



## SEMB-DEBUG APP: DIGITALISASI PROSES DEBUG PADA TAHAP ANALISIS DAN PENCATATAN DI PT. SCHNEIDER MANUFACTURING BATAM (SEMB)

Regita Ayu Cahyani<sup>1</sup>, Muchamad Fajri Amirul Nasrullah<sup>2</sup>

<sup>1,2</sup>Program Studi Teknik Informatika, Politeknik Negeri Batam

<sup>1</sup>regita.ayu.cahyani3@students.polibatam.ac.id, <sup>2</sup>fajri@polibatam.ac.id

### Abstract

*PT. Schneider Electric Manufacturing Batam (SEMB), an electronics company, currently uses Excel as a medium for collecting product data and recording analysis results by operators during the debug process. On the other hand, engineers are responsible for tracking incoming and outgoing products during the debugging process, including information such as quantity, product type, and other categories. This has led companies to adopt a digital approach to data collection and tracking of product data during the debugging process. One solution that can be considered is the design of a technology-based application that combines data collection and product tracking functions. Such applications can be designed to allow operators to enter product data directly, including relevant analysis results. Meanwhile, engineers can use the app to track incoming and outgoing products during debugging by entering information such as product type, and other categories. By adopting this digital solution, companies can access available data in real time and accurately regarding the data collection and product tracking process during the debug process.*

*Keywords: data, digitalization, debug process, PT. Schneider Electric Manufacturing Batam (SEMB)*

### Abstrak

PT. Schneider Electric Manufacturing Batam (SEMB), sebuah perusahaan elektronik, yang saat ini menggunakan Excel sebagai media pendataan produk dan pencatatan hasil analisis oleh operator selama proses *debug*. Di sisi lain, *engineer* bertanggung jawab dalam melacak produk masuk maupun keluar selama proses *debug*, termasuk informasi seperti jumlah, jenis produk, dan kategori lainnya. Hal ini, membuat perusahaan mengadopsi pendekatan *digital* dalam pendataan serta pelacakan data produk selama proses *debug*. Salah satu solusi yang dapat dipertimbangkan adalah rancang bangun aplikasi berbasis teknologi yang menggabungkan fungsi pendataan dan pelacakan produk. Aplikasi tersebut dapat dirancang untuk memungkinkan operator memasukkan data produk secara langsung, termasuk hasil analisis yang relevan. Sementara itu, *engineer* dapat menggunakan aplikasi tersebut untuk melacak produk masuk dan keluar selama proses *debug* dengan memasukkan informasi seperti jenis produk, dan kategori lainnya. Dengan adopsi solusi *digital* ini, perusahaan dapat mengakses data yang tersedia secara *real time* dan akurat terkait proses pendataan dan pelacakan produk selama proses *debug*.

*Kata kunci: data, digitalisasi, proses debug, PT.Schneider Electric Manufacturing Batam (SEMB)*

### 1. Pendahuluan

PT. Schneider Electric Manufacturing Batam (SEMB) sebuah perusahaan yang bergerak dibidang *electric*, yang berdiri pada tahun 1836 oleh grup *Schneider Electric*. Dalam era *digital* yang terus berkembang ini, PT. Schneider Electric Manufacturing Batam (SEMB), menghadapi tantangan dalam pendataan produk dan pelacakan hasil analisis selama proses *debug*. Saat ini, perusahaan memakai *Excel* sebagai media pendataan produk dan pencatatan hasil analisis yang dilakukan oleh operator. Namun, cara ini kurang memenuhi kebutuhan yang semakin kompleks mengikuti

bertambahnya jumlah produk berdasarkan kategori tertentu.

Di sisi lain, *engineer* memiliki tanggung jawab dalam melakukan pelacakan data produk, serta mengunduh informasi penting seperti jumlah, jenis produk, dan kategori lainnya. Namun, tanpa dukungan teknologi yang memadai, proses ini dapat menjadi rumit serta tidak dapat diakses secara *real time* dan *online*.

Perusahaan menyadari pentingnya mengadopsi pendekatan *digital* yang terstruktur dalam pendataan produk serta pelacakan produk selama proses *debug*. Memanfaatkan teknologi yang tepat, perusahaan dapat

mengakses data secara *real time* dan *online* selama pengguna memiliki koneksi pada *server* perusahaan dengan memanfaatkan *VPN* yang disediakan.

Rancang bangun aplikasi berbasis teknologi yang menggabungkan fungsi pendataan produk dan pelacakan produk secara terintegrasi. Aplikasi ini menyediakan media bagi operator untuk memasukkan data produk secara langsung, termasuk hasil analisis yang relevan yang sesuai dengan alur kerja. Sementara itu, *engineer* dapat menggunakan aplikasi ini untuk melacak produk, serta mengunduh informasi penting seperti jenis produk dan kategori lainnya.

Dengan solusi digital ini, PT. Schneider Electric Manufacturing Batam (SEMB) dapat mengakses data secara *real time*, memberikan rincian informasi terbaru serta manajemen pendataan produk yang lebih baik dan terstruktur.

Proses digitalisasi, khususnya pembuatan *website*, melibatkan pemilihan teknologi yang sesuai. Dalam hal ini, PT. Schneider Electric Manufacturing Batam (SEMB) menggunakan *MSSQL* sebagai database, *C#* sebagai bahasa pemrograman yang dipakai dan diimplementasikan pada aplikasi ini serta *ASP.NET* sebagai kerangka kerja aplikasi ini dengan menggunakan *MVC (Model-View-Controller)* sebagai arsitekturnya.

Penggunaan media *digital* diterapkan dalam pembuatan *website* bernama SEMB-DEBUG. Proyek ini bertujuan untuk mentransformasi penggunaan media konvensional seperti *excel* dengan menerapkan teknologi *digital* pada proses *debug* tahap analisis dan pencatatan, sehingga perusahaan mendapatkan sistem manajemen data yang baik dan dapat diakses secara *real time* dan *online*.

## 2. Landasan Teori

### 2.1. Tinjauan Pustaka

Sebagai perbandingan penelitian yang berkaitan dengan pengolahan dan pencatatan data pada aplikasi SEMB-DEBUG adalah sebagai berikut dapat dilihat pada Tabel berikut:

Tabel 1. Perbandingan

Judul Penelitian	SEMB-DEBUG App: Digitalisasi Proses Debug Pada Tahap Analisis dan Pencatatan (Regita Ayu Cahyani, 2023)	Aplikasi Inventory Manajemen Aset Berbasis Web (Maryamah Safitri, Faridi, Delvi Nirmala, 2019)	Aplikasi Pengelolaan Data Kepegawaian Berbasis Web pada PT. Pelayaran Sakti Inti Makmur Palembang (Novan wijaya, Adelia Rizky Febriyanti, dan Anton Wibowo, 2020)
Bahasa Pemrograman	C# dan Javascript	PHP	PHP
Basis Data	MSSQL	MYSQL	MYSQL
Platform	Web	Web	Web
Metode	Waterfall	Waterfall	Waterfall

Penelitian yang dilakukan oleh Maryamah, Faridi dan Safitri yang berjudul “Aplikasi *Inventory* Manajemen Aset Berbasis *Web*” aplikasi ini dikembangkan sebagai solusi untuk mengatasi kompleksitas dan kesulitan

dalam pengelolaan aset dan pencatatan data secara manual. Metode pengembangan yang digunakan adalah metode *waterfall*. Aplikasi ini dibangun menggunakan *platform web* dengan menggunakan bahasa pemrograman *PHP* dan *JavaScript*, serta *database MySQL* [1].

