiaan,

integritas, dan keamanan data pengguna serta memastikan hanya pengguna yang berwenang yang dapat mengakses sistem.

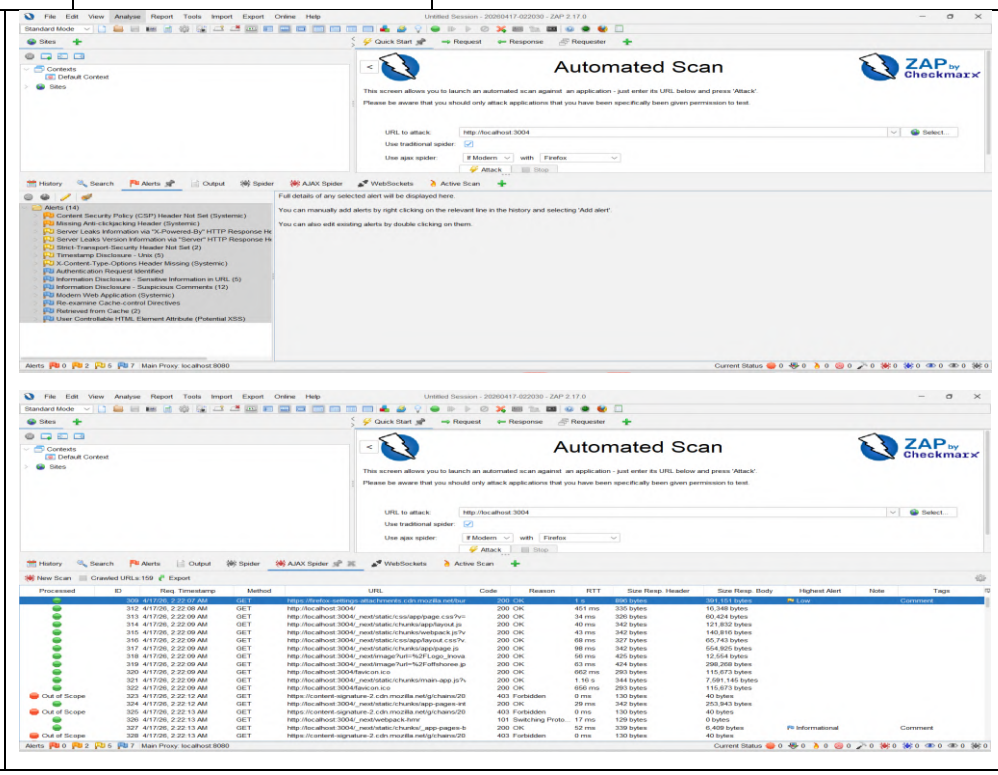
Tabel 4. 17 *Security Testing*

Komponen	**Rincian**
Test Case ID	TC02-NF02
Type	*Security Testing*
Tujuan	Melakukan pengujian untuk emastikan sistem memiliki perlindungan terhadap berbagai ancaman keamanan seperti *SQL Injection, Cross-Site Scripting (XSS),* serta mencegah akses yang tidak sah mungkin masuk ke dalam sistem.
Lingkungan	Pengujian dilakukan pada *server* pengembangan *(localhost / staging)* dengan akses terbatas untuk menghindari kebocoran data.
Alat	*OWASP ZAP*
Metode	1. Melakukan pengujian keamanan dengan pendekatan *penetration testing* pada fitur utama sistem. 2. Mengidentifikasi potensi kerentanan serta melakukan simulasi serangan secara terkontrol untuk menguji ketahanan sistem. 3. Menganalisis tingkat keamanan data baik pada saat proses pengiriman maupun saat data disimpan di dalam sistem.
Ekspektasi	Sistem tidak memiliki celah keamanan yang signifikan dan data sensitif terlindungi dengan mekanisme enkripsi yang memadai, serta akses ke dalam sistem hanya diberikan kepada pengguna yang memiliki otorisasi.
Hasil Uji	
Jumlah *Alert* Keamanan	14

