PENGEMBANGAN FITUR MANAJEMEN HAK AKSES PENGGUNA PADA SISTEM INTERNAL PERUSAHAAN DENGAN METODE *ROLE-BASED ACCESS CONTROL* (RBAC)

(Studi kasus PT. XYZ)

PROYEK AKHIR

Disusun Oleh:

Rizqi Vela Syifa

3312301013

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

Assalmualaikum Wr. Wb.

Puji dan syukur tak henti Penulis panjatkan kehadirat Allah SWT yang telah melimpahkan Rahmat dan Karunia-Nya karena atas izin-Nya lah peneliti dapat menyelesaikan proyek akhir ini dengan judul "**Pengembangan Fitur Manajemen Hak Akses Pengguna Pada Sistem Internal Perusahaan Menggunakan Metode *Roled-Based Access Control* (RBAC)**". Penulisan proyek akhir ini diselesaikan guna melengkapi Program Diploma III Teknik Informatika Politeknik Negeri Batam.

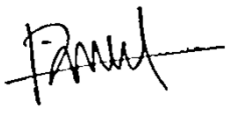
Dalam penyelesaian studi dan Penulisan proyek akhir ini, Penulis banyak memperoleh bantuan baik pengajaran, bimbingan dan arahan dari berbagai pihak. Untuk itu Penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu Penulis dalam menyelesaikan proyek akhir ini, diantaranya yaitu kepada:

1. Tuhan Yang Maha Esa.
2. Ibu Yeni Rokhayati, S.Si., M.Sc., selaku Kepala Program Studi Teknik Informatika.
3. Bapak Ahmad Hamim Thohari, S.S.T., M.T., selaku Kepala Jurusan Teknik Informatika.
4. Bapak Muhammad Idris, S.Tr., M.Tr.Kom. selaku Dosen Pembimbing yang telah meluangkan waktu ditengah kesibukkan beliau, memberikan kritik, saran dan pengarahan kepada Penulis dalam proses Penulisan proyek akhir ini.
5. Seluruh Dosen dan Staff Program Studi Diploma III Teknik Informatika Politeknik Negeri Batam yang telah membekali ilmu pengetahuan yang sangat bermanfaat dan tak ternilai harganya.
6. Bapak Wahyu Hidayat, selaku Manager IT, yang telah memberikan kesempatan, arahan, serta dukungan kepada Penulis selama pelaksaan kegiatan magang dan proses Penulisan proyek akhir ini.

7. Bapak Mahmud Amma Rizki, selaku pembimbing magang, yang telah memberikan bimbingan, arahan, serta masukan yang berharga kepada Penulis selama pelaksaan kegiatan magang hingga proses Penulisan proyek akhir ini.
8. Bapak Habib Syuhada, Bapak Iqbal, Ibu Dini, serta seluruh mentor, yang dengan sabar memberikan bimbingan, ilmu, dan pengalaman berharga selama kegiatan magang hingga proses Penulisan proyek akhir ini.
9. Kedua orang tua Penulis, Bapak Jumari dan Ibu Sumiatun, untuk beliau berdualah proyek akhir ini Penulis persembahkan. Terima kasih atas segala kasih sayang yang diberikan dalam membesarkan dan membimbing Penulis selama ini sehingga Penulis dapat terus berjuang.
10. Saudara Penulis, Dhea Anggi Nuraini. Terima kasih selalu percaya pada mimpi-mimpi Penulis.
11. Terimakasih juga kepada seluruh teman- teman magang saya, teman seperjuangan bimbingan proyek akhir ini, serta Teman-teman terdekat saya Alia Pramestia dan Sarah Isnaini.

Semoga semua motivasi, semangat, ilmu yang selalu saya ingat serta do'a yang diberikan mendapat imbalan dari Allah SWT sebagai amal dan ibadah. Penulis menyadari bahwa proyek akhir ini jauh dari kata sempurna, oleh karena itu kritik dan saran dari berbagai pihak sangat Penulis harapkan demi perbaikan kedepan.

Batam, 15 Mei 2026



Rizqi Vela Syifa
NIM. 3312301013

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

ABSTRAK

Manajemen hak akses pengguna merupakan aspek penting dalam menjaga keamanan dan efisiensi operasional sistem internal perusahaan. Di PT. XYZ, proses manajemen hak akses masih dilakukan secara manual melalui formulir kertas dan berlapis persetujuan, sehingga berpotensi menimbulkan kesalahan dan pemberian akses yang tidak sesuai kebutuhan. Proyek akhir ini bertujuan untuk mengembangkan fitur manajemen hak akses pengguna berbasis *Role-Based Access Control* (RBAC) yang dapat mengatur hak akses berdasarkan peran pengguna seperti *Requestor*, *Head of Department Requestor*, *Head of Department IT* dan *Administrator*. Metode pengembangan yang digunakan adalah *Agile Development*, dengan tahapan meliputi perencanaan, perancangan, implementasi, dan pengujian sistem. Hasil dari proyek ini berupa aplikasi berbasis web yang mampu mengelola proses *request*, persetujuan, dan pembuatan akun pengguna secara digital, terstruktur, dan aman. Sistem ini berhasil meminimalisasi risiko *over-privilege* dan kesalahan manusia karena otorisasi izin akses (*permissions*) tidak lagi diberikan secara manual kepada masing-masing akun indivi, melainkan dikelompokkan secara terpusat dan diikat secara otomatis pada tingkatan peran (*role*). Selain itu, berdasarkan hasil pengujian *User Acceptance Testing* (UAT), seluruh 42 skenario pengujian fungsional telah dinyatakan berhasil (*Pass*), membuktikan bahwa alur persetujuan dan manajemen akun berjalan sesuai dengan spesifikasi. Dengan penerapan RBAC, aplikasi ini diharapkan dapat memberikan kontribusi signifikan dalam peningkatan keamanan dan konsistensi administrasi manajemen akses di PT. XYZ.

Kata Kunci: Manajemen Hak Akses, RBAC, Sistem Internal, Agile Development, Aplikasi Web.

BAB I PENDAHULUAN

1.1. Latar Belakang

Pengelolaan hak akses yang terstruktur dengan baik sangat dibutuhkan untuk menjaga keamanan dan kelancaran operasional internal perusahaan (Alim et al., 2024). Hal ini semakin penting bagi perusahaan berskala besar seperti PT. XYZ, yang memiliki ribuan pengguna terdiri dari karyawan, subkontraktor dan klien dengan kebutuhan akses yang berbeda. PT XYZ menggunakan satu sistem internal perusahaan yang tersentralisasi pada sebuah portal aplikasi yang terdiri dari berbagai macam fitur atau menu utama dalam membantu operasional Perusahaan seperti *Production & Quality*, *Engineering*, dan *Warehouse*.

Saat ini, mekanisme manajemen hak akses di PT. XYZ masih berjalan secara manual. Setiap pengguna yang membutuhkan akun, Requestor harus mengisi *paper form* dan meminta persetujuan dari *Head of Department* terkait, kemudian dilanjutkan ke *Head of Department* IT. Setelah seluruh persetujuan diperoleh, akun baru dibuat oleh Administrator. Proses yang kompleks ini tidak hanya memakan waktu, tetapi juga berpotensi menimbulkan kesalahan manusia (*human error*) (Putrawan & Harahap, 2024) serta mengurangi risiko keamanan akibat pemberian hak akses yang berlebihan (*over-privilege*) (Yuricha & Phan, 2023), mengingat proses manual sebelumnya tidak memiliki pemetaan peran yang jelas untuk tiap pengguna.

Melihat kondisi tersebut, diperlukan suatu manajemen hak akses yang terstandarisasi dan terstruktur dengan baik agar pengguna mendapatkan izin akses yang benar-benar sesuai dengan peran dan tanggung jawab jabatannya (Alim et al., 2024). Standarisasi manajemen hak akses menjadi bagian penting dalam sistem pengendalian internal perusahaan. Hal ini bertujuan untuk menjaga konsistensi pemberian otorisasi, mencegah penyalahgunaan oleh akun yang tidak sah atau tidak terkelola, dan mempermudah pemantauan log aktivitas pengguna (Sagita et al., 2026).

Salah satu pendekatan yang dapat diterapkan untuk menjawab kebutuhan tersebut adalah *Role-Based Access Control* (RBAC). Dengan RBAC, hak akses ditentukan berdasarkan peran yang dimiliki pengguna di dalam organisasi, bukan ditetapkan secara individual. Misalnya, seorang staff hanya diberikan akses untuk menginput dan melihat data, sedangkan seorang manajer memiliki tambahan hak untuk melakukan persetujuan dan menghasilkan laporan. Penerapan RBAC menjadikan manajemen akses lebih konsisten, serta mengurangi risiko *over-privilege* (Prasetia, 2024; Zikra et al., 2025).

1.2. Rumusan Masalah

Berdasarkan latar belakang tersebut maka Penulis merumuskan masalah sebagai berikut:

1. Bagaimana mengimplementasikan manajemen hak akses pengguna pada aplikasi agar proses manajemen akses lebih terstruktur, efisien, dan aman?
2. Bagaimana manajemen hak akses pengguna dapat memastikan setiap pengguna hanya memperoleh hak akses sesuai fungsi dan tanggung jawabnya?

1.3. Tujuan

Tujuan dari tugas akhir ini adalah sebagai berikut:

1. Mengimplementasikan manajemen hak akses pengguna pada aplikasi menggunakan metode *Role-Based Access Control* (RBAC) untuk mewujudkan sistem yang terstruktur, efisien, dan aman.
2. Menerapkan *Role-Based Access Control* (RBAC) secara benar dengan mendefinisikan *Role* serta mapping hak akses berdasarkan fungsi dan tanggung jawabnya.

1.4. Batasan Masalah

Batasan masalah merupakan faktor yang dilakukan dalam penelitian sehingga dapat diketahui dengan jelas faktor tersebut yang

digunakan dalam penelitian. Pada permasalahan ini, ada beberapa batasan masalah yaitu:

1. Ruang lingkup pengaturan hak akses pada penelitian ini difokuskan pada level aplikasi, tanpa mencakup aspek keamanan jaringan dan server.
2. Fokus hanya pada pendekatan *Role-Based Access Control* (RBAC), tidak membahas metode lain.
3. Sistem ini tidak mencakup menu pengelolaan sumber daya manusia (*Human Resource Information System*) secara langsung. Pembaruan data jabatan (*Position*) dilakukan secara manual oleh Administrator berdasarkan informasi atau dokumen fisik resmi dari departemen HRD/Kepegawaian.

1.5. Manfaat

Manfaat yang diharapkan dengan adanya proyek akhir ini adalah:

1. Membantu perusahaan mengelola hak akses pengguna secara terstruktur dan efisien, mendukung peningkatan produktivitas operasional Perusahaan.
2. Menggantikan proses manual berbasis kertas menjadi sistem digital yang terintegrasi, berkontribusi pada inovasi dan digitalisasi proses bisnis.
3. Mengurangi potensi *human error* dan *over-privilege* melalui tata kelola akses yang transparan dan akuntabel.

1.6.Kolaborasi dan Sinergi dalam Pelaksanaan Proyek

Proyek Akhir ini dilaksanakan secara individu oleh mahasiswa. Meskipun dikerjakan secara mandiri, pelaksanaannya tetap melibatkan kolaborasi dan sinergi yang kuat dengan berbagai pihak sebagai tim pendukung untuk memastikan kualitas sistem sesuai dengan standar industri. Bentuk kolaborasi dan sinergi yang dilakukan selama pelaksanaan proyek adalah sebagai berikut:

1. Diskusi rutin dengan pembimbing lapangan, Bapak Mahmud Amma

Rizki selaku pembimbing magang di PT. XYZ, dilakukan secara berkala untuk memantau progres dan mendapatkan arahan teknis terkait implementasi fitur manajemen hak akses pengguna (RBAC) di lingkungan perusahaan.

2. Koordinasi dengan tim internal, yaitu jajaran mentor IT (Bapak Habib Syuhada, Bapak Iqbal, dan Ibu Dini), dilakukan dalam hal integrasi data dengan sistem internal perusahaan serta pengujian bersama untuk memastikan sistem yang dikembangkan sejalan dengan kebijakan keamanan TI perusahaan.
3. Konsultasi dengan dosen pembimbing, Bapak Idris, dilakukan secara rutin setiap minggu untuk mendapatkan bimbingan akademik, memastikan kesesuaian metodologi pengembangan perangkat lunak, serta memastikan kualitas penulisan laporan sesuai standar Politeknik Negeri Batam.
4. Pengumpulan umpan balik dari calon pengguna dilakukan melalui proses *User Acceptance Testing* (UAT) dengan pihak departemen terkait di PT. XYZ dan pengujian langsung untuk memastikan antarmuka sistem mudah digunakan (*user-friendly*) dan fungsionalitasnya sesuai dengan kebutuhan hak akses.

BAB II TINJAUAN PUSTAKA

2.1 Tinjauan Solusi dan Sistem Terkait

Berikut beberapa penelitian terkait dari "Pengembangan Fitur Manajemen Hak Akses Pengguna Pada Sistem Internal Perusahaan Dengan Metode *Role-Based Access Control* (RBAC)"

Tabel 2. 1 Contoh Tinjauan Aplikasi Sejenis

Peneliti/Tahun	**Judul/Topik Penelitian**	**Metode/Teknologi**	**Kelebihan**	**Kekurangan/ Celah**
Zikra et al., (2025)	Perancangan Sistem Manajemen Proyek Berbasis *Website* dengan *Multi-Level Role-Based Access Control*	*Waterfall, Laravel 12, Filament Shield, Tailwind CSS*	Menerapkan multi-level RBAC sehingga akses tiap peran (Super Administrator, Manajer Proyek, Ketua Tim, Pengembang, Penguji QA, dan Klien) lebih aman, transparan, dan terstruktur.	Tidak memiliki fitur cetak laporan proyek sehingga menyulitkan penyampaian laporan fisik ke pihak manajemen.
Setiawan et al., (2024)	Rancang Bangun *Secure Document* Management *Sistem* (DMS) Menggunakan Metode *Agile-SSDLC*	*Agile, MySQL, Codeigniter 4*	Penerapan RBAC membedakan hak akses Administrator dan editor sehingga sistem lebih aman dan terstruktur, serta didukung *Agile-SSDLC* dengan *SonarQube* untuk deteksi kerentanan.	Hanya ada *Role* Administrator & editor, belum mendukung *Role* lebih kompleks (misalnya *viewer*, auditor, atau manager).
Putrawan & Harahap (2024)	Implementasi Metode *Role-Based Access*	*Waterfall*	Semua fitur utama berjalan valid sesuai	Belum ada integrasi sistem

	Control Pada Aplikasi *E-Raport* di MIN 15 Langkat Berbasis *Android*		*Role* masing-masing (Administrator, guru, wali kelas, murid/wali murid).	keamanan tambahan seperti enkripsi data
Prasetia (2024)	*Role-Based Access Control* (RBAC) untuk Sistem Otorisasi Terpusat Berbasis Flask Studi Kasus PT. XYZ	*Python, Flask, PostgreSQL*,	Meningkatkan keamanan, efisiensi, dan efektivitas pengelolaan hak akses secara terpusat.	Tampilan aplikasi belum responsif di perangkat mobile dan kurang fleksibel untuk menangani skenario akses yang kompleks.
Penggalih & Silmina (2025)	Implementasi *Dashboard* Proyek Dengan Sistem RBAC Pada PT.XYZ	*XP, Laravel, Spatie, PostgreSQL*	Meningkatkan keamanan hak akses, transparansi data, serta efektivitas manajemen proyek melalui dashboard terpusat.	Belum mendukung *import Excel* secara paralel sehingga performa sistem masih terbatas pada skala besar.
Rizqi Vela Syifa (2025)	Pengembangan Fitur Manajemen Hak Akses Pengguna Pada Sistem Internal Perusahaan Dengan Metode *Role-Based Access Control* (RBAC) (Studi kasus PT. XYZ)	*Agile, PostgreSQL, Next Js, Nest Js*	Penerapan metode RBAC dapat meningkatkan keamanan aplikasi karena hak akses setiap pengguna dibatasi sesuai perannya, sehingga memudahkan pengelolaan pengguna dan mengurangi risiko penyalahgunaan akses.	Menyediakan fitur Manajemen hak aksesberbasis peran (RBAC) serta unduh EXCEL untuk akun pengguna aktif, sehingga akses lebih terstruktur dan mudah disesuaikan.

2.2 Landasan Konsep dan Teknologi

2.2.1 Pengembangan

Secara umum, pengembangan sistem dapat diartikan sebagai proses mengubah kebutuhan dan perancangan menjadi sebuah perangkat lunak yang utuh. Hal ini sering kali melibatkan digitalisasi dari sistem yang sebelumnya berjalan manual agar menjadi lebih efektif dan efisien (Jayanti & Aslamiyah, 2024).

2.2.2 Manajemen Hak Akses

Manajemen akses adalah proses pengelolaan hak dan izin pengguna dalam mengakses sumber daya sistem informasi, seperti data, aplikasi, dan layanan. Proses ini meliputi tahapan identifikasi, autentikasi saat masuk ke sistem, dan pemberian otorisasi kepada pengguna. Tujuannya adalah untuk menutup celah keamanan sekaligus memastikan semua aktivitas pengguna berjalan sesuai dengan standar operasional perusahaan (Alim et al., 2024).

2.2.3 Sistem Pengendalian Internal

Sistem pengendalian internal mencakup berbagai prosedur dan struktur yang diterapkan oleh manajemen perusahaan. Pada implementasi sistem informasi, pengendalian ini berfungsi untuk melindungi data kredensial, menjaga keandalan informasi, serta memastikan bahwa proses pengajuan akses oleh *requestor* (mulai dari Admin Departemen hingga Admin IT) berjalan sesuai dengan kebijakan keamanan yang ada (Sagita et al., 2026).

2.2.4 Role-Based Access Control (RBAC)

Role-Based Access Control (RBAC) Adalah model kontrol akses yang membatasi akses ke sumber daya dalam sistem berdasarkan peran yang dimiliki oleh pengguna. Dalam RBAC, setiap pengguna diberikan peran tertentu, dan peran tersebut menentukan hak akses yang dimiliki untuk mengakses objek-objek sistem. Penerapan RBAC membuat pengelolaan hak akses menjadi lebih tertata dan efisien. Izin akses (*permissions*) tidak lagi diberikan satu per satu kepada setiap pengguna,

melainkan dikelompokkan berdasarkan peran masing-masing di dalam sistem (Yuricha & Phan, 2023).

2.2.5 Website

Saat ini, fungsi *website* tidak hanya terbatas sebagai media informasi atau promosi. *Website* juga banyak dimanfaatkan sebagai platform utama untuk mengelola basis data, mendukung alur kerja (*workflow*) operasional, serta mempermudah komunikasi data internal perusahaan yang dapat diakses dengan mudah melalui peramban web (Gumilar & Firmansyah, 2023).

2.2.6 Next.js

Next.js adalah sebuah *framework* berbasis *React* yang menyediakan fondasi untuk membangun aplikasi web. Sebagai *framework*, *Next.js* tidak hanya mengatur kebutuhan konfigurasi dan perkakas *React*, tetapi juga menawarkan struktur, fitur, serta optimalisasi tambahan agar pengembangan aplikasi menjadi lebih mudah dan efisien (Vercel, 2025).

2.2.7 NestJS

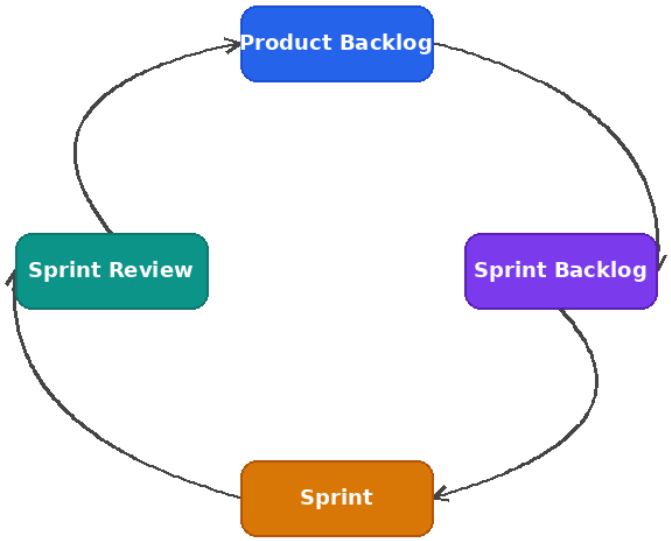
NestJS adalah kerangka kerja untuk membangun aplikasi sisi server *Node.js* yang efisien dan skalabel. *Framework* ini menggunakan *JavaScript* modern, dibangun dengan *TypeScript* serta sepenuhnya mendukung penggunaannya, namun tetap memberikan fleksibilitas bagi pengembang untuk menulis kode menggunakan *JavaScript* murni. Selain itu, *Nest* menggabungkan konsep Pemrograman Berorientasi Objek (OOP), Pemrograman Fungsional (FP), dan Pemrograman Reaktif Fungsional (FRP) (NestJS, 2025).