Penelitian lain yang berjudul “Aplikasi Pengelolaan Data Kepegawaian Berbasis Web pada PT. Pelayaran Sakti Inti Makmur Palembang” oleh Novan wijaya, Adelia Rizky Febriyanti, dan Anton Wibowo adalah sebuah solusi untuk mengatasi pengolahan data kepegawaian secara manual, diantaranya rekapitulasi data pegawai, absensi, surat peringatan, cuti pegawai dan lain-lain. Aplikasi ini bertujuan untuk mempermudah melakukan pencatatan dan pengolahan data yang dilakukan oleh HRD. Dibangun dengan bahasa pemrograman *PHP* dan *MySQL* dengan metode penelitian *iterative* yang mengkombinasikan proses pada model air terjun dan *iterative* pada model *prototype* dan *platform Web* [2].

Sementara itu, untuk penelitian yang berjudul “SEMB-DEBUG App: Digitalisasi Proses *Debug* Pada Tahap Analisis dan Pencatatan” oleh penulis pada tahun 2024, aplikasi ini bertujuan untuk mendigitalisasi atau mentransformasi proses *debug* pada tahap analisis dan pencatatan di PT. Schneider Manufacturing Batam (SEMB) yang semula menggunakan *excel* sebagai media pendataan. Dibangun menggunakan *platform ASP.NET* dengan menggunakan bahasa pemrograman *C#* dan *JavaScript*, serta *database Microsoft SQL Server*. Metode pengembangan sistem yang digunakan adalah metode *waterfall*.

#### A. Microsoft SQL Server (MSSQL)

*Microsoft SQL Server*, sebagai sistem manajemen basis data relasional (*RDBMS*) yang dikembangkan oleh *Microsoft*, memiliki peran krusial dalam transformasi *digital* PT. Schneider Electric Manufacturing Batam (SEMB). Sebagai bagian dari landasan teknis dalam implementasi aplikasi SEMB-DEBUG, *SQL Server* didesain untuk menyimpan, mengatur, dan mengakses data produk. Dengan kemampuannya yang telah terbukti dalam mengelola berbagai jenis data bisnis, seperti data pelanggan, keuangan, inventaris, dan produksi, *SQL Server* menjadi pilihan yang tepat untuk mendukung perusahaan dalam memperbarui pendekatan mereka terhadap pendataan produk dan pelacakan selama proses *debug*. Dengan demikian, penggunaan *SQL Server* sebagai bagian utama dari aplikasi SEMB-DEBUG memberikan landasan yang kuat bagi PT. Schneider Electric Manufacturing Batam (SEMB) untuk mencapai tujuan digitalisasi proses *debug* ini [3].

#### B. Bootstrap

Dalam konteks digitalisasi PT. Schneider Electric Manufacturing Batam (SEMB), *Bootstrap* menjadi kerangka kerja krusial dalam merancang tampilan

website aplikasi SEMB-DEBUG. *Bootstrap* mempercepat pembuatan tampilan *website* yang responsif, menarik, dan *user-friendly* bagi pengembang web dan desainer. Dengan penggunaan *Bootstrap* yang efektif mendukung visi perusahaan dalam meningkatkan pengalaman pengguna melalui aplikasi SEMB-DEBUG [4].

#### C. C# (C-Sharp)

C# atau C-sharp adalah bahasa pemrograman yang mendukung dalam pengembangan aplikasi SEMB-DEBUG. C# menjadi bahasa pemrograman yang menjanjikan dalam dunia industri sehingga mendukung aplikasi yang sangat kompleks. C# merupakan versi canggih dari C++ karena ada anggapan bahwa tanda # adalah perpaduan dari 4 buah tanda tambah yang disusun sedemikian rupa. C# juga mendukung programing dengan .NET melalui aplikasi visual studio yang sangat populer [5].

#### D. Javascript

*JavaScript* memegang peran krusial dalam pengembangan aplikasi SEMB-DEBUG. Sebagai bahasa pemrograman, *JavaScript* meningkatkan interaktivitas dan responsivitas aplikasi, memfasilitasi pendataan dan pelacakan produk secara real time. Dengan sifatnya yang tingkat tinggi, bertaraf klien, dan berorientasi objek, *JavaScript* memastikan aplikasi berfungsi optimal. Melalui penggunaan *JavaScript*, perusahaan dapat memanfaatkan editor teks, menciptakan solusi digital yang efektif dan terstruktur sesuai dengan visi perusahaan [6].

#### E. ASP.NET

Dalam proses transformasi *digital* di PT. Schneider Electric Manufacturing Batam (SEMB), *ASP.NET* dan *ASP.NET Core* berperan sebagai pemangkin utama dalam pengembangan aplikasi SEMB-DEBUG yang inovatif. Sebagai fondasi pengembangan, *ASP.NET* memberikan kerangka kerja yang kokoh, memungkinkan pembangunan aplikasi *web* seperti SEMB-DEBUG yang membutuhkan manajemen data yang canggih dan kemampuan akses secara *real time*. Sementara itu, kehadiran *ASP.NET Core* sebagai versi *open source* yang lebih adaptif memberikan fleksibilitas tak terbatas, memungkinkan aplikasi ini berjalan di berbagai *platform*, dari *macOS* hingga *Linux* dan *Windows*, sehingga merangkul beragam kebutuhan perusahaan yang berkembang. Dengan peran krusialnya, *ASP.NET* dan *ASP.NET Core* menjadi tonggak penting dalam menjawab tantangan *digital*, membawa SEMB-DEBUG menuju arah yang lebih inovatif dan berkelanjutan [7].

#### F. Model-View-Controller (MVC)

Untuk mendukung aplikasi SEM-DEBUG, maka dibutuhkan penggunaan metode yang tepat, salah satunya adalah metode *model view controller*. MVC adalah sebuah konsep yang diperkenalkan oleh *Trygve*

*Reenkaug*, penemu *Smalltalk*, untuk menggambarkan data dan pemrosesan (*model*), jembatan antara *model* dan *view* (*controller*) dan menggambarkan tampilan (*view*) agar dapat direpresentasikan pada antarmuka [8].