No	**Temuan**	**Tingkat Resiko**	**Penjelasan**
1.	Content Security Policy (CSP)	Menengah	Aplikasi tidak mengirimkan header *Content-Security-Policy* pada *response HTTP* baik dari *Next.js* maupun *Flask.* Hal ini menunjukkan bahwa sistem belum

	Header Not Set.		membatasi sumber konten yang dapat dijalankan oleh *browser*. Kondisi ini dapat meningkatkan risiko serangan Cross Site Scripting (XSS), terutama pada fitur yang melibatkan input pengguna seperti *login*, *scanning assets*, dan validasi data.
2.	*Missing Anti-clickjacking Header*	Menengah	Aplikasi tidak menerapkan header keamanan seperti *X-Frame-Options* atau *frame-ancestors*. Hal ini memungkinkan aplikasi ditampilkan dalam *iframe* oleh pihak lain dan berpotensi dimanfaatkan untuk serangan *clickjacking*.
3.	*User Controllable HTML Element Attribute (Potential XSS)*	Menengah	Ditemukan elemen *HTML* yang dapat dikontrol oleh pengguna tanpa validasi atau sanitasi yang memadai. Kondisi ini berpotensi menyebabkan serangan Cross Site Scripting (XSS).
4.	*Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)*	Rendah	Data dari *Server* menampilkan informasi teknologi yang digunakan melalui *header HTTP* seperti *X-Powered-By*. Informasi ini dapat dimanfaatkan oleh penyerang untuk mengidentifikasi target serangan.
5.	*Server Leaks Version Information*	Rendah	*Server* menampilkan informasi versi pada *header response*. Hal ini dapat membantu

	via "Server" HTTP Response Header Field		penyerang dalam menemukan kerentanan yang spesifik terhadap versi tersebut.
6.	*Strict-Transport-Security Header Not Set*	Rendah	Pada *testing* tidak adanya header HTTP *Strict Transport Security* (HSTS) yang memungkinkan komunikasi dilakukan melalui *HTTP* yang tidak aman. *HSTS* memastikan *browser* hanya menggunakan koneksi *HTTPS* yang aman.
7.	*Timestamp Disclosure - Unix*	Rendah	Sistem aplikasi menampilkan informasi yaitu *timestamp* dalam format *Unix* yang dapat digunakan untuk menganalisis pola sistem atau aktivitas tertentu.
8.	*X-Content-Type-Options Header Missing*	Rendah	*Header X-Content-Type-Options* tidak diset ke *"nosniff"*, sehingga *browser* dapat melakukan *MIME sniffing* yang berpotensi menyebabkan eksekusi konten berbahaya.
9.	*Authentication Request Identified*	*Informational*	Sistem terdeteksi memiliki mekanisme autentikasi seperti penggunaan *token* pada *endpoint API*. Temuan ini hanya bersifat informatif dan bukan kerentanan.
10.	*Information Disclosure - Sensitive Information in URL*	*Informational*	Ditemukan parameter dalam *URL* yang berpotensi mengandung informasi sensitif. Hal ini tidak disarankan karena *URL* dapat tersimpan di *browser history* atau *log server*.
11.	*Information Disclosure -*	*Informational*	*Response* aplikasi mengandung komentar yang berpotensi memberikan informasi

	Suspicious Comments		tambahan terkait struktur atau konfigurasi sistem.
12.	*Modern Web Application*	*Informational*	Aplikasi teridentifikasi sebagai *modern web application* berbasis *Next.js*. Untuk pengujian lebih lanjut disarankan menggunakan *AJAX Spider* karena otomatis jauh lebih efektif dibanding *standard spider*.
13.	*Re-examine Cache-control Directives (Cache-Control Header Not Properly Set)*	*Informational*	*Header cache-control* masih belum dikonfigurasi dengan baik sehingga memungkinkan *browser* ataupun *proxy* menyimpan data yang seharusnya tidak di-*cache*.
14.	*Retrieved from Cache*	*Informational*	Beberapa *response* diambil dari *cache* yang dapat berpotensi menyebabkan kebocoran data apabila konfigurasi *caching* tidak tepat.
Hasil Uji *Security Testing*			