2.2.8 PostgreSQL

PostgreSQL adalah sistem basis data objek-relasional sumber terbuka yang kuat, yang menggunakan dan memperluas bahasa *SQL*, serta menyediakan banyak fitur untuk menyimpan dan mengelola beban data yang kompleks secara aman (PostgreSQL, 2025).

2.3 Metode Pengembangan Produk

Penelitian ini menggunakan metode *Agile* dengan model proses Scrum dalam pengembangan fitur manajemen hak akses pengguna. Scrum dipilih karena sesuai dengan karakteristik proyek ini yang bersifat iteratif dan membutuhkan penyesuaian bertahap berdasarkan kebutuhan pengguna yang berkembang selama proses pengembangan berlangsung. Dalam Scrum, pengembangan dibagi ke dalam siklus pendek yang disebut *Sprint*, di mana setiap *Sprint* menghasilkan bagian sistem yang dapat diuji dan dievaluasi (Atallah & Mardi, 2024). Adapun tahapan-tahapan yang dilakukan dalam model Scrum pada proyek ini dapat dilihat pada Gambar 1.



Gambar 1 Metode Agile

Penjelasan dari masing-masing tahapan pada Gambar 1 adalah sebagai berikut:

1. *Product Backlog*

   Tahap ini merupakan langkah awal dalam Scrum, yaitu mendefinisikan seluruh kebutuhan sistem secara menyeluruh. Kebutuhan dikumpulkan melalui observasi langsung dan wawancara dengan pihak terkait di PT. XYZ, menghasilkan daftar fitur yang diprioritaskan meliputi manajemen hak akses, alur persetujuan

berjenjang, dan pengelolaan pengguna berbasis RBAC (Dewi et al., 2024).

2. *Sprint Backlog*

   Sebelum setiap *Sprint* dimulai, dilakukan perencanaan untuk menentukan fitur mana yang akan dikerjakan dalam satu siklus pengembangan. Pada tahap ini juga ditentukan pembagian pekerjaan antara pengembangan front-end menggunakan Next.js dan back-end menggunakan NestJS (Dewi et al., 2024).

3. *Sprint*

   Sprint adalah periode dilakukanya pemaparan hasil produk yang akan disajikan kepada pihak terkait dalam bentuk *prototype*. Untuk peningkatan produk yang baru, sprint dianggap sebagai proyek dengan tujuan dan cakupan tertentu dimana semua fitur yang dimasukkan pada tahap ini telah diselesaikan, diuji, dan siap untuk diterapkan (Dewi et al., 2024).

4. *Sprint Review*

   Setelah setiap *Sprint* selesai, hasil pengembangan ditunjukkan kepada pembimbing magang dan stakeholder di PT. XYZ untuk mendapatkan umpan balik (Dewi et al., 2024).

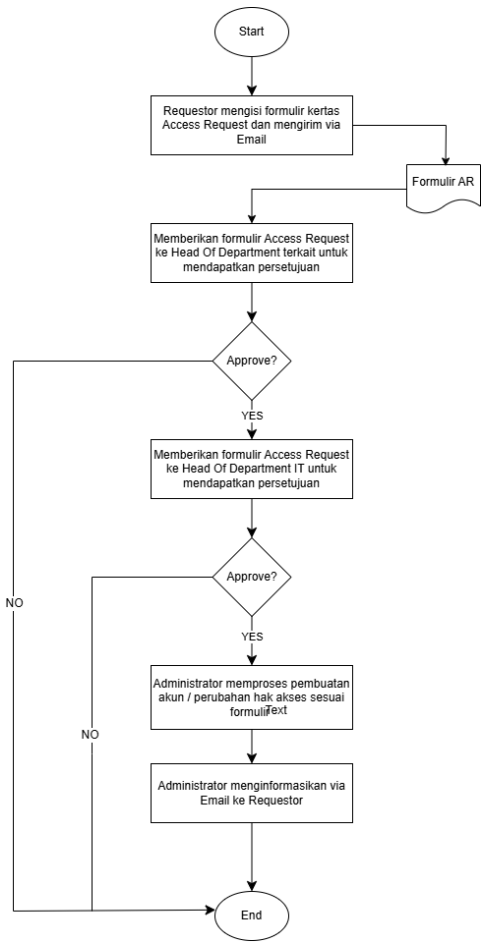
BAB III ANALISIS DAN PERANCANGAN

3.1 Analisis Kebutuhan

3.1.1 Proses Bisnis Berjalan

Saat ini, mekanisme pengelolaan hak akses pengguna di PT. XYZ masih dilakukan secara konvensional menggunakan formulir fisik (*paper form*). Selain itu, alur persetujuan (approval) yang berlapis juga sepenuhnya masih dikerjakan secara manual. Kondisi ini sering kali memicu berbagai kendala operasional, mulai dari lamanya waktu penyelesaian pengajuan akses, tingginya risiko kesalahan pencatatan (*human error*), hingga kemungkinan adanya pemberian hak akses yang melampaui batas wewenang pengguna (*over-privilege*).

Untuk memberikan gambaran lebih rinci mengenai proses bisnis yang sedang berjalan saat ini, tahapan pengajuan hak akses tersebut dapat dilihat pada Gambar 2.



Gambar 2 Proses Request Berjalan

Berdasarkan alur *flowchart* pada Gambar 2, proses dimulai ketika seorang Requestor yang membutuhkan pembuatan akun atau perubahan akses mengisi formulir fisik. Alur persetujuan secara berjenjang tersebut berjalan dengan skenario sebagai berikut:

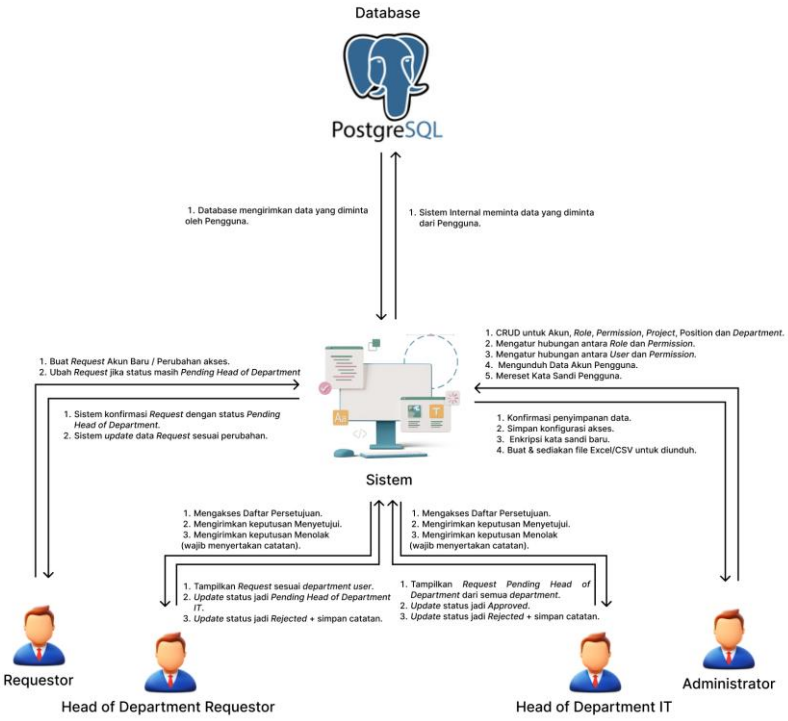
1. Formulir fisik yang telah diisi pertama-tama diserahkan kepada *Head of Department* terkait untuk diverifikasi kebenarannya. Pada tahap ini, jika pengajuan ditolak, maka formulir dikembalikan kepada *Requestor*. Namun jika disetujui, *Head of Department* akan memberikan tanda tangan persetujuan tahap pertama (*first approval*).
2. Berkas fisik yang sudah ditandatangani tersebut kemudian diteruskan secara manual kepada *Head of Department IT* guna divalidasi kembali dari sisi ketersediaan teknis dan kebijakan keamanan. Sama seperti tahap sebelumnya, jika tidak memenuhi syarat, pengajuan dapat ditolak dan dikembalikan.
3. Apabila pengajuan disetujui sepenuhnya, barulah berkas formulir fisik tersebut diserahkan kepada Administrator.
4. Administrator kemudian mengeksekusi pembuatan atau penyesuaian akun di dalam sistem dengan berpatokan pada data tersebut.

Seluruh tahapan operasional tersebut saat ini berjalan secara terpisah. Karena dokumen berpindah tangan secara fisik, proses ini sangat memakan waktu dan riwayat pengajuannya tidak terekam di dalam basis data digital. Oleh karena itu, perusahaan sangat membutuhkan implementasi sistem manajemen hak akses yang terpusat, terdokumentasi dengan baik, dan diatur secara ketat berdasarkan peran (*role*) masing-masing pengguna.

.

3.1.2 Gambaran Umum Sistem

Sistem manajemen hak akses berbasis web ini dirancang khusus untuk mendigitalisasi seluruh alur pengelolaan akun, mulai dari proses pengajuan (*request*), persetujuan berjenjang, hingga manajemen pengguna. Dalam operasionalnya, sistem ini diintegrasikan dengan basis data PostgreSQL yang bertugas menyimpan serta mengelola kredensial pengguna, dan konfigurasi *role*. Adapun arsitektur interaksi atau gambaran umum dari sistem ini dapat dilihat secara lebih rinci pada Gambar 3.



Gambar 3 Gambaran Umum

Berdasarkan arsitektur pada Gambar 3 sistem ini melibatkan empat aktor utama. Penetapan hak akses (*role*) untuk setiap pengguna dipetakan berdasarkan hierarki dan posisi fungsional karyawan di dalam struktur organisasi PT. XYZ. Rincian peran dan interaksi dari masing-masing aktor adalah sebagai berikut:

a. Requestor

Peran ini dijalankan oleh pengguna yang secara fungsional bertindak sebagai perwakilan (Admin Departemen)

yang membutuhkan akses. Requestor memiliki wewenang untuk menginisiasi dan mengajukan permintaan pembuatan akun baru atau perubahan hak akses operasional secara mandiri ke dalam sistem.

b. *Head of Department* Requestor

Peran ini ditetapkan secara spesifik kepada pengguna yang memiliki jabatan sebagai *Manager* atau *Head* pada departemen terkait. Penentuan ini merujuk pada struktur organisasi perusahaan, di mana pejabat tersebut memiliki otoritas untuk memberikan persetujuan level pertama (*first approval*) bagi staf di bawah departemennya.

c. *Head of Department* IT

Peran ini diberikan khusus kepada pengguna yang menjabat sebagai *Head of Department* pada divisi IT. Kriteria ini bersifat tunggal dan didasarkan pada tanggung jawab teknis untuk memberikan verifikasi akhir (*final approval*) sebelum akun dieksekusi.

d. Administrator

Peran *Administrator* (Admin IT) bertugas mengeksekusi sistem setelah seluruh alur persetujuan selesai. Selain itu, pembaruan kriteria *role* juga dikelola secara manual oleh Administrator apabila terdapat instruksi resmi dari departemen HRD/Kepegawaian mengenai mutasi, promosi, atau perubahan jabatan karyawan. Data posisi terbaru yang diterbitkan oleh HRD inilah yang menjadi landasan utama bagi Administrator untuk memperbarui *role* dan *permissions* pengguna di dalam sistem.

3.2 Perancangan

Bagian ini membahas secara menyeluruh mengenai spesifikasi arsitektur serta alur kerja dari fitur manajemen hak akses pengguna yang dibangun. Proses perancangan ini bertujuan untuk memberikan gambaran logis yang terstruktur mengenai bagaimana sistem akan mengotomatisasi seluruh proses bisnis di dalam perusahaan.

3.2.1 Kebutuhan Fungsional

Kebutuhan fungsional menjabarkan deskripsi mengenai layanan, fitur, maupun tindakan yang harus mampu dieksekusi oleh perangkat lunak guna memenuhi kebutuhan operasional pengguna. Dalam pengembangan sistem ini, kebutuhan fungsional dirancang secara spesifik untuk mengakomodasi siklus pengajuan (*request*), persetujuan berjenjang, hingga pengelolaan akun terpusat berbasis peran. Daftar lengkap kebutuhan fungsional perangkat lunak yang dikembangkan disajikan pada Tabel 3.1:

Tabel 3. 1 Kebutuhan Fungsional

No	Kebutuhan Fungsional
FR-01	Requestor, *Head of Department* Requestor, *Head of Department* dan Administrator dapat melakukan *login* sesuai dengan peran yang dimiliki.
FR-02	Requestor, *Head of Department* Requestor, *Head of Department* IT dan Administrator dapat melakukan *logout* dari sistem.
FR-03	Requestor dapat membuat *Request* akun baru atau perubahan hak akses.
FR-04	Requestor dapat merubah *Request* yang statusnya masih *Pending* dan belum disetujui oleh *Head of Department* Requestor.
FR-05	*Head of Department* Requestor dapat melihat detail *Request* yang diajukan *user* dalam *Department*nya.
FR-06	*Head of Department* Requestor dapat menolak dan menyetujui *Request* yang diajukan.
FR-07	*Head of Department* Requestor dapat memberikan catatan ketika menolak *Request*.
FR-08	*Head of Department* IT dapat melihat detail semua *Request* di seluruh *Department*.
FR-09	*Head of Department* IT dapat menolak dan menyetujui setiap *Request* yang diajukan.
FR-10	*Head of Department* IT dapat memberikan catatan ketika menolak *Request*.

FR-11	Administrator dapat mengelola Akun Pengguna.
FR-12	Administrator dapat mengelola *Role*.
FR-13	Administrator dapat mengelola *Project*.
FR-14	Administrator dapat mengelola *Permission*.
FR-15	Administrator dapat mengelola *Department*.
FR-16	Administrator dapat mengelola *Position*.
FR-17	Administrator dapat mengirimkan tautan reset kata sandi pengguna.
FR-18	Administrator dapat unduh Akun Pengguna.

3.2.2 Kebutuhan Non-Fungsional

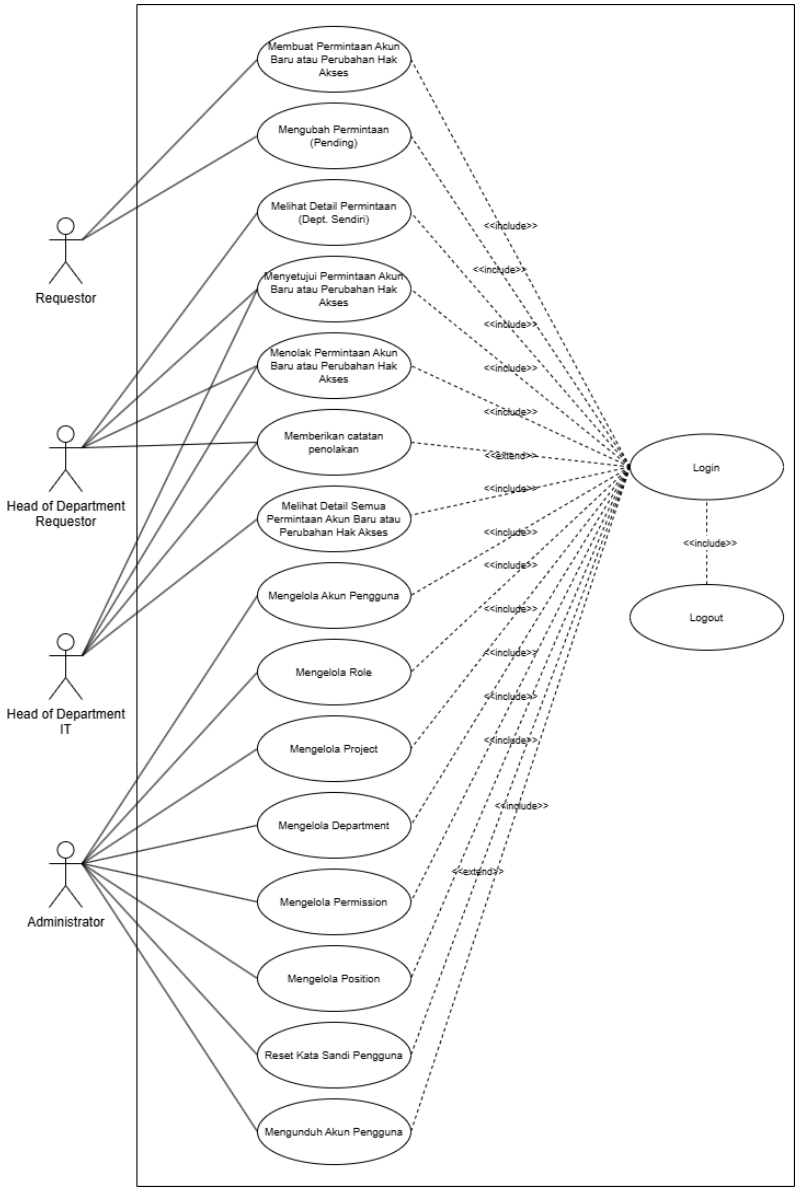
Kebutuhan non-fungsional merupakan penjelasan tentang atribut-atribut sistem, yang tidak berkaitan dengan fungsi sistem secara langsung. Dalam kebutuhan non-fungsional menjelaskan bagaimana sistem harus beroperasi, bukan apa yang harus dilakukan. Adapun rincian kebutuhan non-fungsional dari sistem manajemen hak akses yang dikembangkan dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Kebutuhan Non-Fungsional

No	Kebutuhan Non-Fungsional
NFR-01	Sistem membatasi hak akses pengguna secara legal dan aman menggunakan metode *Role-Based Access Control* (RBAC) untuk melindungi data perusahaan.
NFR-02	Antarmuka sistem didesain agar mudah dipahami, sehingga aman dan nyaman dioperasikan oleh berbagai tingkat pengguna.
NFR-03	Sistem dapat beroperasi dengan responsif dan stabil, baik saat diakses melalui *website* maupun perangkat *mobile*.
NFR-04	Sistem menggunakan Bahasa Inggris sebagai standar komunikasi profesional yang etis di dalam lingkungan perusahaan.
NFR-05	Sistem menjamin transparansi proses melalui rekam jejak aktivitas (*audit log*) untuk mencegah tindakan tidak etis.
NFR-06	Sistem mendukung keberlanjutan lingkungan melalui proses kerja *paperless*.

3.2.3 Diagram Use Case

Use Case Diagram merupakan salah satu diagram dari UML *(Unified Modeling Language)* yang menggambarkan interaksi antara aktor atau pengguna dengan fungsi-fungsi yang ada pada sistem yang akan dibuat. Interaksi tersebut dapat digambarkan sebagai berikut pada Gambar 4.



Gambar 4 Use Case Diagram

3.2.4 Skenario *Use Case*

Skenario *Use Case Diagram* merupakan rangkaian dekripsi alur kerja yang terjadi selama interaksi antara aktor dengan sistem berdasarkan *Use Case* tertentu. Dalam scenario *Use Case Diagram* dari Manajemen hak akses Pengguna dengan metode *Role-Based Access Control* (RBAC), dapat dijelaskan sebagai berikut:

Tabel 3.3 menjelaskan proses bagi semua pengguna untuk masuk kedalam sistem menggunakan *username* dan kata sandi yang valid. Jika berhasil, pengguna akan diarahkan ke halaman utama sesuai dengan hak aksesnya. Jika gagal, sistem akan menampilkan pesan kesalahan.