### 2.2. Metode Penelitian

#### A. Metode Pengumpulan Data

##### a. Data Primer

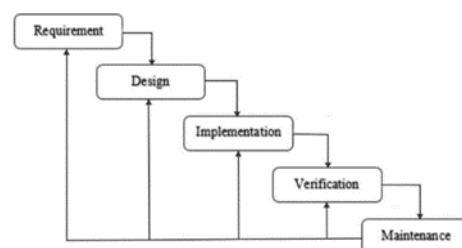
Proses pengumpulan data primer dilakukan melalui pertemuan atau wawancara dengan pengguna, baik operator maupun *engineer*, untuk memahami proses *debug* secara menyeluruh, tahapan yang dilaluinya, serta kebutuhan data yang diperlukan untuk integrasi dan pelacakan produk, termasuk penyesuaian alur kerja dengan sistem dan manajemen data yang sesuai dengan kebutuhan pengguna.

##### b. Data Sekunder

Data sekunder diperoleh melalui studi literatur yang berkaitan dengan pengolahan data produk dan pengembangan aplikasi SEMB-DEBUG, yang melibatkan penelusuran jurnal, buku, serta penelitian serupa.

#### B. Metode Pengembangan Sistem

Metode pengembangan sistem ini mengadopsi pendekatan *system development life cycle* (SDLC) yaitu model air terjun (*waterfall*). Dicitrakan oleh sifatnya yang sistematis dan berurutan, model *Waterfall* adalah metodologi pengembangan perangkat lunak klasik. Model ini tepat disebut “air terjun” karena penyelesaian setiap tahap secara berurutan sebelum melanjutkan ke tahap berikutnya [9].



Gambar 1. Metode Pengembangan *Waterfall*

Sumber : (Wahid, 2020) [9]

##### 1) Kebutuhan

Pada tahap analisis kebutuhan, telah dilakukan analisis dan persiapan untuk menentukan kebutuhan aplikasi. Ini mencakup analisis terhadap fitur-fitur yang diperlukan, kebutuhan pengguna, serta batasan-batasan aplikasi yang akan dikembangkan. Proses ini melibatkan identifikasi kebutuhan karyawan terhadap fitur-fitur yang diinginkan dalam aplikasi.

##### 2) Desain

Pada tahap desain sistem, telah dilakukan perancangan antarmuka serta desain basis data aplikasi SEMB-

DEBUG dengan memperhatikan kebutuhan pengguna. Tahap ini mencakup pembuatan struktur basis data yang terdiri dari nama basis data dan tabel yang sesuai menggunakan *Microsoft SQL Server (MSSQL)*. Selain itu, perancangan tampilan antarmuka telah dilakukan menggunakan aplikasi seperti Figma untuk memastikan antarmuka yang intuitif dan *user-friendly*.

### 3) Implementasi

Tahap implementasi telah mencakup pengembangan situs web. Implementasi dilakukan dengan memproses kode front end dan back end sesuai dengan kebutuhan pengguna dan desain sistem. Pengembangan ini menggunakan bahasa pemrograman *C#*, *CSS*, *JavaScript*, dan *Microsoft SQL Server (MSSQL)*. Langkah awal mencakup pembuatan front end berdasarkan desain antarmuka yang telah disiapkan sebelumnya, diikuti dengan pengembangan back end menggunakan bahasa pemrograman *C#* dan *JavaScript*, dengan penerapan arsitektur *MVC (Model-View-Controller)* untuk memastikan pemisahan yang jelas antara logika aplikasi, antarmuka pengguna, dan kontrol alur data.

### 4) Pengujian

Pada tahap ini telah dilakukan pengujian terhadap aplikasi yang telah dikembangkan untuk mendeteksi dan memperbaiki bug atau kesalahan dalam implementasi serta memastikan aksesibilitas dan kemudahan navigasi bagi pengguna. Pengujian ini mencakup pemeriksaan seluruh fitur aplikasi untuk memastikan tidak ada bug ketika digunakan oleh pengguna. Metode pengujian yang digunakan adalah *Blackbox*, yang berfokus pada verifikasi bahwa semua fungsi aplikasi bekerja sesuai dengan spesifikasi yang telah ditetapkan tanpa memeriksa kode internal [10].

### 5) Pemeliharaan

Pada tahap pemeliharaan, telah dilakukan pemeliharaan secara berkala untuk memastikan aplikasi beroperasi dengan optimal. Selain itu, pemeliharaan ini juga mencakup pemeriksaan fungsi semua halaman dan tautan untuk memastikan semuanya berfungsi dengan baik dan tidak ada masalah yang mengganggu pengguna.

## 3. Hasil dan Pembahasan

Aplikasi SEMB-DEBUG ini menyediakan sarana pengelolaan data produk dan pelacakan hasil analisis produk. Aplikasi ini membantu pengguna dalam mencari dan mengumpulkan data seperti jumlah, jenis, dan kategori produk. Dengan adanya solusi digital ini, perusahaan bisa mengatasi masalah dalam melacak produk selama proses *debug* dan pengelolaan secara *online*. Selain itu, data yang tersedia secara *real time* akan membantu dalam pengambilan keputusan yang

lebih cepat dan tepat terkait dengan manajemen produk selama proses *debug*.

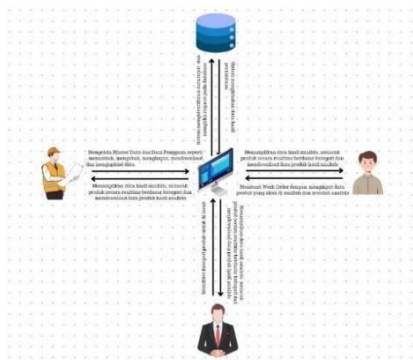
Berfokus pada penerapan teknologi informasi dalam pengembangan solusi *digital*. Wilayah teknologi yang diterapkan mencakup penggunaan basis data *MSSQL*, bahasa pemrograman *C#*, dan kerangka kerja *ASP.NET* dengan arsitektur *MVC*. Pengetahuan terus diperbarui dengan mengikuti perkembangan terbaru dalam bidang teknologi informasi, termasuk pembaruan dalam bahasa pemrograman, kerangka kerja, dan teknologi basis data.

Rancang bangun aplikasi SEMB-DEBUG sesuai dengan tujuan awal PT. Schneider Electric Manufacturing Batam (SEMB) untuk mengatasi tantangan dalam pendataan produk dan pelacakan hasil analisis selama proses *debug*. Dengan pendekatan *digital* terstruktur, aplikasi ini mengintegrasikan fungsi pendataan dan pelacakan produk secara *real time*, meningkatkan manajemen data, dan mengatasi masalah aksesibilitas data. Menggunakan teknologi seperti *MSSQL*, *C#*, dan *ASP.NET* serta *MVC*, rancang bangun aplikasi ini berhasil mentransformasi proses *debug* menjadi lebih terstruktur dan *online*.