BAB V
KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, pengembangan, dan pengujian sistem Otomatisasi Validasi *Serial Number* dan *Scan Code* pada Aset *IT* Menggunakan *YOLO* dan *OCR* guna Efisiensi Pendataan dan Keamanan Aset *IT* di PT XYZ Indonesia, maka dapat disimpulkan bahwa tujuan proyek akhir ini telah tercapai. Kesimpulan utama yang diperoleh adalah sebagai berikut:

1. Sistem deteksi objek aset *IT* menggunakan metode *YOLO* berhasil dirancang dan diimplementasikan, sehingga mampu mengenali perangkat dan material secara otomatis melalui kamera dengan tingkat akurasi yang baik dalam berbagai kondisi.
2. Sistem ekstraksi teks menggunakan *OCR* berhasil dikembangkan untuk membaca *serial number* dan *scan code*, sehingga proses identifikasi aset dapat dilakukan secara otomatis dan mengurangi kesalahan input manual.
3. Sistem *scanning* berbasis *session* melalui fitur *scanning preparation* berhasil dibangun, sehingga proses pendataan aset menjadi lebih terstruktur, mulai dari perencanaan, proses *scanning*, hingga validasi data.
4. Sistem validasi dan verifikasi aset berhasil diimplementasikan, di mana hasil *scanning* dapat dilakukan proses *approve* sehingga menjadi data aset yang final, dan proses *reject* sebelum disimpan sebagai data aset, sehingga meningkatkan keakuratan dan keamanan data inventaris.
5. Sistem *monitoring*, laporan, dan manajemen aset berbasis *web* berhasil dikembangkan, sehingga pengguna dapat memantau aktivitas inventaris secara *real-time*, mengelola data aset, serta menghasilkan laporan secara otomatis dan terintegrasi.

Dengan terealisasinya seluruh tujuan tersebut, sistem mampu meningkatkan efisiensi proses pendataan, validasi, dan pengelolaan aset *IT* secara lebih cepat, akurat, terstruktur, serta mendukung keamanan data inventaris di PT XYZ Indonesia.

5.2 Saran

Berdasarkan hasil pengembangan dan pengujian sistem yang telah dilakukan, terdapat beberapa saran yang dapat dipertimbangkan untuk pengembangan sistem di masa mendatang:

1. Sistem dapat dikembangkan dengan penambahan fitur notifikasi otomatis untuk mengingatkan pengguna terkait jadwal pengecekan inventaris atau *scanning* yang belum dilakukan, sehingga dapat meminimalkan keterlambatan dalam proses pendataan aset.
2. Sistem dapat ditingkatkan dengan kemampuan integrasi dengan teknologi identifikasi lain seperti *RFID* atau *QR Code* berbasis industri, sehingga proses identifikasi aset menjadi lebih fleksibel dan tidak hanya bergantung pada *serial number* atau *scan code*.
3. Perlu dilakukan pengembangan model *AI* dengan penambahan *dataset* yang lebih beragam dan kondisi lingkungan yang lebih kompleks, sehingga dapat meningkatkan akurasi deteksi *YOLO* dan *OCR* terutama untuk objek-objek yang kecil pada kondisi pencahayaan rendah.
4. Sistem dapat dilengkapi dengan fitur *mobile application (Android/iOS)* secara *native*, sehingga proses *scanning* dan *monitoring* inventaris dapat dilakukan secara lebih optimal di lapangan tanpa ketergantungan penuh pada *browser*.
5. Perlu ditambahkan fitur audit *trail* dan *log* keamanan yang lebih mendalam, seperti pelacakan histori perubahan data aset secara detail, sehingga meningkatkan transparansi dan keamanan dalam pengelolaan inventaris *IT*.

Dengan adanya pengembangan lebih lanjut berdasarkan saran tersebut, diharapkan sistem dapat menjadi solusi inventaris aset *IT* yang semakin optimal, adaptif, dan mampu mendukung kebutuhan operasional perusahaan secara berkelanjutan.