Tabel 3.3 Use Case Scenario Login

Skenario *Use Case* 1: *Login*	
Nomor	UC-01
Nama	*Login*
Aktor	Pengguna dengan semua *Role* (Requestor, *Head of Department* Requestor, *Head of Department* IT, Administrator)
Tujuan	Pengguna masuk ke dalam sistem internal perusahaan
Kondisi Awal	1. Pengguna memiliki *username* yang sudah terdaftar di sistem 2. Pengguna memiliki kata sandi yang valid.
Kondisi Akhir	1. Pengguna berhasil *login* ke akun sesuai *Role* masing-masing. 2. Pengguna dapat masuk ke halaman *dashboard* sesuai hak aksesnya.
Skenario Utama	1. Pengguna membuka halaman *login* sistem internal perusahaan. 2. Pengguna mengisi *username* dan kata sandi dengan benar.

	3. Pengguna berhasil *Iogin.*
Skenario Alternatif	1. Jika data belum terdaftar maka *login* gagal. 2. Jika *username* atau kata sandi salah maka sistem menampilkan pesan "*Invalid username or password*".

Tabel 3.4 memungkinkan pengguna yang sedang aktif untuk keluar secara aman dari sistem, proses ini akan mengakhiri sesi pengguna dan mengembalikannya ke halam login.

Tabel 3.4 Use Case Scenario Logout

Skenario *Use Case* 2: *Logout*	
Nomor	UC-02
Nama	*Logout*
Aktor	Pengguna dengan semua *Role* (Requestor, *Head of Department* Requestor, *Head of Department* IT, Administrator)
Tujuan	Pengguna keluar dari sistem internal perusahaan
Kondisi Awal	1. Pengguna sudah terdaftar dan memiliki akun sah. 2. Pengguna sedang *login* ke sistem.
Kondisi Akhir	1. Pengguna berhasil keluar dari sistem.
Skenario Utama	1. Pengguna klik menu *Logout*. 2. Sistem menghapus sesi *Login.* 3. Pengguna diarahkan ke halaman *login*.
Skenario Alternatif	1. Jika sistem meminta konfirmasi *logout* dan pengguna membatalkannya, maka sesi tetap aktif.

Tabel 3.5 ini menggambarkan proses Requestor mengajukan pembuatan akun baru atau perubahan hak akses, lalu sistem menyimpan *Request* dengan status "*Pending Dept Head Approval*".

Tabel 3.5 Use Case Skenario Requestor Membuat Request

Use Case* Skenario *Use Case* 3: Membuat *Request	
Nomor	UC-03
Nama	Membuat *Request* Akun Baru atau Perubahan Hak Akses
Aktor	Requestor
Tujuan	Requestor mengajukan *Request* pembuatan akun baru atau perubahan hak akses.
Kondisi Awal	Requestor sudah berhasil *Login* ke dalam sistem.
Kondisi Akhir	*Request* tersimpan di sistem dengan status "*Pending Dept Head Approval*".
Skenario Utama	1. Requestor memilih menu "*Create User Request*". 2. Pengguna mengisi detail data akun yang diminta. 3. Pengguna menekan tombol "*Submit Request*". 4. Sistem menyimpan *Request* dengan status menunggu "*Pending Dept Head Approval*".
Skenario Alternatif	1. Jika data wajib (seperti Proyek atau *Group*/*Role*) tidak terisi. Sistem akan menampilkan pesan peringatan: "*Please fill in all required fields.*" 2. Jika Requestor mengonfirmasi pembatalan *Request* tidak disimpan,

	dan sistem kembali ke halaman utama.

Tabel 3.6 menggambarkan bagaimana Requestor dapat memodifikasi detail *Request*nya sebelum *Request* tersebut disetujui oleh *Head of Department* Requestor.

Tabel 3.6 Use Case Skenario Requestor Mengubah Request

Skenario *Use Case* 4: Mengubah *Request*	
Nomor	UC-04
Nama	Mengubah *Request* (Status *Pending*)
Aktor	Requestor
Tujuan	Requestor merubah *Request* pembuatan akun baru atau perubahan hak akses.
Kondisi Awal	Requestor sudah berhasil *Login* ke dalam sistem.
Kondisi Akhir	*Request* yang dipilih berhasil diperbarui di sistem dengan status tetap "*Pending*".
Skenario Utama	1. Requestor memilih menu "*Request List*". 2. Requestor mencari dan memilih *Request* yang ingin diubah. 3. Requestor mengklik tombol "*Edit*". 4. Requestor melakukan perubahan data yang diinginkan (misalnya, mengubah *Department*). 5. Requestor menekan tombol "*Update*" 6. Sistem menampilkan pesan "*The data has been updated successfully.*"
Skenario Alternatif	1. Jika data wajib (setelah diubah) terdeteksi tidak lengkap. Sistem

	akan menampilkan pesan peringatan: "*Please fill out this field"*

Tabel 3.7 menggambarkan proses *Head of Department* Requestor meninjau detail *Request* yang berasal dari *Department* yang sama dengan *Head of Department* tersebut, sebelum memberikan keputusan. Sistem menampilkan informasi lengkap dari *Request* yang dipilih.

Tabel 3.7 Use case skenario Head of Department Requestor Melihat Detail Request

Skenario *Use Case* 5: Melihat Detail *Request*	
Nomor	UC-05
Nama	*Head of Department* Requestor Melihat Detail *Request* (Sesuai *Department*)
Aktor	*Head of Department* Requestor
Tujuan	*Head of Department* Requestor dapat meninjau detail *Request* yang diajukan oleh Requestor dari *Department* yang sama, sebelum memberikan keputusan persetujuan/penolakan.
Kondisi Awal	*Head of Department* Requestor sudah *login* ke sistem.
Kondisi Akhir	Detail *Request* yang valid dan relevan tampil di sistem.
Skenario Utama	1. *Head of Department* Requestor memilih menu "*List Approvals*". 2. *Head of Department* Requestor mengklik tombol "*Details*" pada salah satu *Request* dari daftar. 3. Sistem menampilkan detail lengkap *Request* tersebut.

Skenario Alternatif	Jika *Request* tidak ditemukan, sistem menampilkan pesan *error*.

Tabel 3.8 menjelaskan proses *Head of Department* Requestor dalam menyetujui atau menolak *Request* yang diajukan oleh Requestor. Sistem akan memperbarui status *Request* menjadi "*Approved*", "*Rejected*", dan "*Pending*" sesuai keputusan yang diberikan.

Tabel 3.8 Use Case Skenario Head of Department Requestor Mengambil Keputusan

Skenario *Use Case* 6: Mengambil Keputusan	
Nomor	UC-06
Nama	*Head of Department* Requestor Mengambil Keputusan Persetujuan (Menyetujui / Menolak)
Aktor	*Head of Department* Requestor
Tujuan	*Head of Department* Requestor berhasil memberikan keputusan (Menyetujui atau Menolak) atas *Request* yang relevan dengan *Department* nya.
Kondisi Awal	1. *Head of Department* Requestor sudah *login* ke sistem. 2. *Head of Department* Requestor meninjau detail *Request* (Status *Pending*) dari *Department* nya.
Kondisi Akhir	*Request* berstatus "*Pending IT Head Approval*" dan siap diteruskan ke *Head of Department* IT, atau *Request* berstatus "*Rejected*".
Skenario Utama	1. *Head of Department* Requestor sudah *login* ke sistem.

	2. *Head of Department* Requestor mengakses dan membuka detail *Request*. 3. *Head of Department* Requestor berada di halaman detail *Request* yang akan diputuskan. 4. *Head of Department* Requestor memilih opsi "*Approved*". 5. Sistem menampilkan pesan konfirmasi persetujuan.
Skenario Alternatif	1. Jika *Head of Department* Requestor memilih opsi Tolak, maka status *Request* diperbarui menjadi "*Rejected*". 2. *Head of Department* Requestor dapat menambahkan catatan atau alasan penolakan.

Tabel 3.9 ini menggambarkan proses *Head of Department* Requestor dalam memberikan alasan penolakan terhadap *Request* yang tidak disetujui. Sistem menyimpan catatan tersebut dan menautkannya pada data *Request* yang ditolak.

Tabel 3.9 Use Case Skenario Head of Department Requestor Memberikan Catatan Penolakan

Skenario *Use Case* 7: Memberikan Catatan Penolakan	
Nomor	UC-07
Nama	*Head of Department* Requestor Memberikan Catatan Penolakan
Aktor	*Head of Department* Requestor
Tujuan	*Head of Department* Requestor dapat memberikan alasan penolakan pada *Request* yang tidak disetujui.

Kondisi Awal	1. *Head of Department* Requestor sudah *login* ke sistem. 2. *Head of Department* Requestor telah mengklik tombol "*Rejected*" pada halaman Detail *Request*.
Kondisi Akhir	Alasan penolakan tersimpan di kolom deskripsi/catatan.
Skenario Utama	1. *Head of Department* Requestor menolak *Request* yang diajukan. 2. Sistem menampilkan kolom input catatan alasan. 3. *Head of Department* Requestor mengisi catatan alasan penolakan. 4. *Head of Department* Requestor menekan tombol "*Save*". 5. Sistem menyimpan catatan alasan dan menautkannya pada *Request* yang ditolak.
Skenario Alternatif	Jika kolom catatan dibiarkan kosong, sistem menampilkan pesan peringatan (*alert*) bahwa alasan wajib diisi.

Tabel 3.10 menjelaskan proses *Head of Department* IT meninjau detail *Request* yang telah disetujui oleh *Head of Department* Requestor sebelum memberikan keputusan akhir. Sistem menampilkan informasi lengkap dari *Request* yang dipilih untuk memastikan kelayakan sebelum disetujui atau ditolak.

Tabel 3.10 Use Case Skenario Head of Department IT Melihat Detail

Skenario *Use Case* 8: Melihat Detail *Request*	
Nomor	UC-08
Nama	*Head of Department* IT Melihat Detail *Request*
Aktor	*Head of Department* IT

Tujuan	*Head of Department* IT dapat meninjau detail lengkap semua *Request* yang telah disetujui *Head of Department* Requestor.
Kondisi Awal	1. *Head of Department* IT sudah berhasil "*Login*" ke sistem. 2. Terdapat *Request* yang memiliki status "*Pending IT Head Approval*".
Kondisi Akhir	Detail *Request* tampil di sistem.
Skenario Utama	1. *Head of Department* IT memilih menu "*Pending IT Mgr/Asst. IT Mgr*". 2. *Head of Department* IT mengklik tombol "*Details*" pada *Request* yang ingin ditinjau. 3. Sistem menampilkan detail lengkap *Request* tersebut.
Skenario Alternatif	Jika *Request* tidak ditemukan, sistem menampilkan pesan *error*.

Tabel 3.11 menjelaskan proses *Head of Department* IT dalam menyetujui atau menolak *Request* yang telah disetujui oleh *Head of Department* Requestor. Sistem akan memperbarui status *Request* menjadi "*Approved*", "*Pending*" atau "*Rejected*" sesuai keputusan yang diberikan.

Tabel 3.11 Use Case Skenario Head of Department IT Menyetujui / Menolak Request

Skenario *Use Case* 9: Mengambil Keputusan	
Nomor	UC-09
Nama	*Head of Department* IT Mengambil Keputusan Persetujuan (Menyetujui / Menolak)
Aktor	*Head of Department* IT

Tujuan	*Head of Department* IT berhasil memberikan keputusan akhir (Menyetujui atau Menolak) atas *Request* yang telah disetujui *Head of Department* Requestor.
Kondisi Awal	1. *Head of Department* IT sudah berhasil *Login* ke sistem. 2. *Head of Department* IT sedang melihat halaman Detail *Request* yang memiliki status "*Pending IT Head Approval*"
Kondisi Akhir	*Request* berstatus "*Approved*" atau *Request* berstatus "*Rejected*".
Skenario Utama	1. *Head of Department* IT sudah *login* ke sistem. 2. *Head of Department* IT mengakses dan membuka detail *Request*. 3. *Head of Department* IT berada di halaman detail *Request* yang akan diputuskan. 4. *Head of Department* IT memilih opsi "*Approved*". 5. Sistem menampilkan pesan konfirmasi persetujuan.
Skenario Alternatif	1. Jika *Head of Department* IT memilih opsi Tolak, maka status *Request* diperbarui menjadi "*Rejected*". 2. *Head of Department* IT dapat menambahkan catatan atau alasan penolakan.

Tabel 3.12 menggambarkan proses *Head of Department* IT dalam memberikan alasan penolakan terhadap *Request* yang tidak

disetujui. Sistem menyimpan catatan tersebut dan menautkannya pada data *Request* yang ditolak.

Tabel 3.12 Use Case Skenario Head of Department IT Memberikan Catatan Penolakan

Skenario *Use Case* 10: Memberikan Catatan Penolakan	
Nomor	UC-010
Nama	*Head of Department* IT Memberikan Catatan Penolakan
Aktor	*Head of Department* IT
Tujuan	*Head of Department* IT dapat memberikan alasan penolakan pada *Request* yang tidak disetujui.
Kondisi Awal	1. *Head of Department* IT sudah *login* ke sistem. 2. *Head of Department* IT telah mengklik tombol "*Rejected*" pada halaman Detail *Request*.
Kondisi Akhir	Alasan penolakan tersimpan di kolom deskripsi/catatan.
Skenario Utama	1. *Head of Department* IT menolak *Request* yang diajukan. 2. Sistem menampilkan kolom input catatan alasan. 3. *Head of Department* IT mengisi catatan alasan penolakan. 4. *Head of Department* IT menekan tombol "*Save*". 5. Sistem menyimpan catatan alasan dan menautkannya pada *Request* yang ditolak.
Skenario Alternatif	Jika kolom catatan dibiarkan kosong, sistem menampilkan pesan peringatan (*alert*) bahwa alasan wajib diisi.

Tabel 3.13 menjelaskan proses Administrator dalam mengelola akun pengguna, meliputi penambahan, perubahan, dan penghapusan akun. Sistem akan menyimpan perubahan yang dilakukan, menampilkan pesan *error* jika data tidak lengkap, dan memastikan data akun selalu sesuai dengan kebutuhan akses pengguna.

Tabel 3.13 Use Case Skenario Administrator Mengelola Akun Pengguna

Skenario *Use Case* 11: Mengelola Akun Pengguna	
Nomor	UC-011
Nama	Mengelola Akun Pengguna
Aktor	Administrator
Tujuan	Administrator dapat menambah, mengubah, dan menghapus akun pengguna dalam sistem.
Kondisi Awal	1. Administrator sudah *login* ke sistem. 2. Administrator berada di halaman Daftar Akun Pengguna. 3. *Role* yang akan ditetapkan sudah tersedia dalam sistem.
Kondisi Akhir	Akun pengguna berhasil dibuat, diperbarui, atau dihapus dari sistem.
Skenario Utama	1. Administrator *login* ke dalam sistem. 2. Administrator membuka halaman daftar akun pengguna 3. Administrator menekan tombol Buat Akun pada *Request* tersebut. 4. Administrator mengisi detail kebutuhan pengguna termasuk *position*/jabatan sebagai dasar penetapan *role*.

	5. Administrator memilih *Role* yang akan ditetapkan. 6. Administrator menekan simpan, sistem membuat dan menyimpan akun baru. 7. Administrator memilih akun dari daftar, lalu menekan Edit. 8. Administrator melakukan perubahan. 9. Administrator menekan Simpan Perubahan, sistem memperbarui data akun. 10. Administrator memilih akun, lalu menekan hapus. 11. Sistem meminta konfirmasi. 12. Administrator mengonfirmasi, sistem menghapus akun tersebut.
Skenario Alternatif	Jika data wajib tidak lengkap saat penambahan/perubahan, sistem menampilkan pesan *error* dan menolak menyimpan.

Tabel 3.14 menjelaskan proses Administrator dalam mengelola *Role*, termasuk menambah, mengubah, dan menghapus *Role* serta mengatur hak akses yang terkait. Sistem menyimpan setiap perubahan yang dilakukan dan menampilkan pesan *error* jika data yang dimasukkan tidak valid.

Tabel 3.14 Use Case Skenario Administrator Mengelola Role

Skenario *Use Case* 12: Mengelola *Role*	
Nomor	UC-012
Nama	Mengelola *Role*
Aktor	Administrator

Tujuan	Administrator dapat membuat, memperbarui, dan menghapus *Role* serta mengelola izin yang terkait dengan *Role* tersebut.
Kondisi Awal	1. Administrator sudah *login* ke sistem. 2. Administrator berada di halaman Manajemen *Role*.
Kondisi Akhir	*Role* baru berhasil dibuat, diperbarui, atau dihapus dari sistem.
Skenario Utama	1. Administrator *login* ke dalam sistem. 2. Administrator memilih tombol "*Add Role*". 3. Sistem menampilkan *form*. 4. Administrator mengisi Nama *Role*. 5. Administrator memilih izin yang akan ditetapkan pada *Role* ini. 6. Administrator menekan "*Save*". 7. Untuk mengubah, Administrator memilih *Role* dari daftar, menekan "Edit", mengubah data (termasuk daftar izin), dan menekan "*Edit*".
Skenario Alternatif	1. Administrator memilih *Role* dari daftar dan menekan "*Delete*". Sistem menampilkan dialog konfirmasi penghapusan. 2. Jika data wajib (misalnya, Nama *Role*) dikosongkan atau sudah ada (duplikasi) saat penyimpanan. Sistem menampilkan pesan "*Role already exists"*.

Tabel 3.15 menjelaskan proses Administrator mengelola data *Project*. Administratoristrasi ini memastikan setiap *Project* terhubung wajib ke satu *Department* penanggung jawab.

Tabel 3.15 Use Case Skenario Administrator Mengelola Project

Skenario *Use Case* 13: Mengelola *Project*	
Nomor	UC-013
Nama	Mengelola *Project*
Aktor	Administrator
Tujuan	Administrator berhasil mengelola data *project* dan terhubung ke *Department* penanggung jawab.
Kondisi Awal	1. Administrator sudah *login* ke sistem. 2. Administrator berada di halaman Manajemen *Project*.
Kondisi Akhir	*Project* baru berhasil disimpan dan terhubung ke *Department* yang valid.
Skenario Utama	1. Administrator *login* ke dalam sistem. 2. Administrator memilih tombol "*Add Project*". 3. Sistem menampilkan *form*. 4. Administrator mengisi Nama *Project* dan memilih *Department* penanggung jawab. 5. Administrator menekan "*Save*". 6. Untuk mengubah, Administrator memilih *Project*, mengubah nama, dan "*Edit*". 7. Untuk Hapus, sistem mengonfirmasi penghapusan.

Skenario Alternatif	Jika Nama *Project* sudah ada (duplikasi), sistem menolak dan menampilkan pesan "*Project already exists.*"

Tabel 3.16 menjelaskan proses Administrator dalam mengelola data master *Permission*, meliputi penambahan *Permission* baru, perubahan nama, dan penghapusan *Permission*.

Tabel 3.16 Use Case Skenario Administrator Mengelola Permission

Skenario *Use Case* 14: Mengelola *Permission*	
Nomor	UC-014
Nama	Mengelola *Permission*
Aktor	Administrator
Tujuan	Administrator berhasil mengelola struktur *Permission* di sistem.
Kondisi Awal	1. Administrator sudah *login* ke sistem. 2. Administrator berada di halaman Manajemen *Permission*.
Kondisi Akhir	*Permission* baru berhasil disimpan.
Skenario Utama	1. Administrator *login* ke dalam sistem. 2. Administrator memilih tombol "*Add Permission* ". 3. Sistem menampilkan form. 4. Administrator Mengisi Nama *Permission*. 5. Administrator menekan "*Save*". 6. Untuk mengubah, Administrator memilih *Permission*, mengubah nama, dan "*Edit*". 7. Untuk Hapus, sistem mengonfirmasi penghapusan.

Skenario Alternatif	Jika Nama *Permission* duplikat, sistem menampilkan pesan *"Permission with name already exists."*

Tabel 3.17 menjelaskan proses Administrator dalam mengelola data master *Department*, meliputi penambahan *Department* baru, perubahan nama, dan penghapusan *Department*.