### 3.1. Gambaran Umum

Sistem ini dirancang untuk memberikan pengguna (Admin, *User*, dan *Viewer*) akses yang terstruktur dan aman ke dalam *platform*. Dengan menggunakan fitur *login*, setiap pengguna dapat masuk ke dalam sistem menggunakan kredensial mereka masing-masing. Selanjutnya, mereka dapat dengan mudah mengelola data produk, baik itu dengan membuat *work order* baru, mengkonfirmasi *work order*, atau mengisi analisis produk sesuai dengan tugas dan wewenang yang dimiliki. Keandalan sistem juga terbukti dalam kemampuannya untuk menyediakan hasil analisis produk secara *real time* yang dapat diunduh oleh pengguna kapan pun diperlukan.

Selain itu, fitur pelacakan produk memungkinkan Admin, *User*, dan *Viewer* untuk mengakses informasi terkait jumlah, jenis, dan kategori produk. Admin memiliki peran khusus dalam pengelolaan data *master*, yang mencakup tugas-tugas seperti menambah, mengubah, menghapus, dan mengelola data *master*. Terakhir, pengguna memiliki kemampuan untuk mengelola keamanan akun mereka sendiri dengan mengganti *password* sesuai kebutuhan. Dengan demikian, sistem memberikan lingkungan yang efektif dan terstruktur untuk pengelolaan data produk dengan berbagai fungsi yang disesuaikan dengan kebutuhan masing-masing pengguna.



Gambar 2. Gambaran Umum Perangkat Lunak

a. Kebutuhan Fungsional

Dalam pembangunan Aplikasi SEMB-DEBUG, kebutuhan fungsional dan non-fungsional diidentifikasi dan dipisahkan. Aplikasi ini dirancang untuk mendukung tiga jenis pengguna, yaitu Admin, Viewer, dan User, dengan spesifikasi akses yang telah ditetapkan dalam tabel 2.

Kebutuhan fungsional merupakan pernyataan tentang layanan yang harus disediakan oleh sistem, cara sistem merespons input khusus, dan perilaku sistem dalam situasi tertentu. Kebutuhan fungsional digunakan untuk mendefinisikan fungsionalitas proses.

Tabel 2. Deskripsi User

	Deskripsi
Admin	Admin memiliki akses penuh terhadap seluruh fitur yang disediakan dalam aplikasi, termasuk pengolahan data master dan pendataan produk.
Viewer	Viewer hanya memiliki hak akses untuk melihat riwayat pelacakan produk yang masuk dan keluar, melakukan penggantian <i>password</i> , serta meninjau data hasil analisis produk setelah melewati proses <i>debug</i> .
User	User hanya diberikan hak akses untuk melakukan pendataan produk, melihat data <i>master</i> tanpa hak pengelolaan, dan merubah <i>password</i> .

Kebutuhan fungsional aplikasi ditujukan kepada aktor yang memiliki akses ke proses sistem, sebagaimana yang tertera pada tabel 3.

Tabel 3. Kebutuhan Fungsional

Kode	Deskripsi	Aktor
------	-----------	-------

F001	Sistem dapat melakukan <i>login</i> dan <i>logout</i>	Admin, Viewer, User
F002	Sistem dapat menampilkan data pengguna	Admin
F003	Sistem dapat menambahkan data pengguna	Admin
F004	Sistem dapat mengubah data pengguna	Admin
F005	Sistem dapat menghapus data pengguna	Admin
F006	Sistem dapat mengupload data pengguna	Admin
F007	Sistem dapat menampilkan <i>master</i> data	Admin
F008	Sistem dapat menambahkan <i>master</i> data	Admin
F009	Sistem dapat mengubah <i>master</i> data	Admin
F010	Sistem dapat menghapus <i>master</i> data	Admin
F011	Sistem dapat mengupload <i>master</i> data	Admin
F012	Sistem dapat mengubah <i>password</i> pengguna	Admin, User, Viewer
F013	Sistem mampu melacak produk dengan menampilkan diagram berdasarkan kategori tertentu.	Admin, User, Viewer
F014	Sistem mampu menambahkan data produk masuk dengan kategori <i>defect</i> tertentu sebelum pembuatan <i>work order</i> .	Admin, User
F015	Sistem memiliki kemampuan untuk menambahkan <i>work order</i> .	Admin, User
F016	Sistem memiliki kemampuan untuk mengkonfirmasi <i>work order</i> dengan data produk tertentu.	Admin, User
F017	Sistem mampu memulai <i>work order</i> dengan menambahkan banyak data sekaligus pada produk yang sedang dalam proses <i>debug</i> , termasuk serial, status, <i>root cause</i> , <i>failure action</i> , analisis, nomor PCBA, lokasi komponen, dan catatan, berdasarkan jumlah produk yang telah diisi.	Admin, User
F018	Sistem memiliki kemampuan untuk mengubah data produk yang baru ditambahkan, meliputi serial, status, <i>root cause</i> , <i>failure action</i> , analisis,	Admin, User

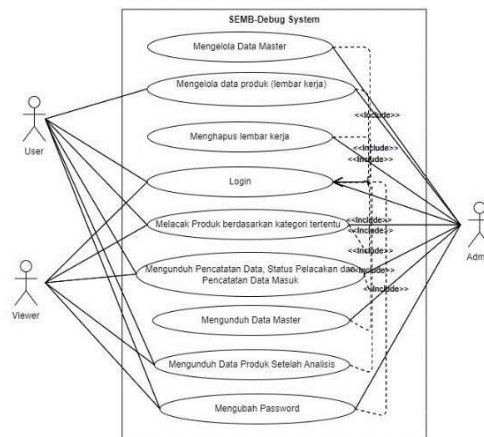
F019	nomor PCBA, lokasi komponen, dan catatan. Sistem memiliki kemampuan untuk menghapus data produk yang baru ditambahkan, meliputi serial, status, <i>root cause</i> , <i>failure action</i> , analisis, nomor PCBA, lokasi komponen, dan catatan.	Admin, User
F020	Sistem memiliki kemampuan untuk mengonfirmasi <i>work order</i> yang telah selesai dikerjakan.	Admin, User
F021	Sistem mampu melakukan perhitungan jumlah total produk setelah proses analisis serta berdasarkan kategori tertentu.	Admin, User, Viewer
F022	Sistem mampu melakukan pencarian data produk berdasarkan kategori tertentu.	Admin, User, Viewer
F023	Sistem dapat melakukan pencarian terhadap pengguna.	Admin
F024	Sistem dapat mengunduh <i>master data</i> .	Admin
F025	Sistem dapat mengunduh data Produk setelah analisis.	Admin, User, Viewer
F026	Sistem dapat mengunduh data <i>logging</i> , <i>tracking status</i> , dan <i>logging data in</i> saat pelacakan produk.	Admin, User, Viewer

NF003 Aplikasi harus memiliki desain antarmuka yang baik agar mudah digunakan oleh pengguna.