DAFTAR PUSTAKA

Aldy Agustian, S. a. (2025, Juni 24). Identifikasi Gender Dalam Keaktifan Pembelajaran Menggunakan Metode Face Recognition. *Jurnal Minfo Polgan, 14*(1), 953-964. doi:https://doi.org/10.33395/jmp.v14i1.14952

Alfisyahri, Q. (2024). *RANCANG BANGUN SISTEM MONITORING AREA PARKIR DAN PELACAKAN POSISI KENDARAAN BERBASIS MIKROKONTROLLER DAN COMPUTER VISION.* Universitas Putra Indonesia YPTK. Padang: Universitas Putra Indonesia YPTK.

Anggia Dasa Putri, A. M. (2023, Juli). Penerapan Metode Mamdani Fuzzy Logic untuk Menentukan Pembelian Alat Berat dalam Proyek Migas di PT SMOE Indonesia. *JURNAL DESAIN DAN ANALISIS TEKNOLOGI(JDDAT), 2*(2), 138-149.

Annisa Hafsah, A. A. (2025, Juli 5). KEAMANAN DATA DALAM SISTEM DATABASE. *Jurnal Ilmiah Nusantara (JINU), 2*(4), 183-197. doi: https://doi.org/10.61722/jinu.v2i4.4997

Bassil, Y. (2012). A Simulation Model for the Waterfall Software Development Life Cycle. *International Journal of Engineering & Technology (iJET), 2*(5), 1-7. Retrieved from https://iet-journals.org/archive/2012/may_vol_2_no_5/255895133318216.pdf

Dimas Prasetyo, W. .. (2017). *Sistem informasi inventaris desa berbasis web.* Universitas Muhammadiyah Surakarta. Surakarta: Universitas Muhammadiyah Surakarta.

Hidayati, N. (2019, Januari). Penggunaan Metode Waterfall Dalam Rancang Bangun Sistem Informasi Penjualan. *Generation Journal, 3*(1), 1-10.

Kuncoro Banu, D. A. (2023, Desember). OCR: Masa Depan Pengenalan Karakter Optik dan Dampaknya pada Kehidupan. *Jurnal Teknologi Informasi, 9*(2), 147-156. Retrieved from https://ejournal.urindo.ac.id/index.php/TI/index

Munthe, I. R. (2017, September). Penerapan Model Waterfall Pada Perancangan Sistem Informasi Pendaftaran Siswa Baru Smk Swasta Teladan Rantauprapat Berbasis Web. *Informatika : Jurnal Ilmiah AMIK Labuhan Batu, 5*(3), 15-21.

Nugraha, K. C. (2023, September 29). DETEKSI PLAT NOMOR KENDARAAN MENGGUNAKAN ALGORITMA YOLO. *Jurnal Algoritma, Logika dan Komputasi* , *6*(2), 605-611. doi: http://dx.doi.org/10.30813/j-alu.v2i2.4739

Nursikuwagus, A. &. (2016). Perangkat Lunak Sistem Manajemen Aset Dalam Penanganan Aset It. *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*.

Ramadhan, G., Putri, S. A., & Alwidan, Z. (2023). *Aplikasi pengelolaan smart parking.* Politeknik Manufaktur Negeri Bangka Belitung. Sungailiat: Politeknik Manufaktur Negeri Bangka Belitung. Retrieved from repository.polman-babel.ac.id

Richo, R. (2025, Februari 2). SISTEM IDENTIFIKASI INFORMASI EXPIRED DATE PRODUK KEMASAN MENGGUNAKAN KOLABORASI METODE YOLO-V11M DAN PADDLEOCR. *Jurnal Rekayasa Sistem Informasi dan Teknologi*, *2*(3), 886-900.

Roy Hanafi, A. H. (2024, Maret 7). Comparison of Web Page Rendering Methods Based on Next.js Framework Using Page Loading Time Test. *TEKNIKA*, *13*(1), 102-108. doi:10.34148/teknika.v13i1.769

Septina Annuria Rakhma, R. T. (2022, Desember). Sistem *Monitoring* Data Aset dan Inventaris IT Berbasis Web pada PT. Pan Brothers Tbk. *JURNAL TOPIK GLOBAL (Jurnal Teknologi, Pendidikan dan Manajemen Global)*, *1*(2), 60-65.