Tabel 3.17 Use Case Skenario Administrator Mengelola Department

Skenario *Use Case* 15: Mengelola *Department*	
Nomor	UC-015
Nama	Mengelola *Department*
Aktor	Administrator
Tujuan	Administrator berhasil mengelola struktur *Department* di sistem.
Kondisi Awal	1. Administrator sudah *login* ke sistem. 2. Administrator berada di halaman Manajemen *Department*.
Kondisi Akhir	Data *Department* berhasil disimpan, diperbarui, atau dihapus dari database sistem.
Skenario Utama	1. Administrator *login* ke dalam sistem. 2. Administrator memilih tombol "*Add Department*". 3. Sistem menampilkan *form*. 4. Administrator Mengisi Nama *Department*. 5. Administrator menekan "*Save*". 6. Untuk mengubah, Administrator memilih *Department*, mengubah nama, dan "*Edit*".

	7. Untuk Hapus, sistem mengonfirmasi penghapusan.
Skenario Alternatif	Jika Nama *Department* duplikat, sistem menampilkan pesan "*Department already exists*."

Tabel 3.18 menjelaskan proses Administrator dalam mengelola data master *Position*, meliputi penambahan *Position* baru, perubahan nama, dan penghapusan *Position*.

Tabel 3.18 Use Case Skenario Administrator Mengelola Position

Skenario *Use Case* 16: Mengelola *Position*	
Nomor	UC-016
Nama	Mengelola *Position* & Penetapan *Role*
Aktor	Administrator
Tujuan	Administrator berhasil mengelola data posisi dan memetakan peran (*role mapping*) pengguna.
Kondisi Awal	1. Administrator sudah *login* ke sistem. 2. Administrator berada di halaman Manajemen *Position*.
Kondisi Akhir	Data posisi beserta relasi peran (*role*) berhasil disimpan, diperbarui, atau dihapus.
Skenario Utama	1. Administrator mengakses navigasi *User Management* dan memilih *Position List*. 2. Administrator menekan tombol "*Add Position*". 3. Sistem menampilkan formulir pengisian data posisi.

	4. Administrator menginputkan Nama *Position* dan memilih *Role Mapping* dari daftar yang tersedia. 5. Administrator menekan tombol "*Save*". 6. Sistem melakukan validasi dan menyimpan data. 7. Untuk *Update*, Administrator menekan ikon Edit, memperbarui data, lalu menyimpan kembali. 8. Untuk *Delete*, Administrator menekan ikon *Trash* dan melakukan konfirmasi pada pop-up dialog.
Skenario Alternatif	Jika nama posisi yang diinputkan sudah terdaftar, sistem menampilkan pesan: "*The position name already exists*.

Tabel 3.19 menjelaskan proses Administrator mengatur ulang kata sandi pengguna secara langsung. Fungsi ini memungkinkan Administrator untuk membuat dan menetapkan kata sandi sementara yang baru kepada pengguna

Tabel 3.19 Use Case Skenario Administrator Mengatur Ulang Kata Sandi Pengguna

Skenario Use Case 17: Mengirim Tautan Atur Ulang Kata Sandi	
Nomor	UC-017
Nama	Mengirim Tautan Atur Ulang Kata Sandi
Aktor	Administrator
Tujuan	Administrator berhasil mengirimkan tautan (*link*) atur ulang kata sandi ke email pengguna.

Kondisi Awal	1. Administrator sudah *login* ke si stem. 2. Administrator berada di halaman Manajemen Akun Pengguna.
Kondisi Akhir	Tautan atur ulang kata sandi berhasil dikirim ke email pengguna terkait.
Skenario Utama	1. Administrator masuk ke halaman daftar akun pengguna. 2. Administrator menekan tombol "*Reset Password*" pada salah satu akun pengguna. 3. Sistem menampilkan konfirmasi (pop-up) untuk memastikan. 4. Administrator menekan tombol konfirmasi ("*Yes, Reset!")*. 5. 5Sistem memproses permintaan, lalu mengirimkan tautan reset ke email pengguna. 6. Sistem menampilkan pesan sukses (*Success*!) bahwa tautan telah terkirim.
Skenario Alternatif	Jika terjadi kegagalan pada server dalam mengirim email, sistem akan membatalkan proses dan menampilkan pesan error ("*Failed to reset password*").

Tabel 3.20 menjelaskan proses Administrator dalam mengunduh daftar akun pengguna ke dalam file *Excel*. Sistem menghasilkan file yang memuat data akun pengguna aktif dan menampilkan pesan jika tidak ada data yang dapat diunduh.

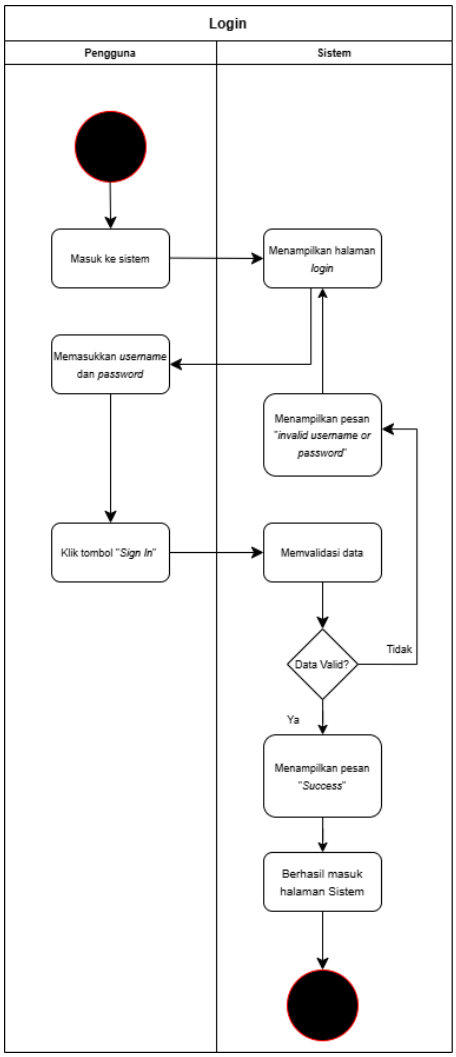
Tabel 3.20 Use Case Skenario Administrator Unduh Akun Pengguna

Skenario *Use Case* 18: Unduh Akun Pengguna	
Nomor	UC-018
Nama	Unduh Daftar Akun Pengguna
Aktor	Administrator
Tujuan	Administrator dapat mengunduh daftar akun pengguna ke dalam file *Excel*.
Kondisi Awal	1. Administrator sudah *login* ke sistem. 2. Terdapat akun pengguna dalam sistem.
Kondisi Akhir	File *Excel* yang berisi daftar akun pengguna berhasil diunduh.
Skenario Utama	1. Administrator *login* ke dalam sistem. 2. Administrator menekan tombol "*Export Excel*". 3. Sistem menghasilkan file *Excel* yang memuat daftar akun pengguna. 4. File *Excel* tersedia untuk diunduh oleh Administrator.
Skenario Alternatif	Jika tidak ada akun pengguna dalam sistem, maka sistem menampilkan pesan "*There is no data to download*." dan tidak menghasilkan file *Excel*.

3.2.5 Activity diagram

Activity diagram berfungsi untuk memodelkan alur kerja (*workflow*) atau aktivitas sekuensial dari sebuah sistem. Pada perancangan sistem manajemen hak akses ini, terdapat beberapa *activity* diagram yang merepresentasikan setiap proses bisnis utama.

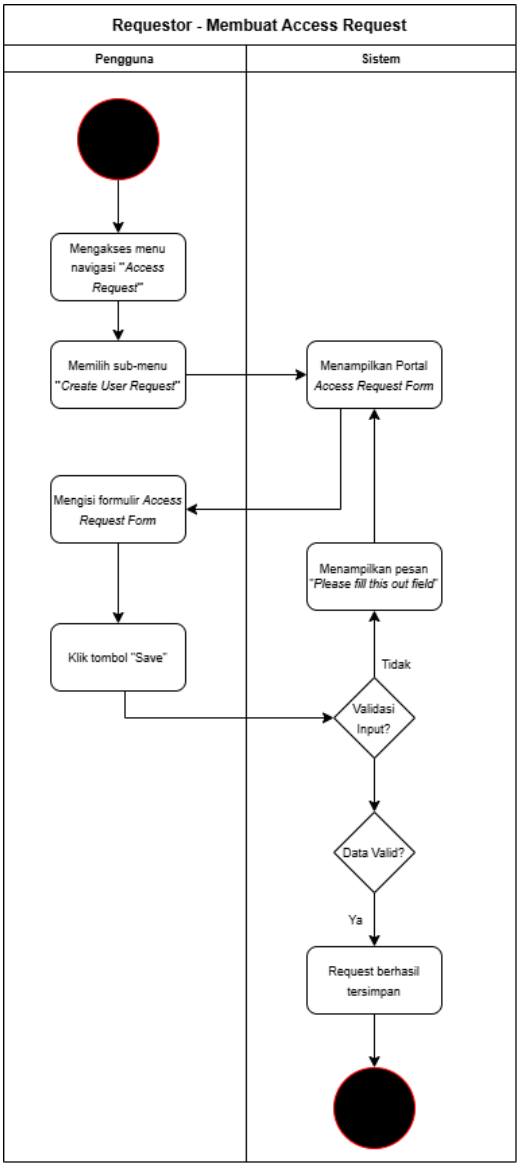
a. Activity Diagram Login



Gambar 5 Activity Diagram Login

Berdasarkan alur pada Gambar 5, mekanisme login pengguna, mulai dari pengisian *username* dan *password* hingga tahap validasi data. Jika data valid, pengguna akan diarahkan ke halaman utama; jika tidak, sistem akan meminta pengguna untuk mencoba kembali.

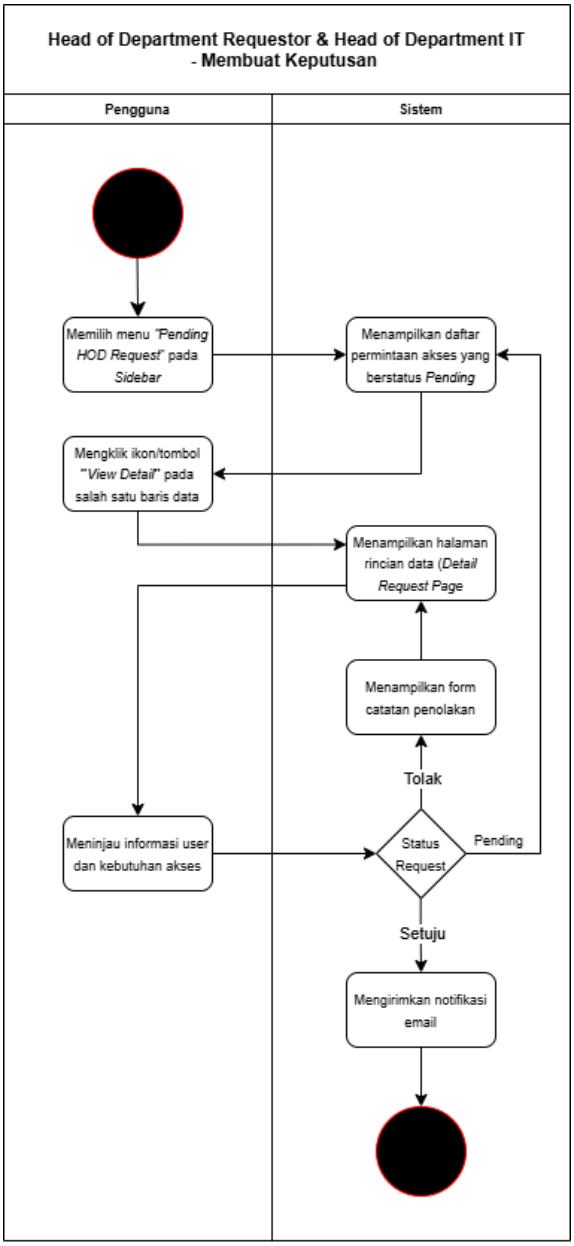
b. Activity Diagram Requestor Membuat Request



Gambar 6 Activity Diagram Requestor Membuat Request

Berdasarkan alur pada Gambar 6, mengilustrasikan alur kerja pengguna dalam melakukan permintaan akses akun. Proses dimulai saat pengguna mengakses menu navigasi *"Access Request*" dan memilih sub-menu *"Create User Request*" mengisi *form Request*, kemudian menyimpannya. Sistem akan memvalidasi data yang dimasukkan, dan jika valid *Request* akan berhasil tersimpan.

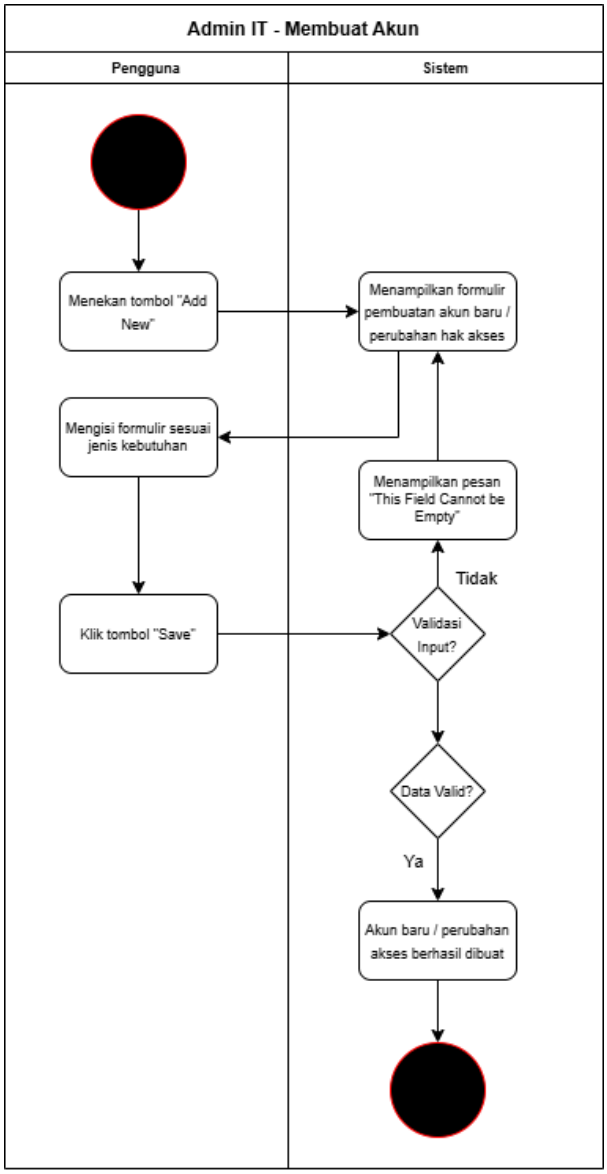
c. *Activity Diagram Head of Department* Membuat Keputusan



Gambar 7 Activity Diagram Requestor Membuat Request

Berdasarkan alur pada Gambar 7, pengambilan keputusan oleh *Head of Department* Requestor dan *Head of Department* IT, dimulai dari pemilihan *Request*, peninjauan detail *form*, hingga keputusan berupa *Approve, Pending,* atau *Rejected.* Sistem otomatis memperbarui status setiap keputusan untuk memastikan transparansi, akuntabilitas, dan efisiensi proses approval.

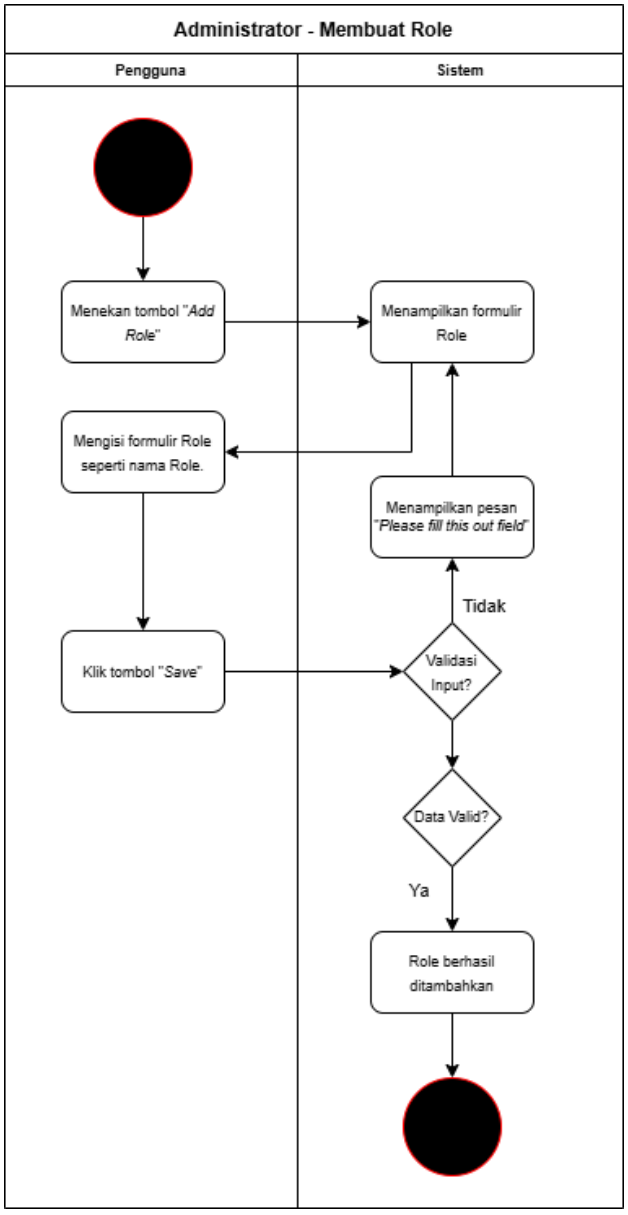
d. Activity Diagram Administrator Membuat Akun



Gambar 8 Activity Diagram Administrator Membuat Akun

Berdasarkan alur pada Gambar 8. proses pembuatan akun baru oleh Administrator, dimulai dari membuka daftar akun, menekan tombol "*Add New*", mengisi form pembuatan akun, dan menyimpannya. Sistem kemudian memvalidasi data, jika valid akun baru berhasil dibuat, sedangkan jika tidak valid sistem menampilkan pesan gagal menambah akun baru.

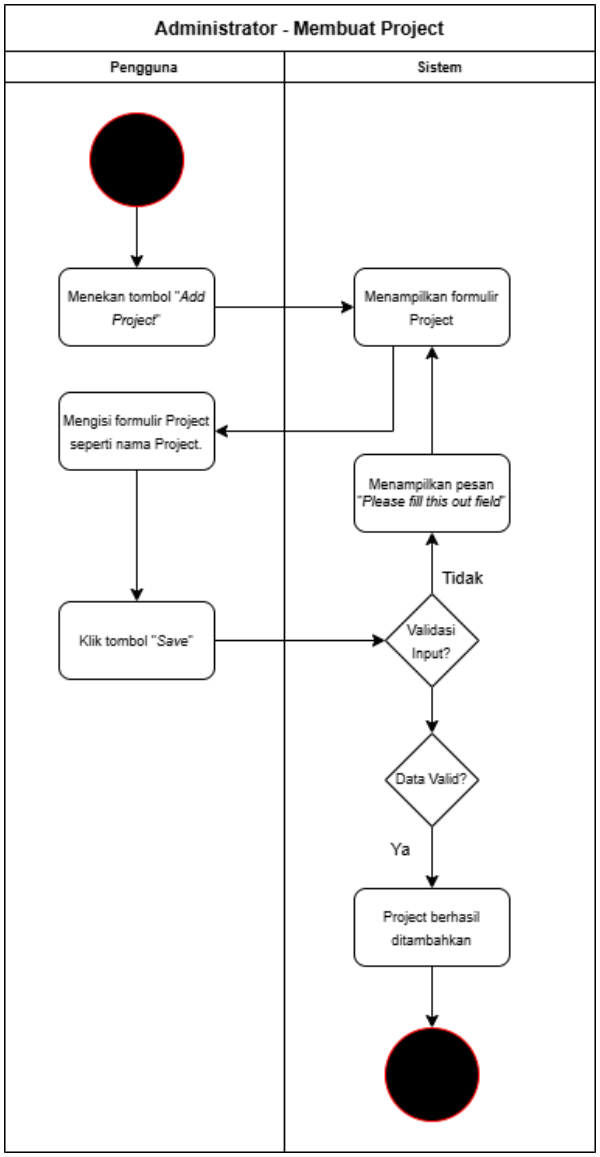
e. *Activity Diagram* Administrator Membuat *Role*



Gambar 9 Activity Diagram Administrator Membuat Role

Berdasarkan alur pada Gambar 9, proses pembuatan *Role* baru oleh Administrator, dimulai dari menekan tombol "*Add Role*", mengisi form pembuatan *Role*, dan menyimpannya. Sistem kemudian memvalidasi data, jika valid *Role* baru berhasil dibuat, sedangkan jika tidak valid sistem menampilkan pesan gagal menambah *Role* baru.