### 3.2. UML Diagram

#### a. Use Case Diagram

Diagram *Use Case* merupakan sebuah alat untuk memvisualisasikan interaksi antara pengguna dan sistem yang berjalan. Diagram ini menggambarkan fungsi-fungsi yang tersedia dalam sistem dan menetapkan hak akses untuk setiap aktor yang terlibat. Dalam konteks digitalisasi PT. Schneider Electric Manufacturing Batam (SEMB), Diagram *Use Case* menjadi instrumen penting untuk memahami bagaimana pengguna, seperti operator dan *engineer*, berinteraksi dengan aplikasi SEMB-DEBUG. Gambar 3 menampilkan Diagram *Use Case* yang digunakan dalam penelitian ini, memberikan gambaran yang jelas tentang fungsionalitas sistem dan hak akses pengguna.



Gambar 3. Diagram Use Case

#### b. Kebutuhan Non Fungsional

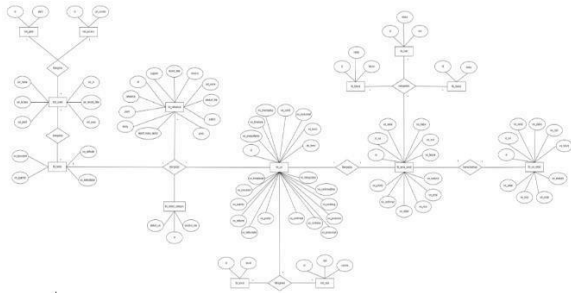
Kebutuhan non-fungsional merujuk pada aspek-aspek perilaku sistem yang mencakup kecepatan, keamanan, dan ketersediaan data. Hal ini sejalan dengan tujuan dibuatnya aplikasi SEMB-DEBUG oleh PT. Schneider Electric Manufacturing Batam (SEMB) dengan melakukan digitalisasi, di mana pemahaman yang mendalam tentang kebutuhan non-fungsional akan memastikan implementasi yang sukses dari aplikasi SEMB-DEBUG.

Tabel 4. Kebutuhan Non-fungsional

Kode	Kebutuhan Non Fungsional
NF001	Akses ke aplikasi dibatasi hanya untuk PT Schneider Electric Manufacturing Batam.
NF002	Aplikasi harus memberikan respons cepat terhadap permintaan pengguna.

#### b. Entity Relationship Diagram

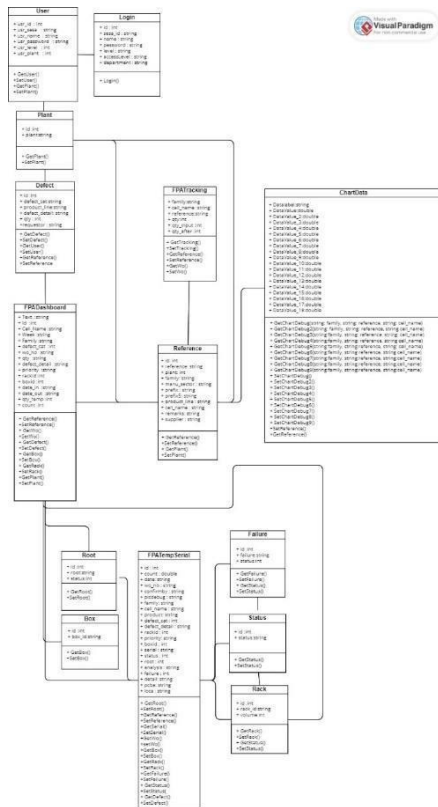
Gambar 3 menunjukkan ER Diagram yang mencakup beberapa tabel, termasuk *mst\_users*, *mst\_plant*, *mst\_access*, *mst\_rack*, *tbl\_boxid*, *tbl\_defect\_category*, *tbl\_failure*, *tbl\_reference*, *tbl\_root*, *tbl\_status*, *tbl\_temp*, *tbl\_temp\_serial*, *tbl\_wo*, dan *tbl\_wo\_detail*. Setiap tabel dalam diagram ini saling terhubung, mencerminkan kompleksitas hubungan data dalam basis data. Misalnya, tabel *tbl\_temp\_serial* dan *tbl\_wo\_detail* memiliki hubungan *one to one*, sementara *mst\_users* ke *tbl\_wo* serta *tbl\_wo* ke *tbl\_wo\_detail* memiliki hubungan *one to many*. Selain itu, tabel *tbl\_defect\_category* juga memiliki hubungan *one to many* dengan *tbl\_wo*. ER Diagram ini menunjukkan pentingnya pemahaman yang terstruktur tentang hubungan data untuk mendukung pengelolaan dan pelacakan data produk dalam proses debug pada aplikasi SEMB-DEBUG.



Gambar 4. Entity Relationship Diagram

c. Class Diagram

*Class diagram* adalah diagram yang sering digunakan dalam pemrograman berorientasi objek untuk menunjukkan struktur kelas dan relasi antar kelas. Diagram ini memvisualisasikan atribut dan metode dari setiap kelas serta hubungannya dengan kelas lain dalam sistem. Pada class diagram yang digunakan dalam aplikasi SEMB-DEBUG, terdapat beberapa kelas seperti kelas User, Login, Plant, Defect, Rack, Box, Status, Failure, Root, Reference, FPATempSerial, FPADashboard, ChartData dan FPA Tracking. Detail struktur dan relasi antar kelas tersebut dapat dilihat dalam gambar berikut ChartData dan FPA Tracking. Detail struktur dan relasi antar kelas tersebut dapat dilihat dalam gambar 5.



Gambar 5. Class Diagram

4. Implementasi dan Pengujian

4.1. Implementasi

1) Halaman Login

Halaman login merupakan titik awal akses saat mengunjungi aplikasi *web*. Pengguna diminta untuk memasukkan SESA dan kata sandi. Jika autentikasi berhasil, pengguna akan diarahkan ke halaman *dashboard*. Namun, jika autentikasi gagal, pengguna tidak akan memiliki hak akses atau data pengguna belum tersedia dalam *database*. Semua pengguna dapat mengakses halaman ini.



Gambar 6. Halaman Login

2) Halaman Dashboard

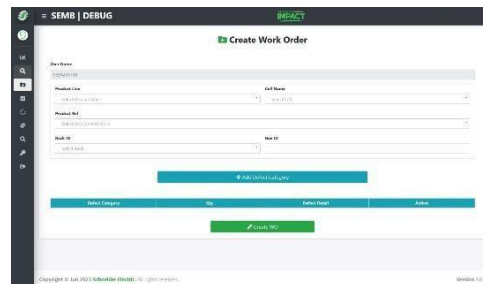
Halaman *dashboard* ditampilkan setelah pengguna berhasil *login*. Di halaman *dashboard*, admin dapat menemukan diagram yang memungkinkan pelacakan produk berdasarkan kategori tertentu. Halaman ini dapat diakses oleh semua pengguna.