Utami, D. A. (2025, September). Implementasi Pengelolaan Aset dan Inventarisasi Barang di Lingkungan Sekretariat Daerah Kota Padang Panjang: Studi. *Jurnal Akademik Pengabdian Masyarakat*, *3*(5), 270-276. doi:https://doi.org/10.61722/japm.v3i5.6553

Zein, A. (2021, Desember). Kecerdasan Buatan Dalam Hal Otomatisasi Layanan. *JurnalIlmu Komputer JIK*, *4*(2), 16-25.

Ofoeda, J., Boateng, R., & Effah, J. (2019). Application programming interface (API) research: A review of the past to inform the future. *International Journal of Enterprise Information Systems (IJEIS)*, *15*(3), 76-95.

Alexandrova, S., Tatlock, Z., & Cakmak, M. (2015, May). RoboFlow: A flow-based visual programming language for mobile manipulation tasks. In *2015 IEEE international conference on robotics and automation (ICRA)* (pp. 5537-5544). IEEE.

Kanani, P., & Padole, M. (2019). Deep learning to detect skin cancer using google colab. *International Journal of Engineering and Advanced Technology Regular Issue*, *8*(6), 2176-2183.

Jin, J., Fang, Z., Chen, L., & Gao, Y. (2025). PostMan: A Productive System for Spatio-temporal Data Management and Analysis: J. Jin et al. *Data Science and Engineering*, 1-24.

Cosentino, V., Luis, J., & Cabot, J. (2016, May). Findings from GitHub: methods, datasets and limitations. In *Proceedings of the 13th international conference on mining software repositories* (pp. 137-141).

DAFTAR LAMPIRAN

Lampiran A. Dokumen Proses Pengumpulan *Requirement*

Dokumen *Work Process*

https://drive.google.com/drive/folders/1iCORJbcE3p7PN1hZH0DnRpa8eGvTthRM?usp=drive_link

Dokumentasi Postman

https://drive.google.com/drive/folders/1AmdHSlUbcKyUe01z5UXNN1x3KRG1EYIc?usp=drive_link

Lampiran B. Dokumentasi Perancangan Sistem

https://drive.google.com/drive/folders/11DPUPhyxQLf_l4BZ3ImRlYmgEADL44ox?usp=sharing

Lampiran C. Dokumentasi *Labeling* Data Perangkat, Material, *Serial Number, dan Scan Code* pada *Roboflow*

https://app.roboflow.com/clinton-alfaro-cf96y/projects?queryText=&pageSize=50&startingIndex=0&browseQuery=true&tab=projects

Detector *Devices*

https://universe.roboflow.com/clinton-alfarocf96y/detector-devices

Detector Material

https://universe.roboflow.com/clinton-alfarocf96y/detector-material

Detector Serial Number

https://universe.roboflow.com/clinton-alfarocf96y/detector-serial-number

Detector Scan Code

https://universe.roboflow.com/clinton-alfarocf96y/detector-scan-code

Lampiran D. Dokumentasi *Training* Data Perangkat, Material, *Serial Number*, dan *Scan Code* pada *Google Colab*

https://drive.google.com/drive/folders/1UD5Bpz_s3aZRFDjgLOiSNC6f00qRDp8N?usp=drive_link

Lampiran E. Link Produk & *Script* Ekstraksi *Frame Dataset Inova*

https://github.com/CLINTON1233/inova-asset-detection.git

https://github.com/CLINTON1233/Dataset-Inventory-Asset-Management.git

Lampiran F. Dokumen Pengujian UAT

Performance Testing

https://drive.google.com/drive/folders/1LJX9eHEuHtQrY2xPOHFp4kb5uA1GW0-6?usp=drive_link

Security Testing

https://drive.google.com/drive/folders/1Jk_RcseUNktSaBRzWLeQDPJ5LlFilaII?usp=drive_link

Dokumen User Acceptance Testing (UAT)

https://drive.google.com/drive/folders/13thRYWyyS3SGyShTdvn1ouabw7ol4sXV?usp=drive_link

Lampiran G. Video Demo Produk

https://drive.google.com/drive/folders/1kHFDOd3uajx6sN-0VkmWVsFIKSx6rJHD?usp=drive_link