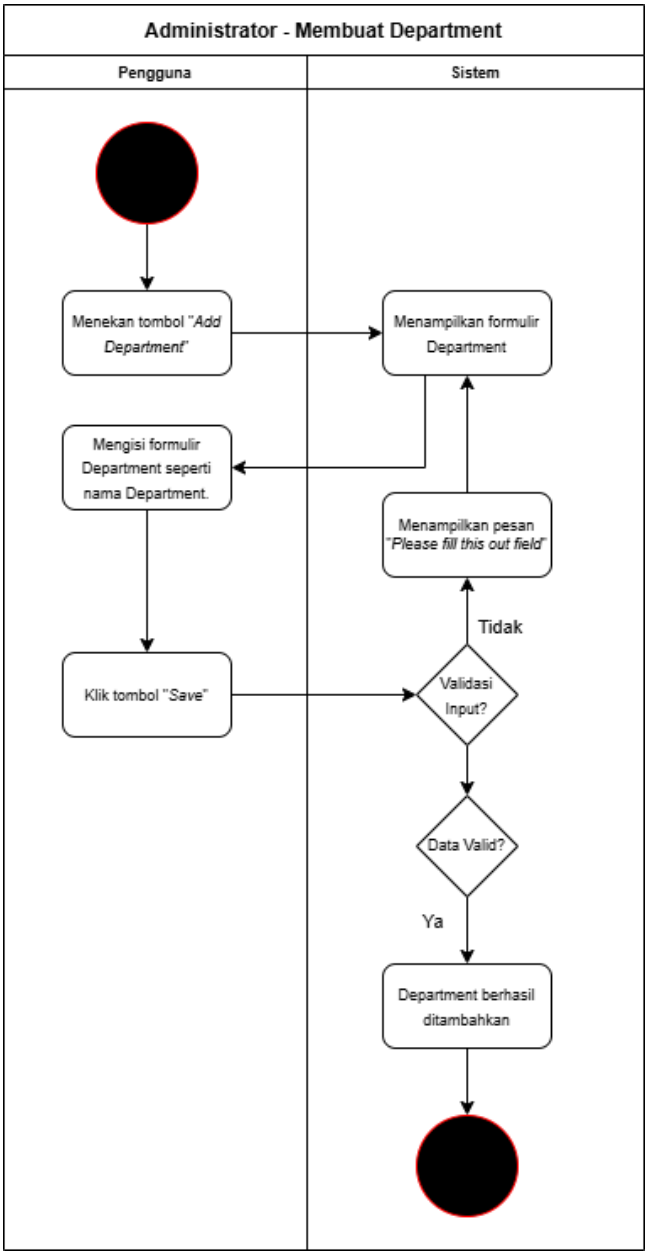
f. Activity Diagram Administrator Membuat *Project*



Gambar 10 Activity Diagram Administrator Membuat Project

Berdasarkan alur pada Gambar 10, proses pembuatan *project* baru oleh Administrator, dimulai dari menekan tombol "*Add Project*", mengisi form pembuatan *project*, dan menyimpannya. Sistem kemudian melakukan validasi data; jika valid, *project* baru berhasil dibuat, sedangkan jika tidak valid, sistem menampilkan pesan gagal menambah *project*.

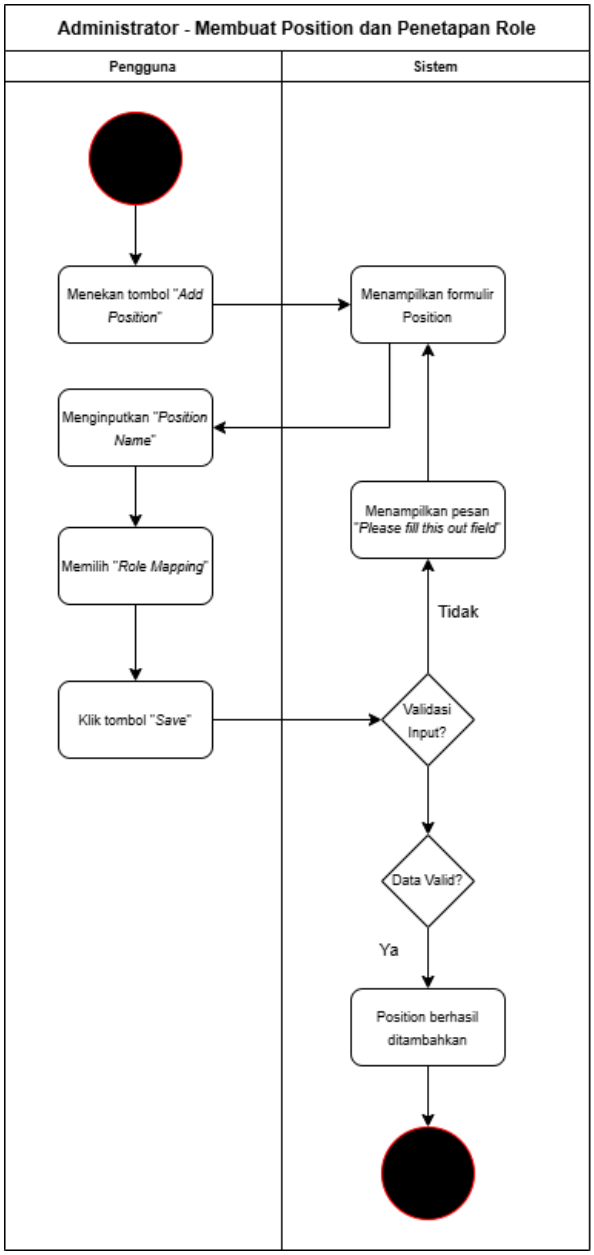
g. *Activity Diagram* Administrator Membuat *Department*



Gambar 11 Activity Diagram Administrator Membuat Department

Berdasarkan alur pada Gambar 11, proses pembuatan *department* baru oleh Administrator, dimulai dari menekan tombol "*Add Department*", mengisi form pembuatan *department*, dan menyimpannya. Sistem kemudian melakukan validasi data; jika valid, *department* baru berhasil dibuat, sedangkan jika tidak valid, sistem menampilkan pesan gagal menambah *department*.

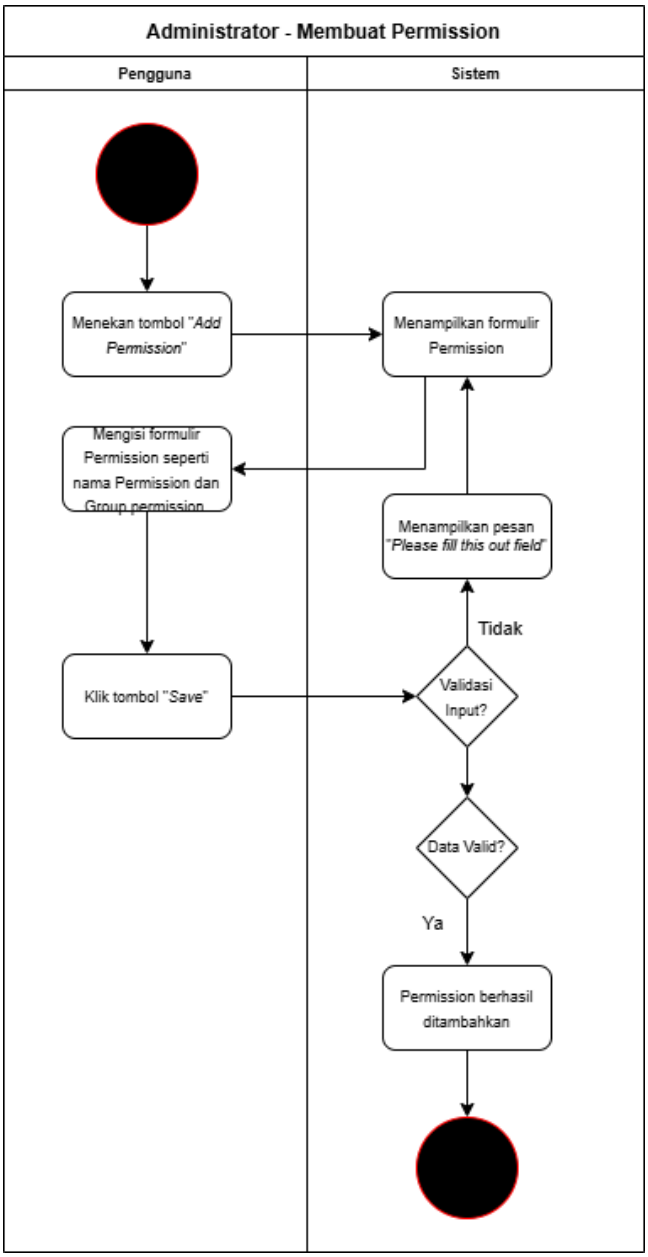
h. Activity Diagram Administrator Membuat *Position*



Gambar 12 Activity Diagram Administrator Membuat Position

Berdasarkan alur pada Gambar 12, proses pembuatan *position* baru oleh Administrator, dimulai dari menekan tombol "*Add Position*", mengisi form pembuatan *position*, dan menyimpannya. Sistem kemudian melakukan validasi data; jika valid, *position* baru berhasil dibuat, sedangkan jika tidak valid, sistem menampilkan pesan gagal menambah *Position*.

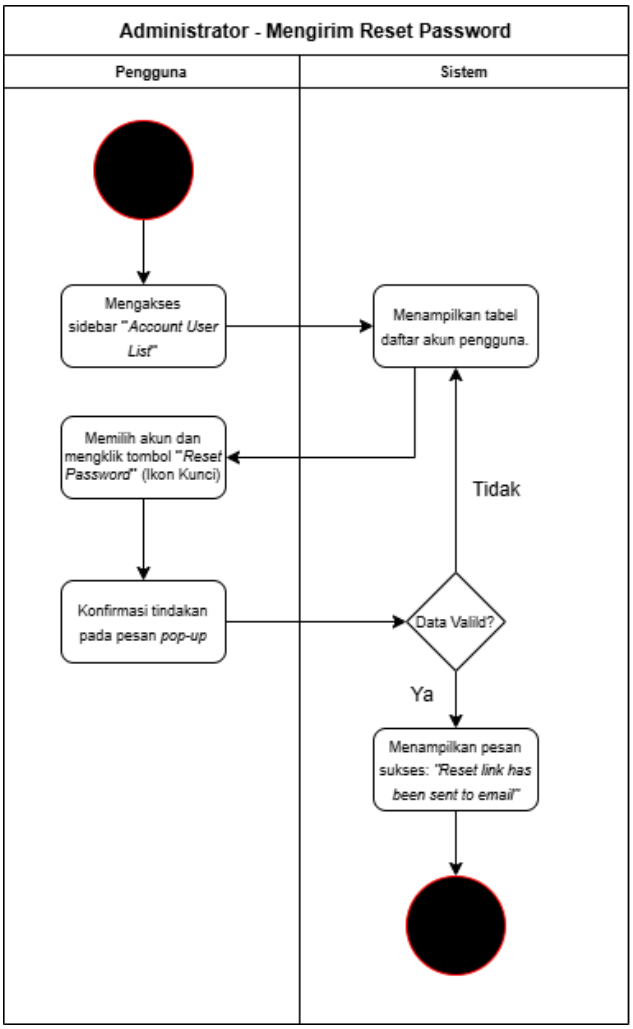
i. *Activity Diagram* Administrator Membuat *Permission*



Gambar 13 Activity Diagram Administrator Membuat Permission

Berdasarkan alur pada Gambar 13, proses penambahan *permission* baru oleh Administrator, dimulai dari menekan tombol "*Add Permisison*"**,** mengisi form *permission* dan menyimpannya. Sistem kemudian melakukan validasi data; jika valid, *Permission* berhasil dibuat, sedangkan jika tidak valid, sistem menampilkan pesan gagal menambah *Permission*.

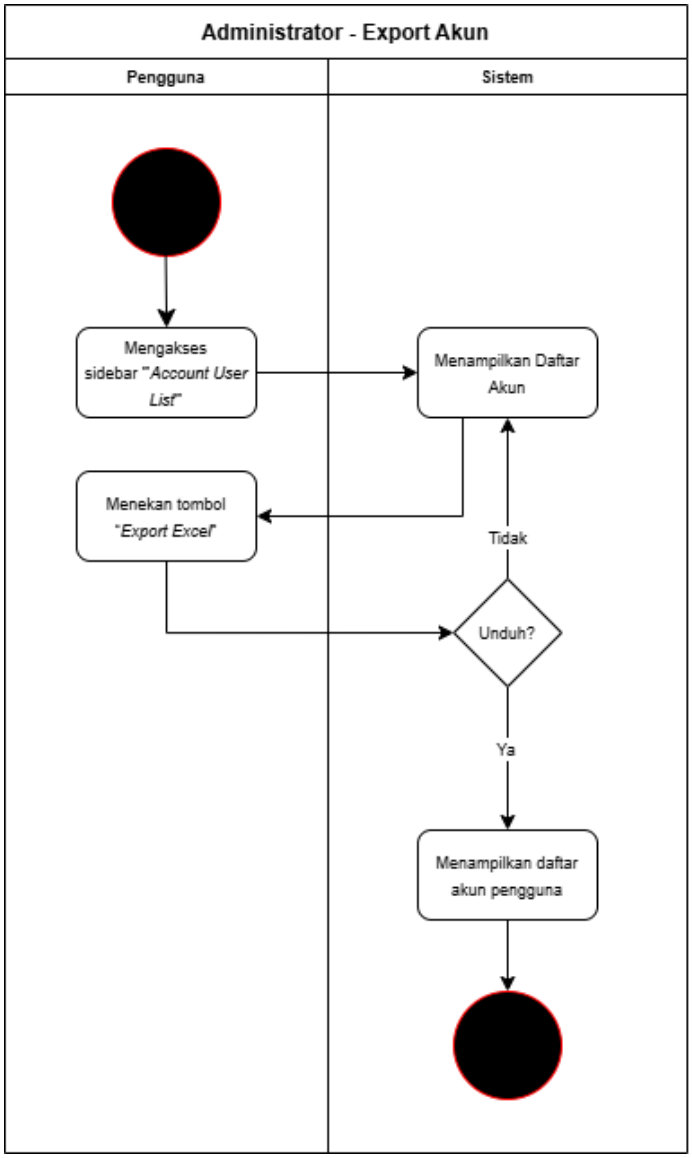
j. *Activity Diagram* Administrator *Reset* Kata Sandi Pengguna



Gambar 14 Activity Diagram Administrator Reset Kata Sandi Pengguna

Diagram ini menggambarkan proses *reset* kata sandi pengguna oleh Administrator. Proses dimulai dari memilih akun pengguna, menekan tombol *"Password"*, mengisi kata sandi baru, lalu menyimpannya. Sistem memvalidasi data; jika valid, kata sandi diperbarui, jika tidak, muncul pesan kesalahan.

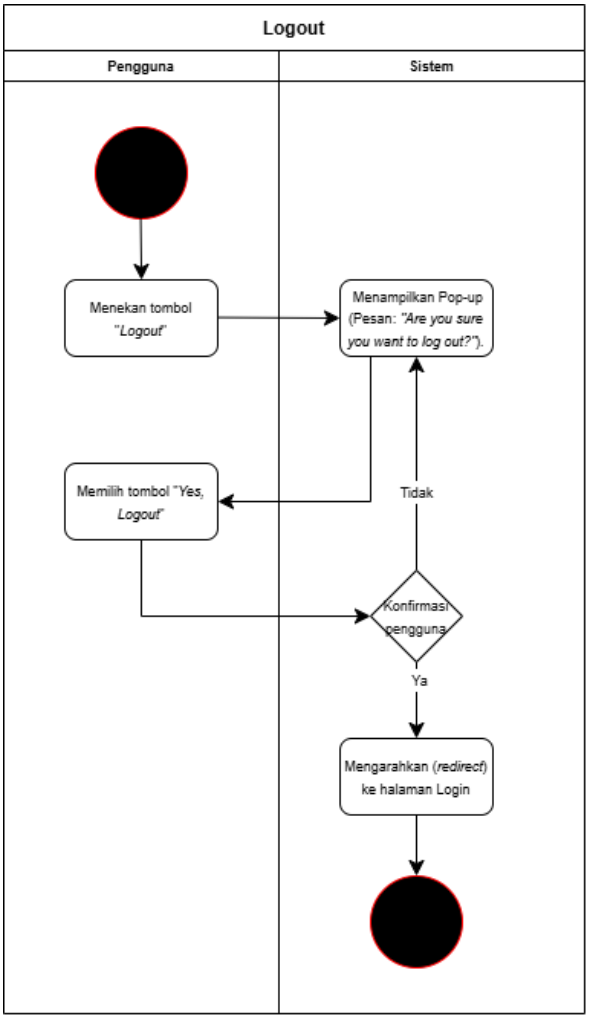
k. Activity Diagram Administrator Export Akun



Gambar 15 Activity Diagram Administrator Mengunduh Daftar Akun

Berdasarkan alur pada Gambar 15, proses Administrator mengunduh daftar akun dengan membuka daftar akun, menekan tombol "*Export Excel*", dan sistem menampilkan file akun yang berhasil diunduh.

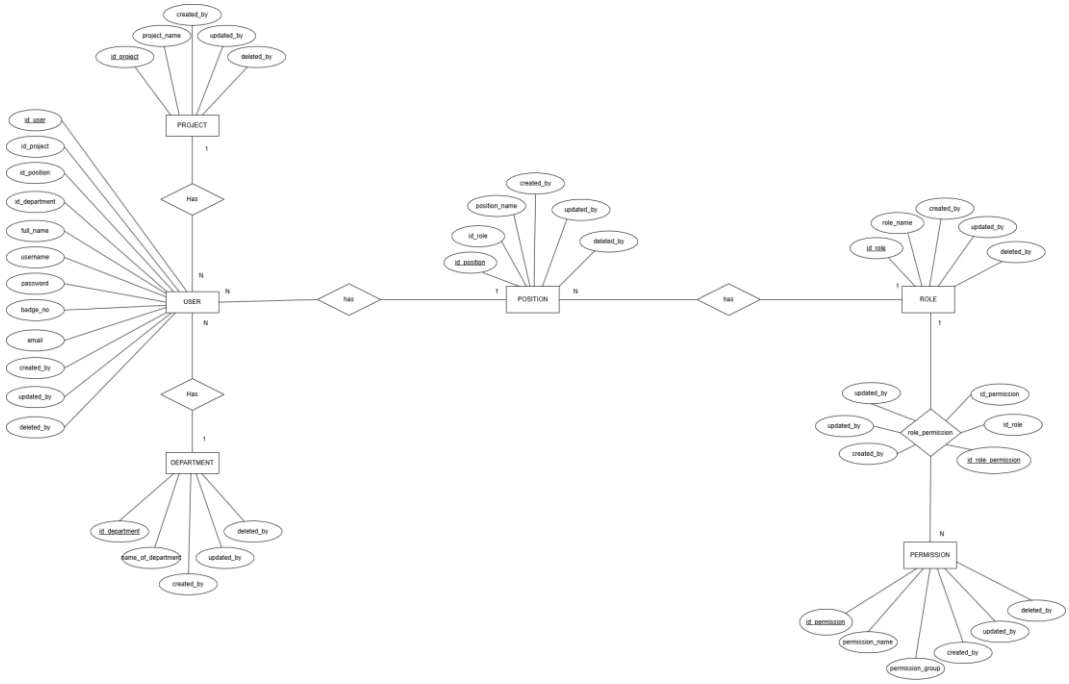
l. Activity Diagram Logout



Gambar 16 Activity Diagram Logout

Berdasarkan alur pada Gambar 16, proses *logout* pengguna, dimulai dari menekan tombol *dropdown* pada profil, memilih opsi keluar, lalu sistem mengonfirmasi dan mengeluarkan akun pengguna dari aplikasi.