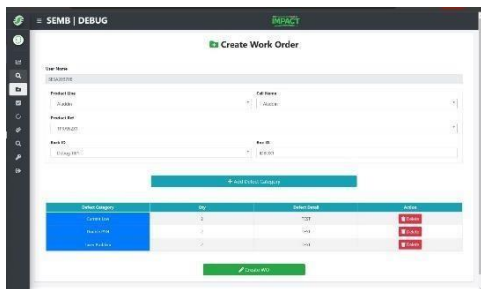
Gambar 7. Halaman Dashboard

3) Halaman Create Work Order

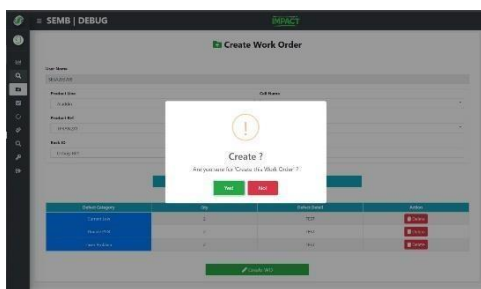
Halaman *Create Work Order (Work Order)* didesain untuk menyesuaikan alur kerja operator di lapangan, sehingga penggunaan menjadi lebih mudah. Halaman ini berfungsi sebagai tempat untuk membuat lembar kerja baru dengan menginput data dan *multiple defect* dalam satu lembar kerja secara simultan.



Gambar 8. Halaman Create Work Order



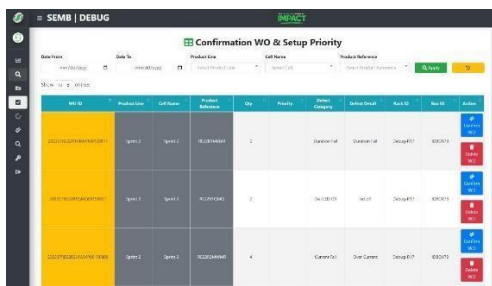
Gambar 9. Halaman Create Work Order Multiple Defect



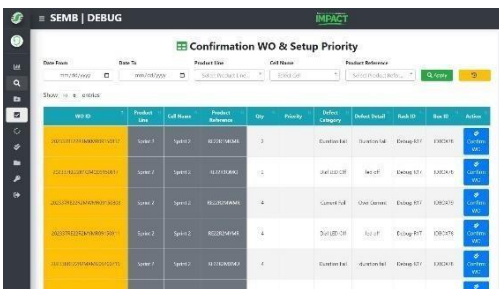
Gambar 10. Halaman Create Work Order

#### 4) Halaman Confirm Work Order

Halaman *Confirm Work Order* adalah tempat untuk mengkonfirmasi lembar kerja yang telah dibuat sebelumnya dan siap untuk dikerjakan pada tahap selanjutnya. Pada halaman ini, terdapat perbedaan fungsionalitas di mana admin memiliki akses untuk menghapus data, sedangkan *user* atau operator hanya dapat melakukan konfirmasi tanpa memiliki kemampuan untuk menghapusnya.



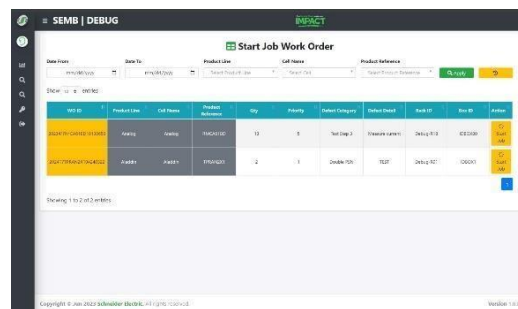
Gambar 11. Halaman Confirm Work Order Untuk Admin



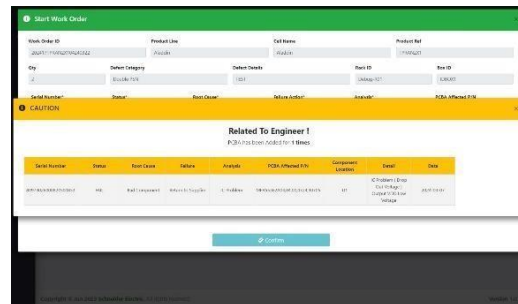
Gambar 12. Halaman Confirm Work Order Untuk User

#### 5) Halaman Start Debug

Pada halaman ini pengguna baik itu user maupun admin dapat melakukan pendataan pada produk berdasarkan lembar kerja yang telah di konfirmasi sebelumnya. Disinilah tahap analisis produk dimulai. Operator mengisi lembar kerja produk yang telah di daftarkan mulai dari serial, hasil analisis hingga meninggalkan catatan jika dibutuhkan berdasarkan jumlah kuantitas produk dalam lembar kerja tersebut. Seperti pada gambar 14 dan 15.

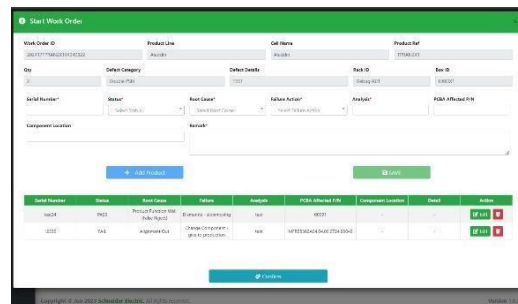


Gambar 13. Halaman Start Debug

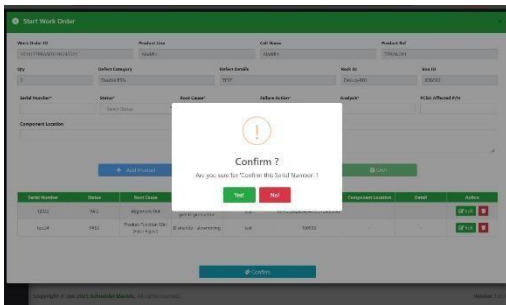


Gambar 14. Modal Peringatan

Hal ini terjadi dikarenakan PCBA number yang diinput sudah pernah melewati proses *debug*, sehingga operator mendapat notifikasi untuk dapat menghubungi *engineer* dan melaporkan bahwa produk tersebut sudah melewati tahapan ini. Setelah melakukan pengisian data, operator dapat mengubah detail data produk yang telah disikan sebelum di konfirmasi serta menghapusnya seperti pada gambar 15 dan 16.



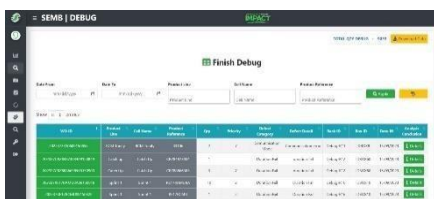
Gambar 15. Modal Pengisian Data



Gambar 16. Modal Konfirmasi Produk

### 6) Halaman *Finish Debug*

Pada Halaman ini, semua pengguna dapat melihat hasil analisis produk dan jumlah keseluruhannya. Pengguna juga dapat memfilter data untuk memudahkan dalam pencarian data. Pengguna juga dapat mengunduh data baik keseluruhan maupun data tertentu.



Gambar 17. Halaman *Finish Debug*



Gambar 18. Halaman *Finish Debug* (Modal Detail)



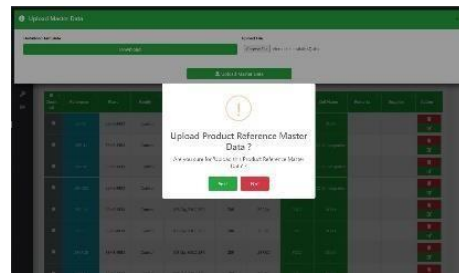
Gambar 19. Halaman *Finish Debug* (Download Data)

### 7) Halaman *Master Data*

Pada Halaman ini berisi data yang digunakan sebagai acuan dalam pendataan baik itu pendataan data produk maupun manajemen pengguna. Sehingga sistem dapat mengklasifikasi data tertentu dan terintegritas antar data satu dengan yang lainnya. admin dapat mengolah *master data*.



Gambar 20. Halaman *Master Data* Untuk Admin



Gambar 21. Mengunggah *Master Data*

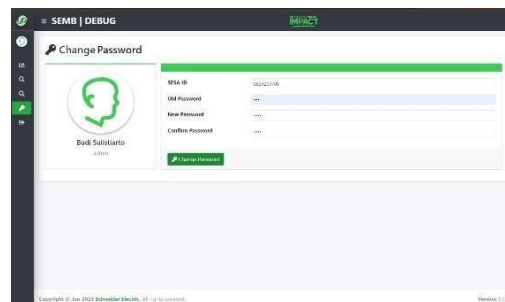
Terdapat sedikit perbedaan dengan halaman *master data* admin, *Master Data* untuk akses *user* hanya dapat dilihat dan tidak dapat mengelola data.



Gambar 22. Halaman *Master Data* Untuk User

### 8) Halaman *Ubah Password*

Pada Halaman ini semua pengguna dapat mengubah *password* dengan memasukkan *sesa*, *password* lama, *password* baru dan konfirmasi *password* baru.



Gambar 23. Halaman *Change Password*

### 9) *Filter Dashboard*

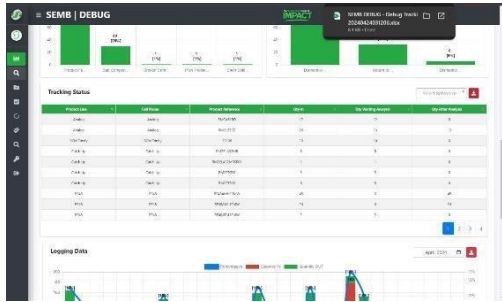
Fitur ini digunakan untuk memudahkan pelacakan data produk baik sebelum maupun sesudah analisis.



Gambar 24. *Filter Dashboard*

### 10) *Download Tracking Status, Logging Data* dan *Logging Data IN*

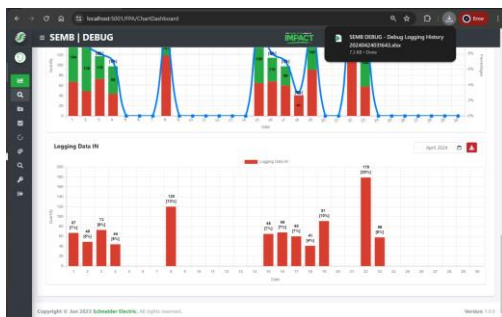
Pada fitur ini pengguna dapat mengunduh *Tracking Status, Logging Data* dan *Logging Data IN*.



Gambar 25. Tracking Status



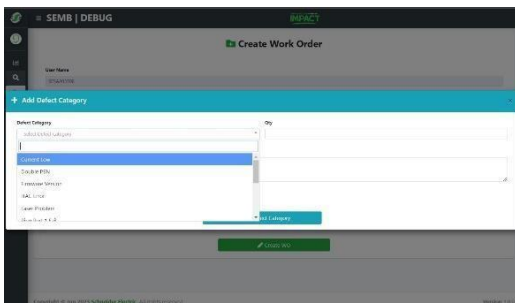
Gambar 26. Logging Data



Gambar 27. Logging Data IN

### 11) Modal Tambah *Multiple Defect Category*

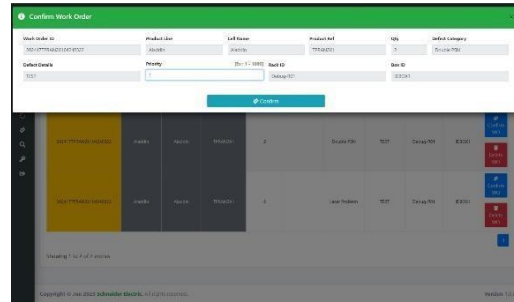
Pada modal penambahan data *Defect Category* Pengguna harus mengisi *Defect Category* yang telah terklasifikasi berdasarkan jenis produk yang telah dipilih sebelumnya, kemudian memasukan berapa jumlah kuantiti produk dengan *defect* tersebut dan detail kerusakannya. Setelah klik tambah *defect*, maka akan memungkinkan pengguna untuk menambah *defect* baru pada jenis produk yang sama sebelum *create work order*.



Gambar 28. Modal Tambah Defect Category

### 12) Modal Konfirmasi *Work Order*

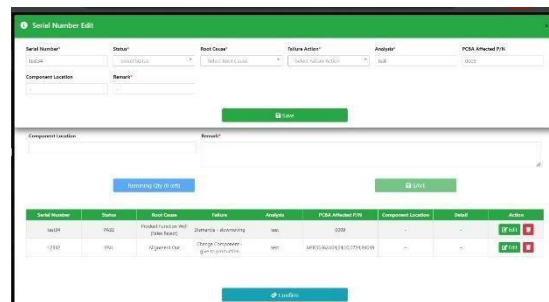
Pada modal ini pengguna secara opsional dapat mengisi jumlah prioritas produk dalam lembar kerja tersebut sebelum mengkonfirmasi.



Gambar 29. Modal Confirm Work Order

### 13) Modal *Edit Serial*

Pada modal ini memungkinkan pengguna untuk memperbaiki kesalahan pengisian ketika mengisi data pada halaman *start debug*. Pengguna dapat mengubah nomor serial, status hingga nomor *pcba*.



Gambar 30. Modal Edit Serial

## 4.2. Pengujian

Aplikasi SEMB-DEBUG ini diuji menggunakan metode *black-box*, yang merupakan pengujian perangkat lunak yang berfokus pada fungsionalitasnya. Dengan metode ini, perangkat lunak diuji untuk memastikan kesesuaian dengan kebutuhan fungsional yang diinginkan. Pengujian ini dilakukan oleh karyawan PT. Schneider Electric Manufacturing Batam (SEMB) meliputi *CS&Q Engineer*, *CS&Q Manager*, operator.