3.1.2 ER Diagram

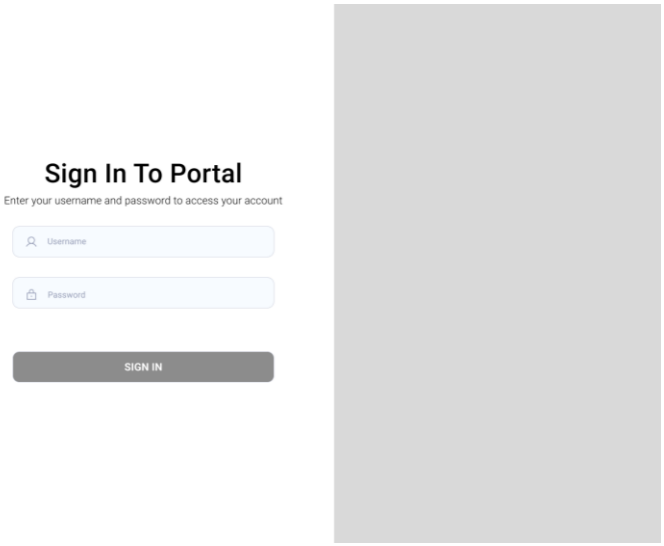


Gambar 17 ER Diagram

Entity-Relationship Diagram (ERD) merupakan model data yang digunakan untuk merepresentasikan hubungan antar entitas di dalam sebuah sistem basis data. ERD ini menggambarkan struktur logis dari penyimpanan data dengan menunjukkan entitas-entitas yang digunakan, atribut yang dimiliki setiap entitas, serta relasi yang menghubungkan antar entitas tersebut. Rancangan struktur basis data dan hubungan antar entitas pada sistem manajemen hak akses ini dapat dilihat secara detail pada Gambar 17.

3.1.3 Perancangan Antarmuka (*Wireframe)*

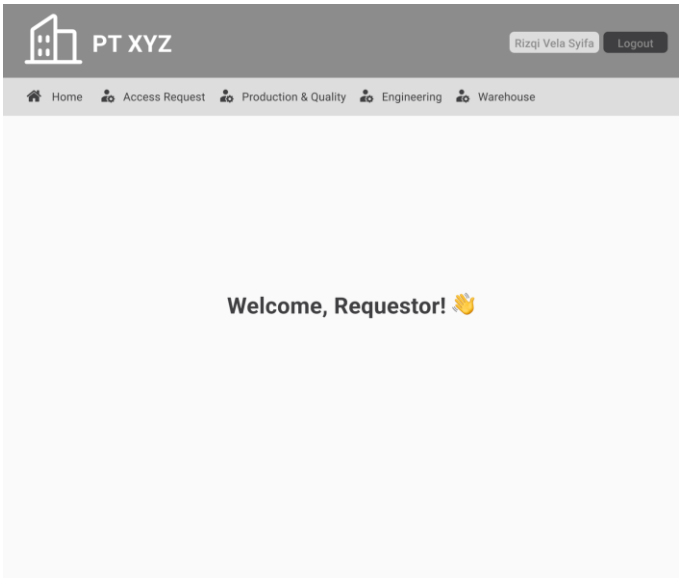
1. *Halaman Login*



Gambar 18 Wireframe Login

Gambar 18 menjelaskan halaman *login* PT XYZ terdapat dua bidang input yaitu *username* dan *password*. Tombol *login* akan mengarahkan pengguna ke halaman beranda Sistem.

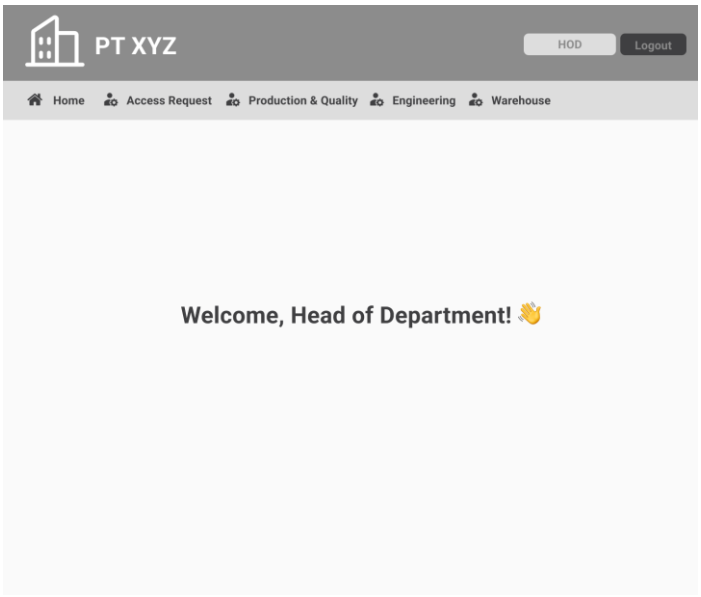
2. *Halaman Beranda*



Gambar 19 Wireframe Beranda Requestor

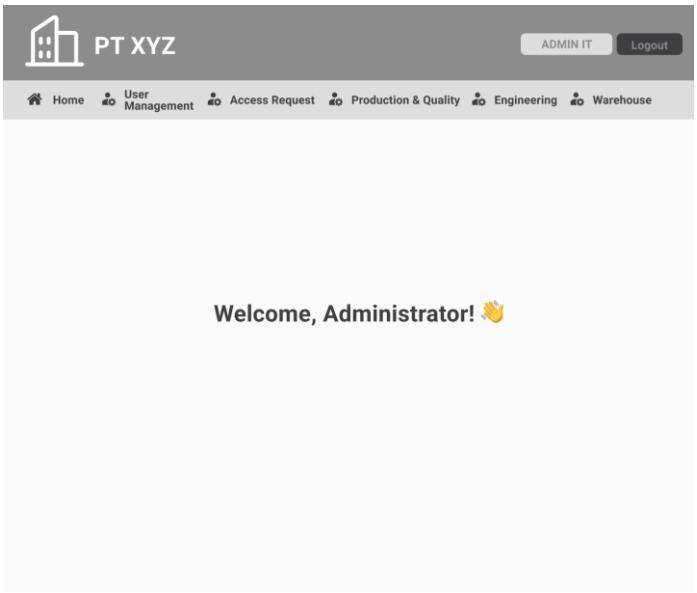
Gambar 19 menjelaskan halaman beranda sistem setelah pengguna berhasil *login* sebagai Requestor. Sistem menampilkan pesan selamat datang dan menyediakan bar navigasi atas untuk mengakses

berbagai menu utama seperti *Access Request, Production, Engineering,* dan *Warehouse*.



Gambar 20 Wireframe Beranda Head of Department Requestor

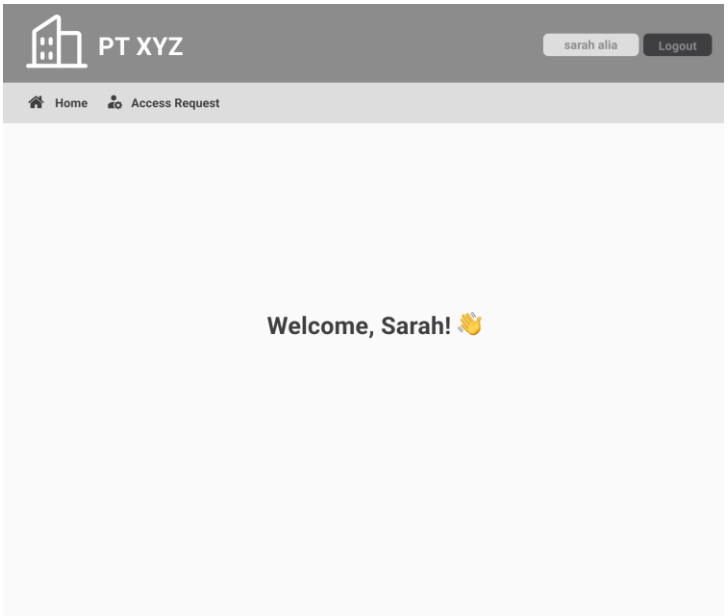
Gambar 20 menjelaskan halaman beranda sistem setelah pengguna berhasil login sebagai *Head of Department*. Sistem menampilkan pesan selamat datang dan menyediakan bar navigasi utama untuk mengakses menu seperti *Access Request, Production & Quality, Engineering,* dan *Warehouse.*



Gambar 21 Wireframe Beranda Administrator

Gambar 21 menjelaskan halaman beranda sistem setelah pengguna berhasil *login* sebagai Administrator. Sistem menampilkan

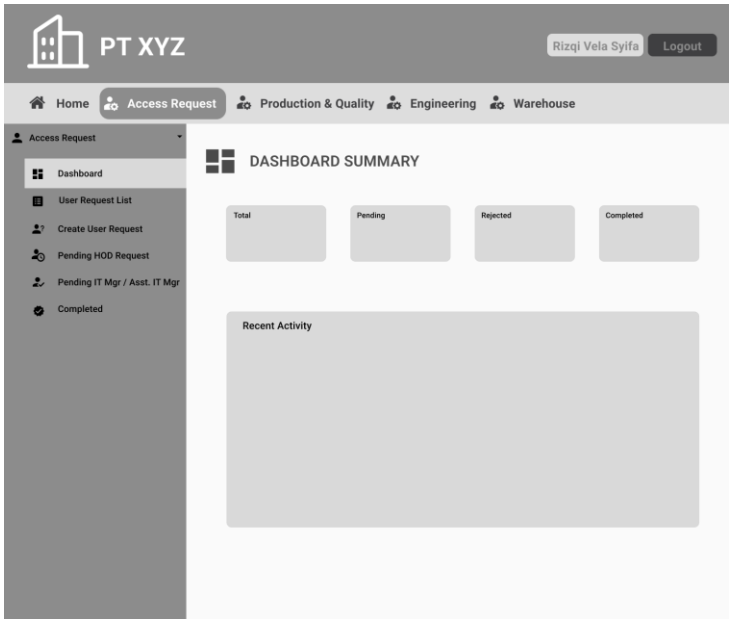
bar navigasi utama yang mencakup menu *User Management* untuk pengelolaan data master, serta menu operasional lainnya seperti *Access Request*, *Production*, *Engineering*, dan *Warehouse*.



Gambar 22 Wireframe Beranda Staff

Gambar 22 menjelaskan halaman beranda sistem setelah pengguna berhasil *login* sebagai *Staff*. Sistem menampilkan pesan selamat datang dan hanya memberikan akses terbatas pada navigasi, yaitu menu *Access Request*.

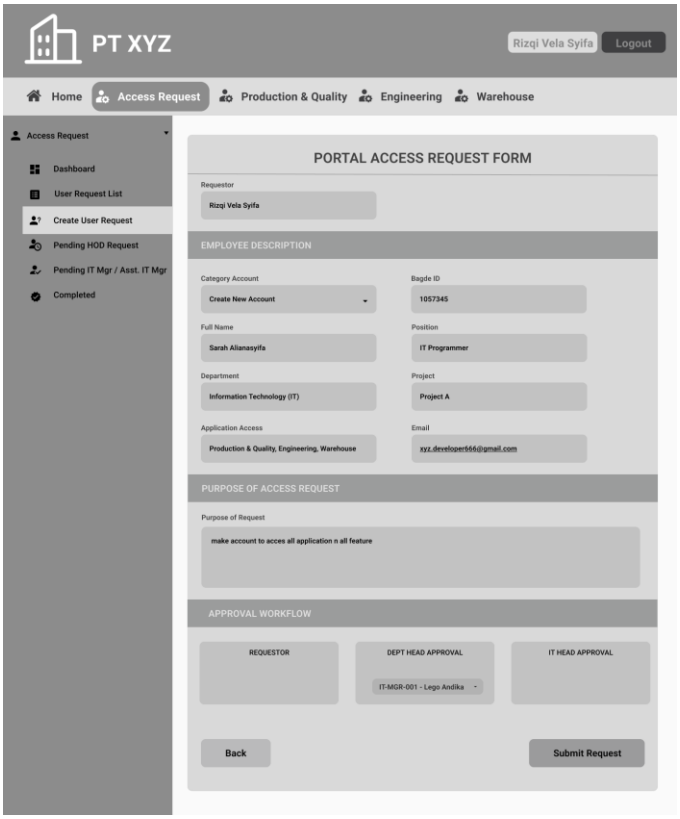
3. *Halaman Dashboard Summary (Access Request)*



Gambar 23 Wireframe Dashboard Summary

Gambar 23 menjelaskan halaman *Dashboard* Summary yang menyajikan statistik pengajuan akses secara real-time. Terdapat empat indikator utama (Total, *Pending*, *Rejected*, dan *Completed*) serta bagian *Recent Activity* untuk memantau riwayat aktivitas terbaru pada menu *Access Request*.

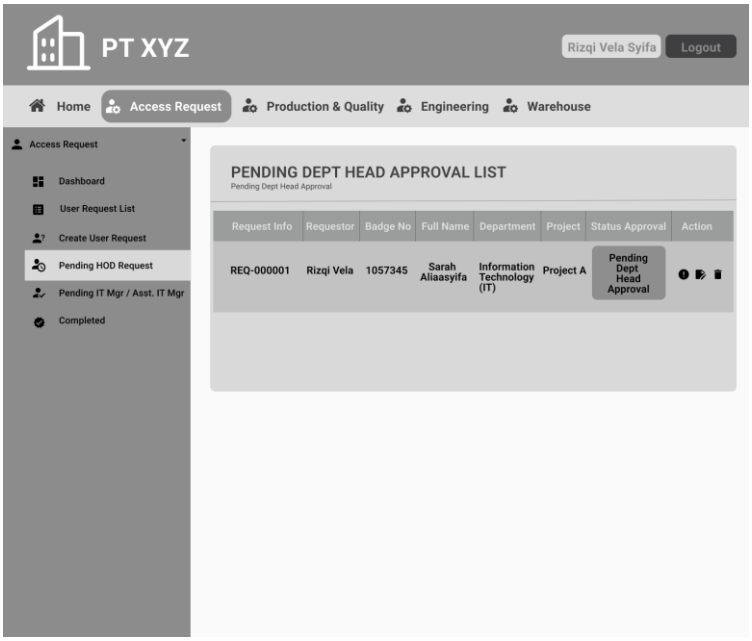
4. *Halaman Formulir Pengajuan Akses*



Gambar 24 Wireframe Formulir Pengajuan Akses

Gambar 24 menjelaskan halaman *Access Request* Form yang digunakan untuk mengajukan akses sistem bagi karyawan. *Formulir* ini mencakup detail profil karyawan (seperti *Badge ID*, *Full Name*, dan *Position*), tujuan pengajuan akses, serta alur persetujuan (*Approval Workflow*) yang melibatkan Requestor, *Dept Head*, hingga *IT Head* sebelum menekan tombol "*Submit Request*"

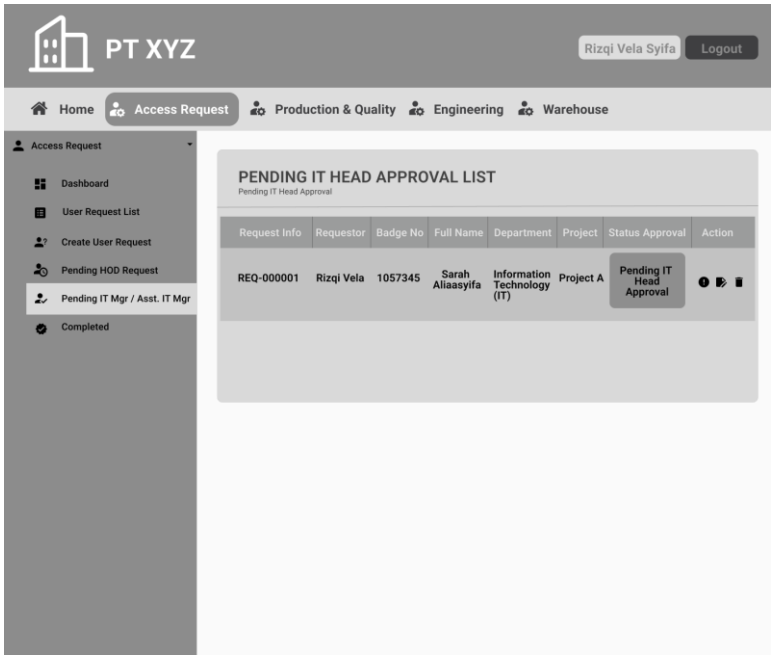
5. *Halaman Pending Head of Department Requestor*



Gambar 25 Wireframe Pending Head of Department Requestor

Gambar 25 menjelaskan halaman *Pending Dept Head Approval List* yang berisi daftar pengajuan akses yang sedang menunggu persetujuan *Head of Department* terkait.

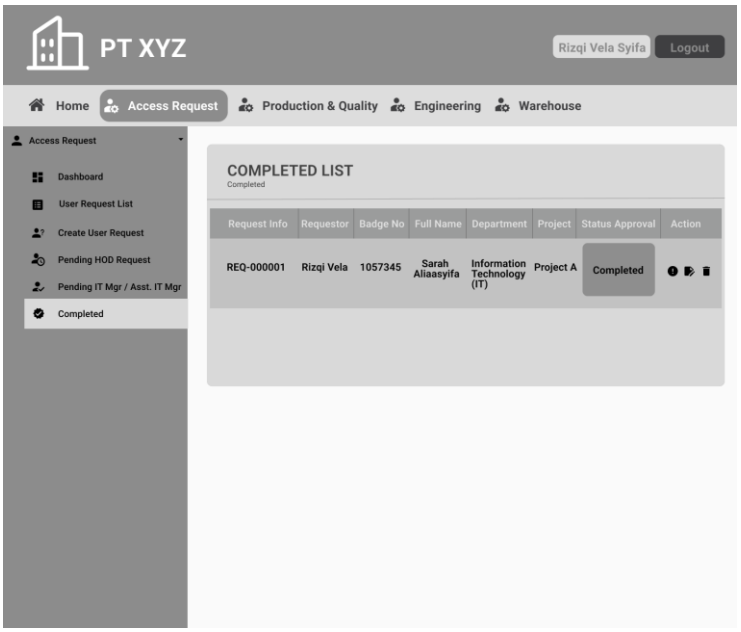
6. *Halaman Pending Head of Department IT*



Gambar 26 Wireframe Pending Head of Department IT

Gambar 26 menjelaskan halaman *Pending IT Head Approval List* yang menampilkan daftar pengajuan akses yang sedang menunggu persetujuan dari *IT Mgr / Asst IT Mgr.*

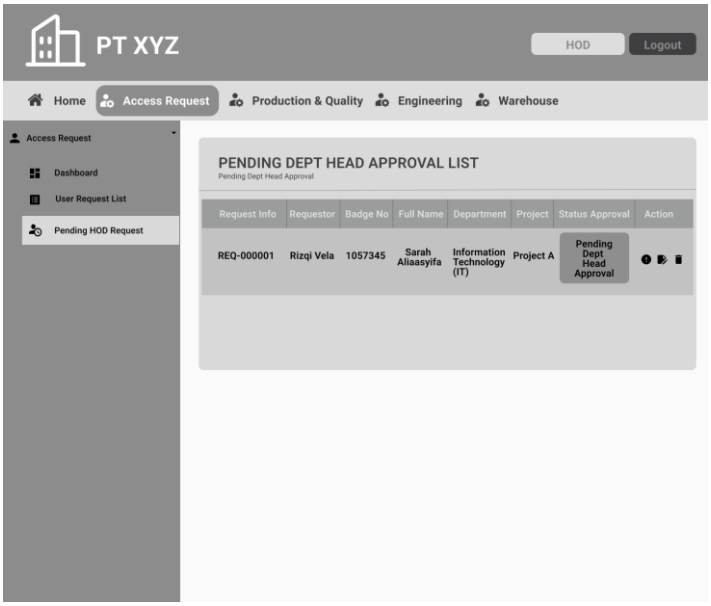
7. *Halaman Completed List*



Gambar 27 Wireframe Completed List

Gambar 27 menjelaskan halaman *Completed List* yang menampilkan daftar seluruh pengajuan akses yang telah berhasil diselesaikan atau disetujui sepenuhnya.