Tabel 5. Hasil Pengujian

Skenario <i>Use Case</i>	Deskripsi Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
<i>Login</i>	Memasukan SESAs dan <i>password</i> pengguna yang benar kemudian klik <i>Log in</i>	Berhasil melakukan <i>login</i> sebagai admin dan <i>user</i>	Berhasil
<i>Login</i>	Memasukan SESAs dan <i>password</i> pengguna yang salah kemudian klik <i>login</i>	Gagal melakukan <i>login</i> sebagai admin dan <i>user</i>	Gagal
Mengelola <i>Master Data</i>	Admin menambahkan <i>master data</i> baru, mengubah data, menghapus, mengupload data dan mengunduh <i>master data</i>	Admin berhasil mengelola data lokasi	Berhasil
Mengelola data produk ( <i>work order</i> )	Admin dan <i>user</i> menambahkan <i>defect category</i> , menghapus data <i>defect category</i> , serta membuat lembar kerja baru	Admin dan <i>user</i> berhasil mengelola data produk	Berhasil
Menghapus <i>Work Order</i>	Admin menghapus lembar kerja pada tabel konfirmasi <i>work order</i> dengan menekan tombol hapus	Admin berhasil menghapus <i>work order</i>	Berhasil
Melacak produk berdasarkan kategori tertentu	Semua pengguna melacak produk berdasarkan kategori tertentu dengan menyaring data	Semua pengguna berhasil melacak data produk berdasarkan kategori tertentu	Berhasil
Mengunduh <i>Logging Data, Tracking Status</i> dan <i>Logging Data IN</i>	Semua pengguna mengunduh <i>Logging Data, Tracking Status</i> dan <i>Logging Data IN</i> dengan menekan tombol unduh	Semua pengguna berhasil mengunduh <i>Logging Data, Tracking Status</i> dan <i>Logging Data IN</i>	Berhasil
Menngunduh <i>master data</i>	Admin mengunduh <i>master data</i> dengan menekan tombol unduh	Admin berhasil mengunduh <i>master data</i>	
Mengunduh data produk setelah analisis	Semua Pengguna Mengunduh data produk setelah analisis dengan menekan tombol unduh	Semua pengguna berhasil mengunduh data produk setelah analisis	
Mengubah <i>Password</i>	Semua pengguna mengubah kata sandi dengan memasukan SESAs, kata sandi lama, kata sandi baru, kata sandi konfirmasi, lalu menekan tombol <i>Change Password</i>	Semua pengguna berhasil mengubah kata sandi	Berhasil
<i>Logout</i>	Semua pengguna melakukan <i>logout</i> dengan menekan tombol keluar pada <i>sidebar</i>	Semua pengguna berhasil keluar dari aplikasi	Berhasil

## 5. Kesimpulan

Aplikasi SEMB-DEBUG ini bertujuan melakukan pengelolaan dan pelacakan data produk selama proses *debug*. Dengan adanya aplikasi ini, tim *engineer*, operator, dan *quality control* dapat melakukan pengelolaan data produk dan pelacakan secara *online*. Proses digitalisasi yang dilakukan dengan menghasilkan aplikasi bernama SEMB-DEBUG mampu menggantikan *Excel* sebagai media pendataan produk sehingga dapat diakses secara *online* dan *real time* selama pengguna dapat terkoneksi pada *server* perusahaan dengan memanfaatkan *VPN* yang disediakan.

Pentingnya adopsi teknologi *digital* ditekankan, menggarisbawahi bahwa SEMB-DEBUG bukan hanya digunakan oleh satu departemen, tetapi untuk seluruh tim. Dengan demikian, aplikasi SEMB-DEBUG mendorong perubahan positif dalam mentransformasi penggunaan *Excel* menjadi *digital* dengan penggunaan teknologi di PT. Schneider Electric Manufacturing Batam (SEMB).

## Daftar Rujukan

- [1] M. Safitri and D. Nirmala, "Aplikasi Inventory Manajemen Aset Berbasis Web," IJCIT Indonesian J. Comput. Inf. Technol., vol. 4, no. 1, pp. 21–26, 2019.
- [2] Wijaya, N. Febriyanti, A. R. and A. Wibowo, "Aplikasi Pengelolaan Data Kepegawaian Berbasis Web Pada Pt. Pelayaran Sakti Inti Makmur Palembang," Jurnal Sisfokom (Sistem Informasi Dan Komputer), vol. 9, no. 1, pp. 42-50, 2020.
- [3] N. Nurhadi, BUKU AJAR: BAHASA PEMROGRAMAN PEMROGRAMAN DATABASE MENGGUNAKAN VISUAL BASIC. NET (VB. NET) DAN MICROSOFT SQL SERVER CONTOH PROGRAM: PEMBUATAN PROGRAM APLIKASI INVENTORI, 2022.
- [4] Suprayogi and A. Rahmanesa, "Penerapan Framework Bootstrap Dalam Sistem Informasi Pendidikan SMA Negeri 1 Pacet Cianjur Jawa Barat," J. Teknol. Inf. dan Komun., vol. 6, no. 2, p. 30, 2019.
- [5] M. Masri and E. Lasmi, "Perancangan Media Pembelajaran Tata Surya Menggunakan Teknologi Augmented Reality Dengan Metode Markerless," vol. 4, no. 1, pp. 40-46, 2019.
- [6] S. Mariko, "Aplikasi website berbasis HTML dan JavaScript untuk menyelesaikan fungsi integral pada mata kuliah kalkulus," Jurnal Inovasi Teknologi Pendidikan, vol. 6, no. 1, pp. 80-91, 2019.
- [7] M. F. A. Nasrullah, R. Yulius, and L. Pratama, "Aplikasi Issue Tracking System Berbasis Web (Studi Kasus: PT Mitra Kuadran Teknologi)," Teknik: Jurnal Ilmu Teknik dan Informatika, vol. 2, no. 2, pp. 47-50, 2022.
- [8] N. Hidayati, "Pengembangan Sistem Informasi Pembelian Obat Secara Kredit Pada Apotik Dengan Menggunakan Model View Controller (MVC)," Jurnal Sains Komputer dan Informatika (J-SAKTI), vol. 4, no. 2, pp. 457-471, 2020.
- [9] A. Wahid, "Analisis metode waterfall untuk pengembangan sistem informasi," J. Ilmu-ilmu MAInform. dan Manaj. STMIK, pp. 1-5, 2020.
- [10] P. Putra, F. Andriyanto, K. Karisman, and T. D. M. Harti, "Pengujian Aplikasi Point of Sale Menggunakan Blackbox Testing," Jurnal Bina Komputer, vol. 2, no. 1, pp. 74-78, 2020.