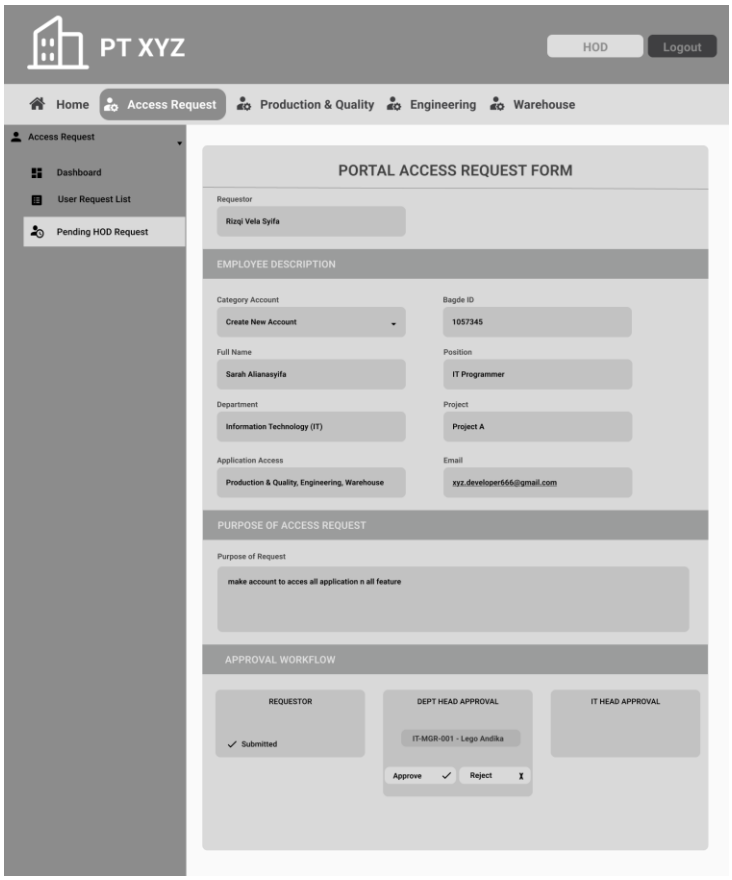
8. *Halaman Pending Head of Department Requestor*



Gambar 28 Wireframe Pending Head of Department Requestor

Gambar 28 menunjukkan halaman *Pending Dept Head Approval List* khusus untuk peran *Head of Department* yang berisi daftar pengajuan akses staf departemennya untuk divalidasi dan diproses melalui halaman detail.

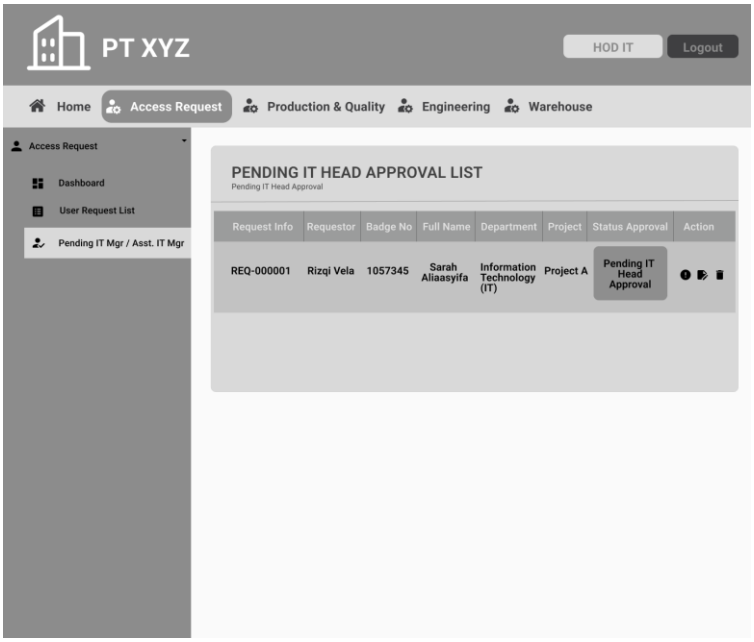
9. *Halaman Persetujuan Head of Department Requestor*



Gambar 29 Wireframe Persetujuan Head of Department Requestor

Gambar 29 menampilkan tampilan *Access Request Form* dari sisi *Head of Department*. *Head of Department* dapat meninjau detail pengajuan akses karyawan, termasuk deskripsi tujuan pengajuan, lalu mengambil tindakan melalui opsi "*Approve*" atau "*Reject*" pada bagian alur persetujuan.

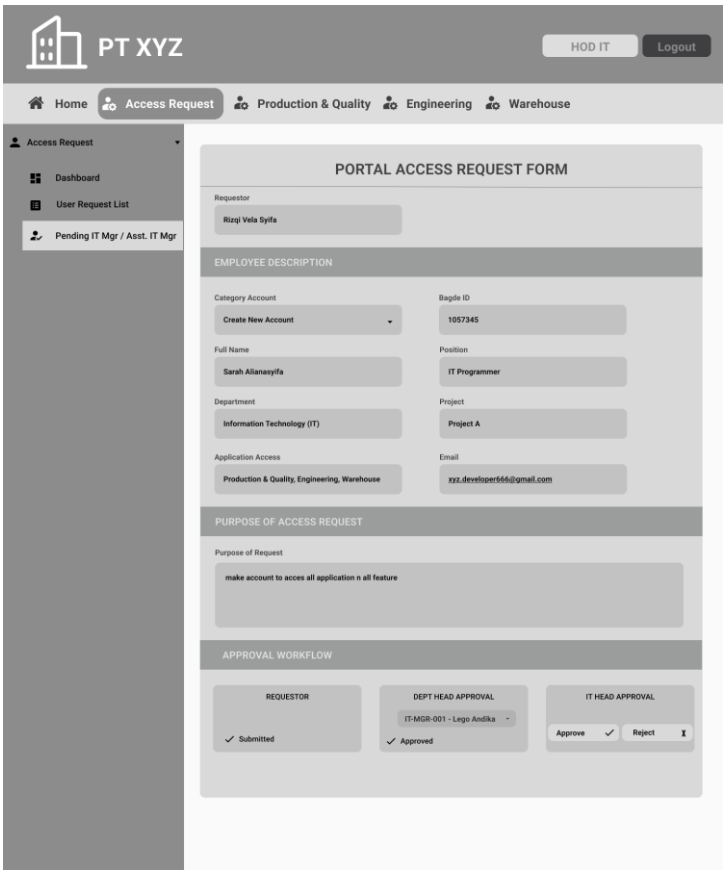
10. Halaman Pending Head of Department IT



Gambar 30 Wireframe Pending Head of Department IT

Gambar 30 menunjukkan halaman *Pending IT Head Approval List* khusus untuk peran *Head of Department* IT yang berisi daftar pengajuan akses semua departemen untuk divalidasi dan diproses melalui halaman detail.

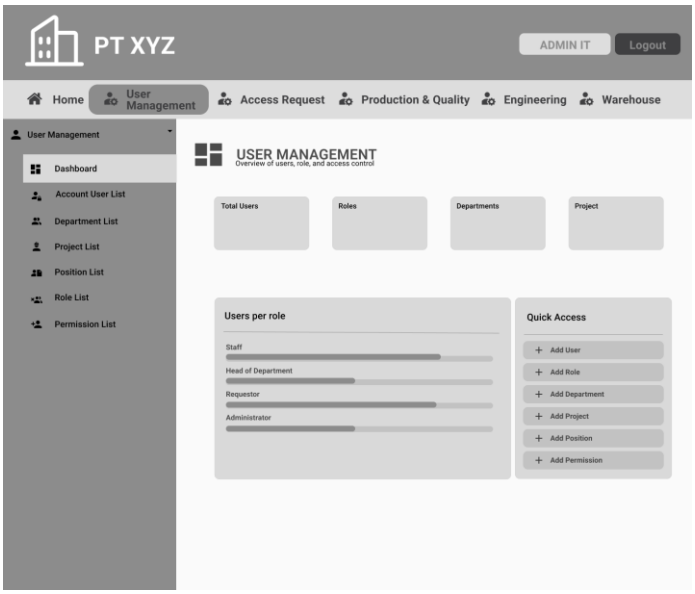
11. Halaman Persetujuan Head of Department IT



Gambar 31 Wireframe Persetujuan Head of Department IT

Gambar 31 menjelaskan tampilan *Access Request Form* dari sisi *Head of Department* IT. Pada tahap ini, pengajuan telah disetujui oleh Head of Department Requestor, sehingga *Head of Department* IT dapat melakukan tinjauan akhir dan mengambil tindakan melalui tombol "*Approve*" atau "*Reject*".

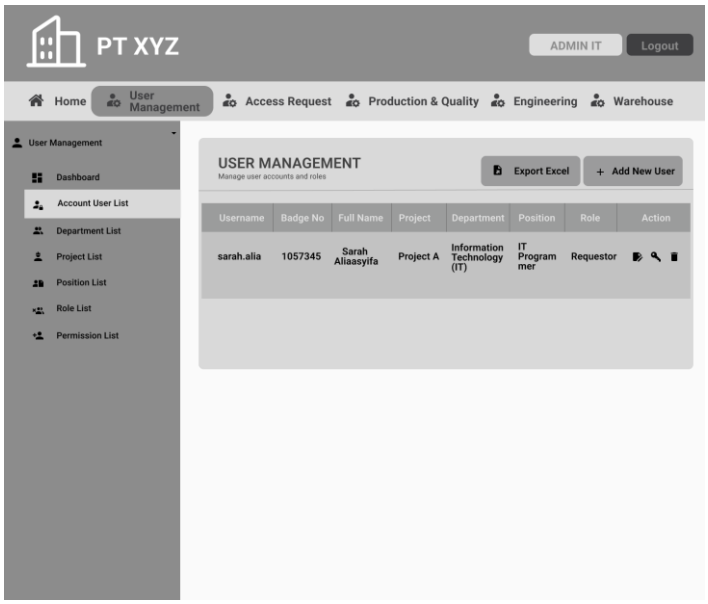
12. Halaman Dashboard User Management



Gambar 32 Wireframe Dashboard User Management

Gambar 32 menjelaskan halaman *Dashboard User Management* yang menyajikan ringkasan statistik data master seperti jumlah pengguna, peran, departemen, dan proyek. Halaman ini dilengkapi fitur *Quick Access* untuk penambahan data cepat.

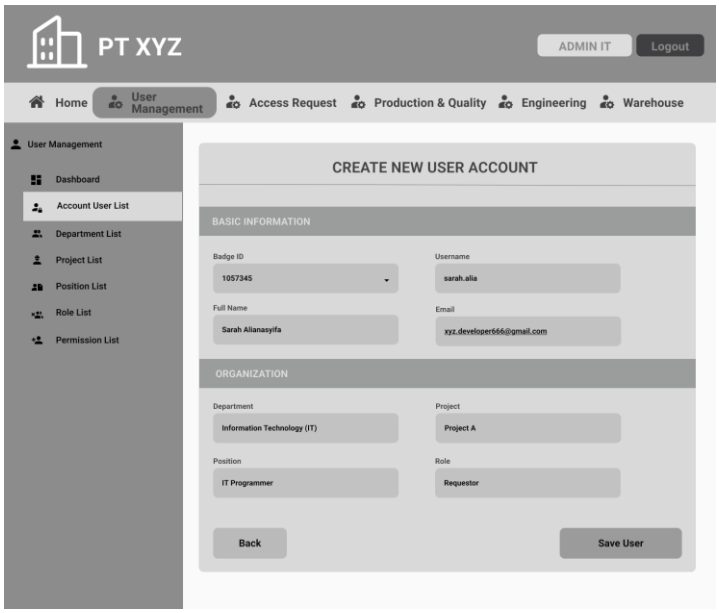
13. Halaman Account User List



Gambar 33 Wireframe Account User List

Gambar 33 menjelaskan halaman *Account User List* yang berfungsi untuk mengelola data akun pengguna dalam sistem. Halaman ini menyediakan tabel informasi detail karyawan, fitur *Export Excel* untuk laporan, serta tombol *Add New User* untuk mendaftarkan akun baru.

14. Halaman Add New User



Gambar 34 Wireframe Add New User

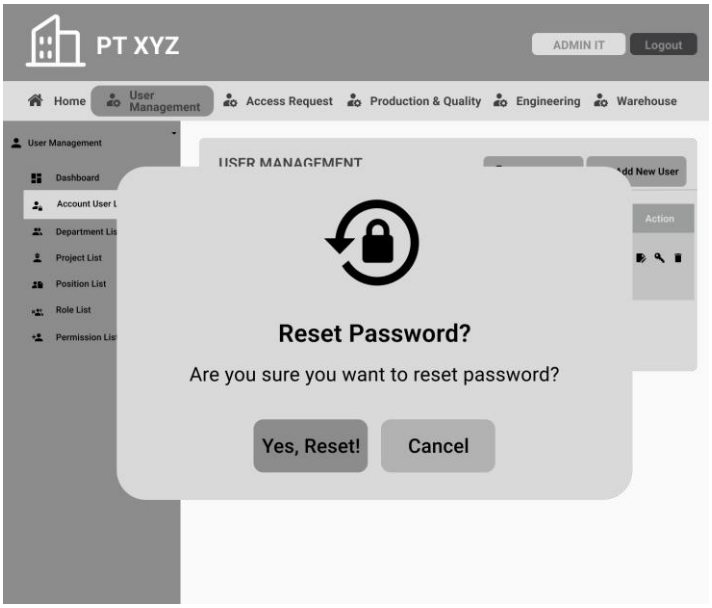
Gambar 34 menjelaskan halaman *Create New User Account* untuk pendaftaran akun baru melalui pengisian data *Basic Information* dan *Organization*. Administrator dapat menetapkan identitas serta peran pengguna sebelum menyimpan data dengan tombol *Save User.*

15. Halaman Edit User

Gambar 35 Wireframe Edit User

Gambar 35 menjelaskan halaman *Edit User* yang memungkinkan Administrator memperbarui data profil serta menambahkan *Permission* tambahan secara spesifik untuk pengguna tersebut.

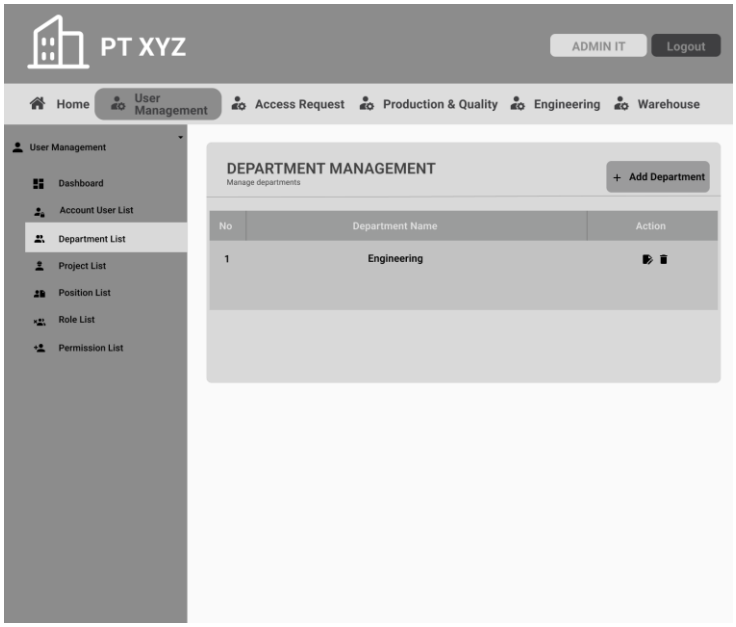
16. Halaman Administrator Reset Password



Gambar 36 Wireframe Administrator Reset Password

Gambar 36 menjelaskan tampilan *Pop-up Reset Password* yang muncul sebagai konfirmasi saat Administrator ingin mengatur ulang kata sandi pengguna. Admin dapat memilih *Yes, Reset!* untuk memproses pengaturan ulang atau *Cancel* untuk membatalkan tindakan tersebut.

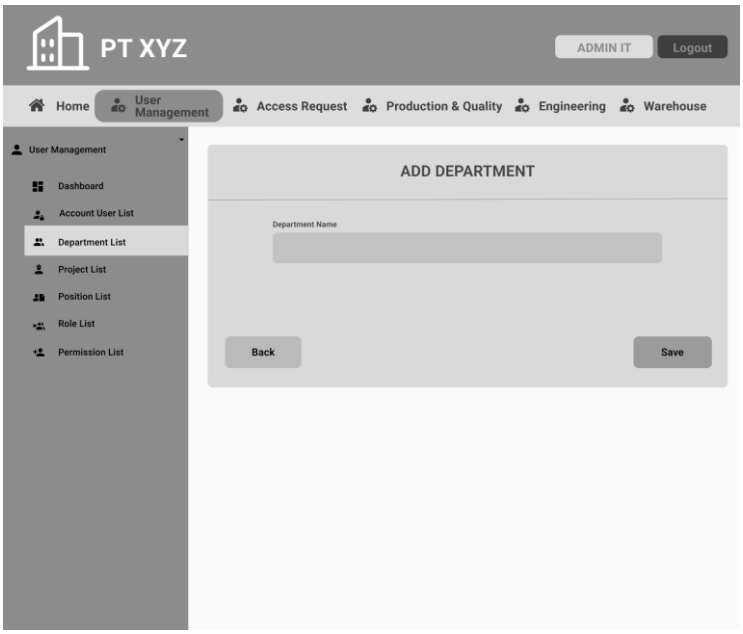
17. Halaman Department List



Gambar 37 Wireframe Department List

Gambar 37 menjelaskan halaman *Department Management* yang digunakan Administrator untuk mengelola departemen. Halaman ini menampilkan daftar departemen terdaftar serta menyediakan tombol *Add Department* untuk menambah data baru.

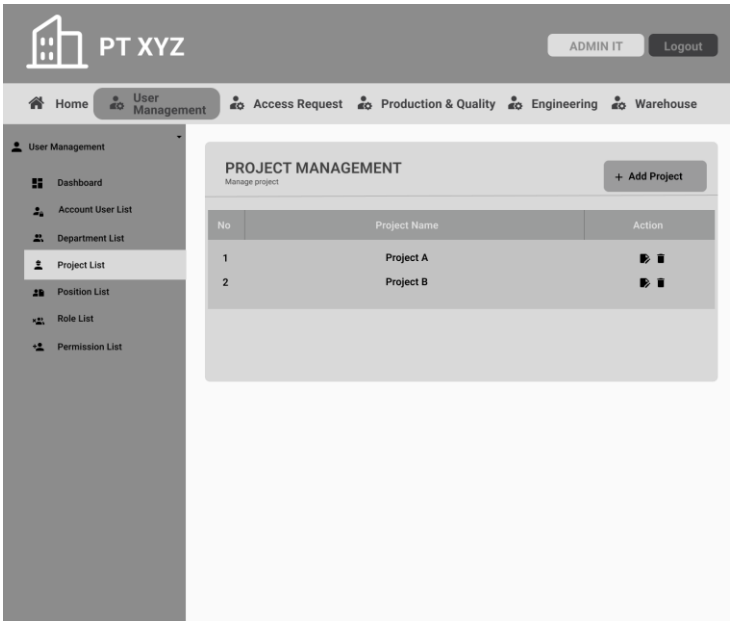
18. Halaman Add Department



Gambar 38 Wireframe Add Department

Gambar 38 memiliki kolom input *Department Name* serta tombol *Save* untuk menyimpan data dan tombol *Back* untuk kembali ke halaman daftar.

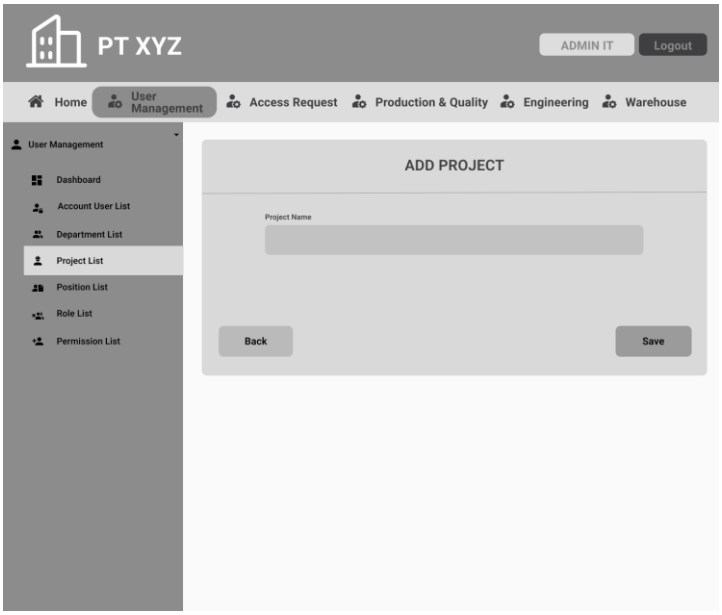
19. Halaman Project List



Gambar 39 Wireframe Add Department

Gambar 39 Administrator dapat menambah data melalui tombol *Add Project* serta mengelola proyek yang ada menggunakan fitur pada kolom *Action*.

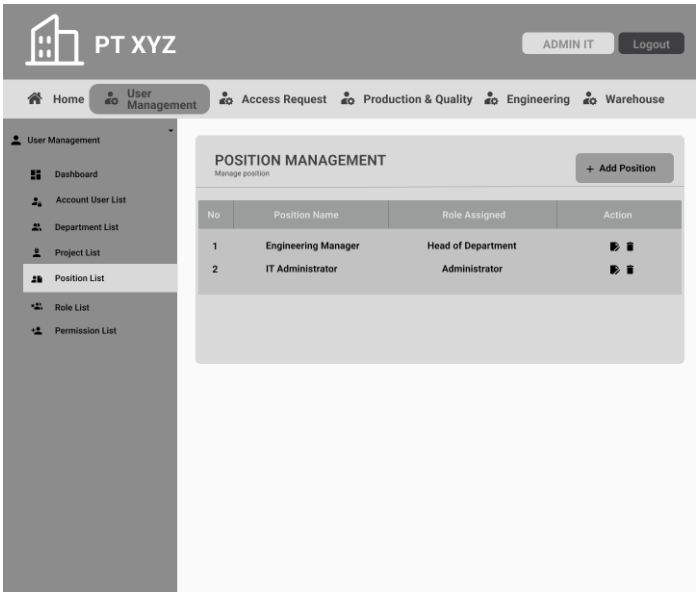
20. Halaman Add Project



Gambar 40 Wireframe Add Project

Gambar 40 memiliki kolom input *Project Name* serta tombol *Save* untuk menyimpan data dan tombol *Back* untuk kembali ke halaman daftar.

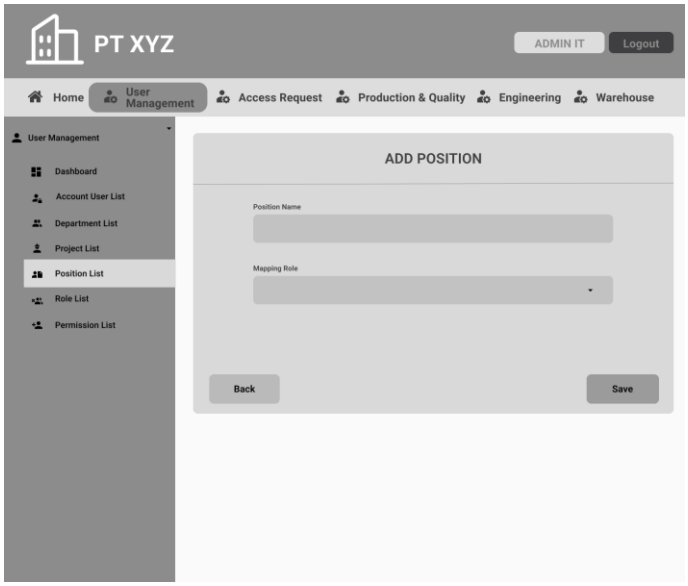
21. Halaman Position List



Gambar 41 Wireframe Position List

Gambar 41 menjelaskan halaman *Position Management* yang menampilkan daftar jabatan beserta peran yang ditetapkan dalam sistem. Administrator dapat mendaftarkan jabatan baru melalui tombol *Add Position* serta mengelola data yang ada melalui fitur pada kolom *Action*.

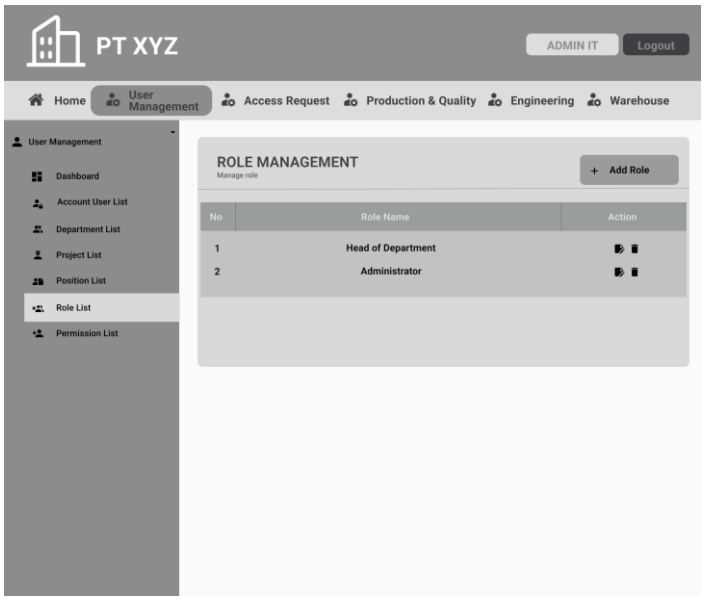
22. Halaman Add Position



Gambar 42 Wireframe Add Position

Gambar 42 memiliki kolom input *Position Name* serta *Mapping Role* untuk menghubungkan jabatan dengan peran tertentu.

23. Halaman Role List

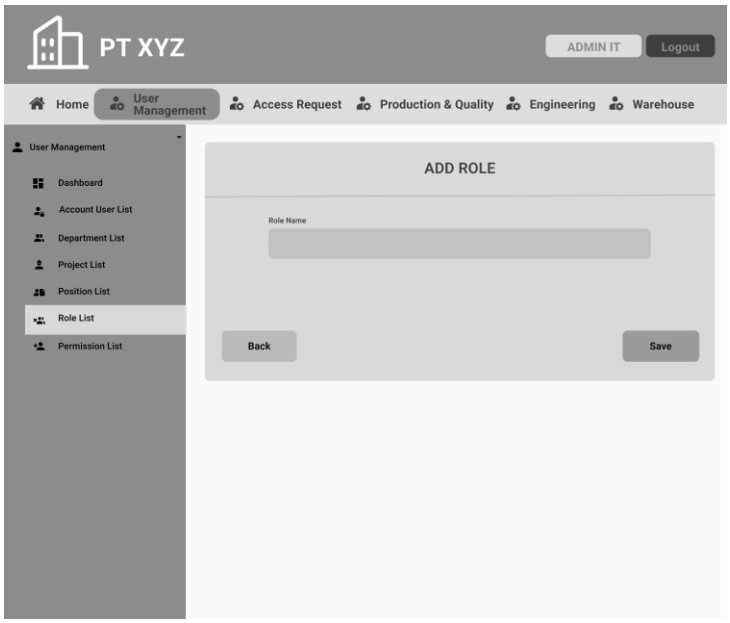


Gambar 43 Wireframe Role List

Gambar 43 menjelaskan halaman *Role Management* yang menampilkan daftar peran (*role*) yang tersedia dalam sistem. Administrator dapat menambahkan peran baru melalui tombol *Add*

Role serta mengelola data yang ada menggunakan fitur pada kolom *Action*.

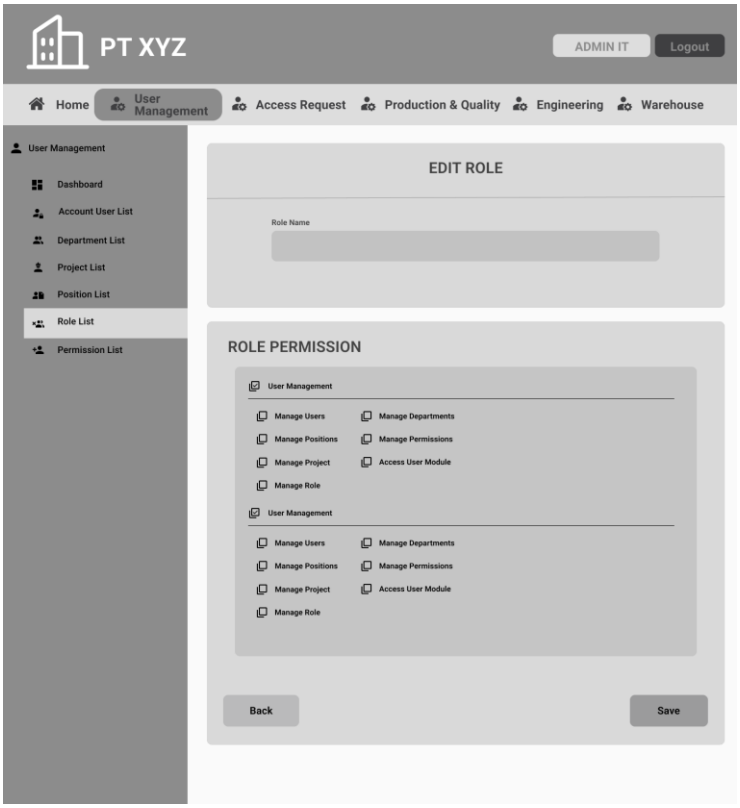
24. Halaman Add Role



Gambar 44 Wireframe Add Role

Gambar 44 memiliki kolom input *Role Name* untuk mendaftarkan nama peran baru ke dalam sistem. Terdapat juga tombol *Save* untuk menyimpan data dan tombol *Back* untuk kembali ke halaman daftar peran.

25. Halaman Edit Role



Gambar 45 Wireframe Edit Role

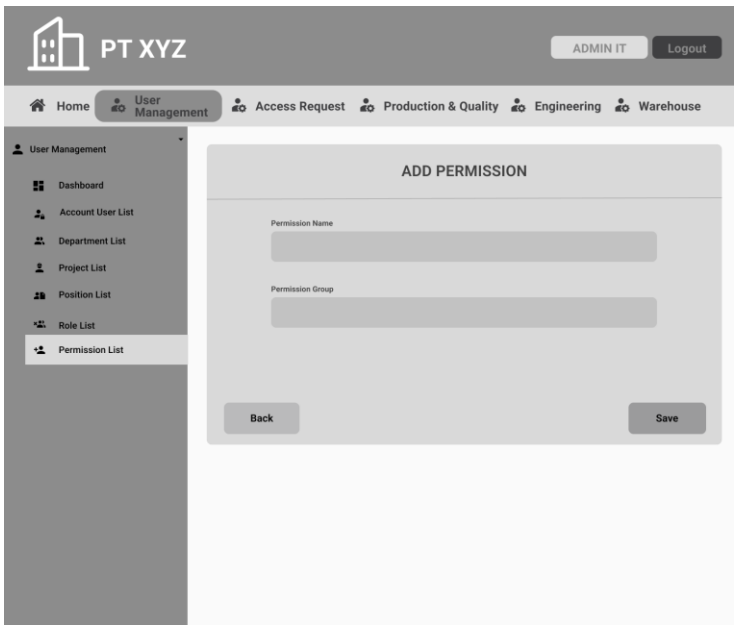
Gambar 45 menjelaskan halaman *Edit Role* yang digunakan untuk memperbarui nama peran serta mengatur *Role Permission*

26. Halaman Permission List

Gambar 46 Wireframe Permission List

Gambar 46 menjelaskan halaman *Permissions Management* yang menampilkan daftar hak akses spesifik beserta grup menunya dalam sistem. Administrator dapat menambah hak akses baru melalui tombol *Add Permission* serta mengelola data yang ada melalui kolom *Action*.

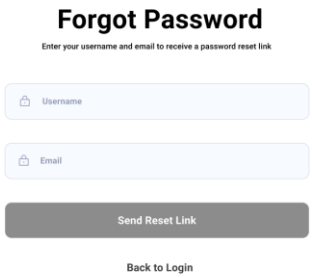
27. Halaman Add Permission



Gambar 47 Wireframe Add Permission

Gambar 47 memiliki kolom input *Permission Name* dan *Permission Group* untuk mendaftarkan hak akses baru berdasarkan kategori menunya.

28. Halaman Forgot Password



Gambar 48 Wireframe Forgot Password

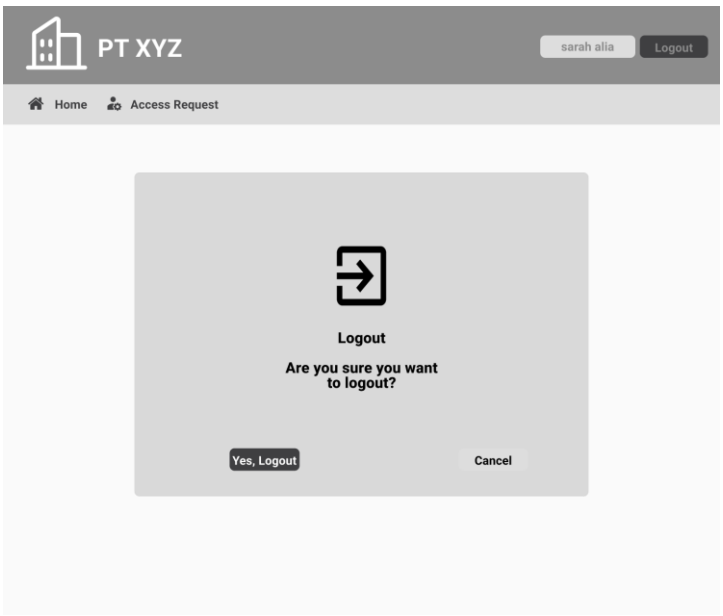
Gambar 48 menjelaskan halaman *Forgot Password* untuk pemulihan akun melalui input *Username* dan *Email*. Pengguna dapat mengeklik *Send Reset Link* untuk menerima tautan pengaturan ulang kata sandi atau memilih *Back to Login* untuk kembali.

29. Halaman Mengatur Kata Sandi

Gambar 49 Wireframe Mengatur Kata Sandi

Gambar 49 pengguna perlu menginput kata sandi baru pada kolom *New Password* dan melakukan verifikasi di kolom *Re-type Password* sebelum menyimpannya melalui tombol *Save New Password*.

30. Halaman Logout



Gambar 50 Wireframe Logout

Gambar 50 menjelaskan tampilan *Pop-up Logout* yang muncul sebagai konfirmasi keamanan saat pengguna mengeklik tombol keluar. Pengguna dapat memilih tombol *Yes*, *Logout* untuk mengakhiri sesi atau *Cancel* untuk kembali ke halaman sebelumnya.

3.1.2 Rencana Pelaksanaan

Adapun rencana pelaksanaan pembuatan tugas akhir dapat disajikan dalam bentuk *gantt chart* sebagai berikut:

No	Kegiatan	Bulan Pelaksanaan								
		Agu 2025	Sep 2025	Okt 2025	Nov 2025	Des 2025	Jan 2026	Feb 2026	Mar 2026	Apr 2026
1	Planning									
2	Design									
3	Develop									
4	Test									
5	Deploy									
6	Review									

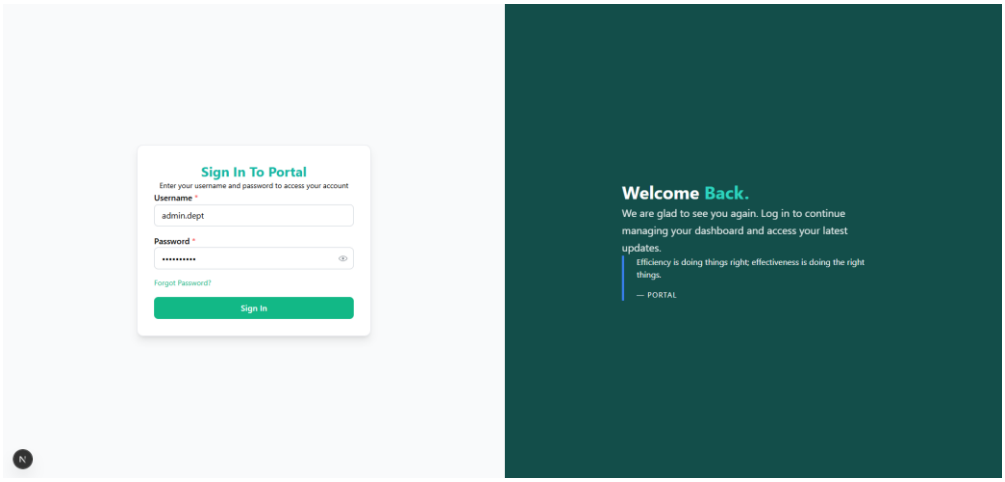
Gambar 51 Rencana Pelaksanaan

Sebagaimana yang ditunjukkan pada Gambar 51, merupakan *Gantt Chart* dari rencana pelaksanaan pembuatan tugas akhir Pengembangan Fitur Manajemen Hak Akses Pengguna pada Sistem Internal Perusahaan dengan *Metode Role-Based Access Control* (RBAC), dari tahap *Planning* sampai dengan *Review*.

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1 Hasil Implementasi

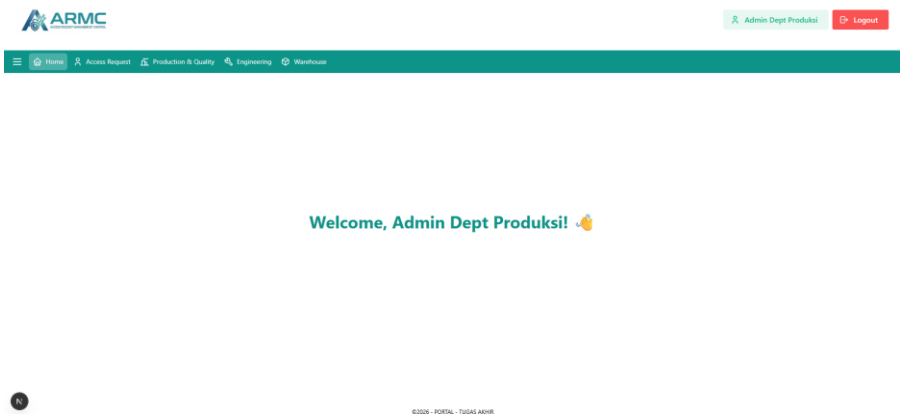
1. *Halaman Login*



Gambar 52 Wireframe Login

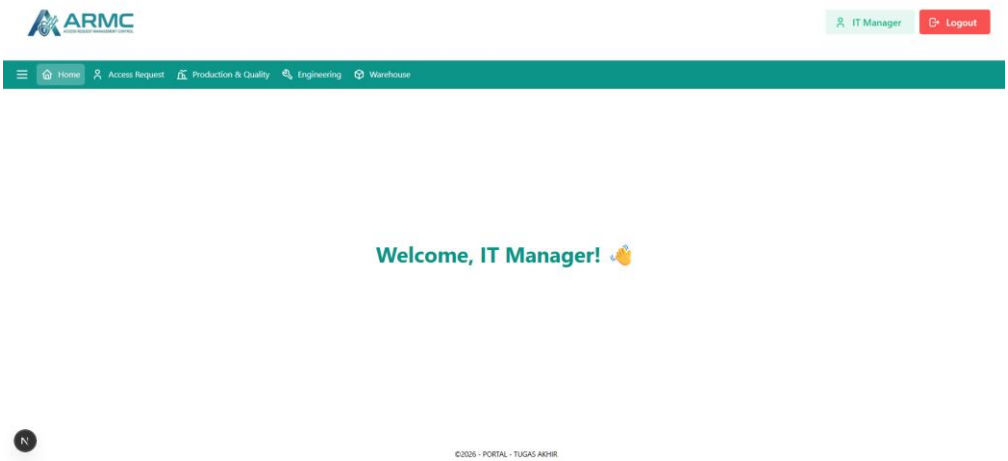
Gambar 52 menjelaskan halaman *login* PT XYZ terdapat dua bidang input yaitu *username* dan *password*. Tombol *login* akan mengarahkan pengguna ke halaman beranda Sistem.

2. *Halaman Beranda*



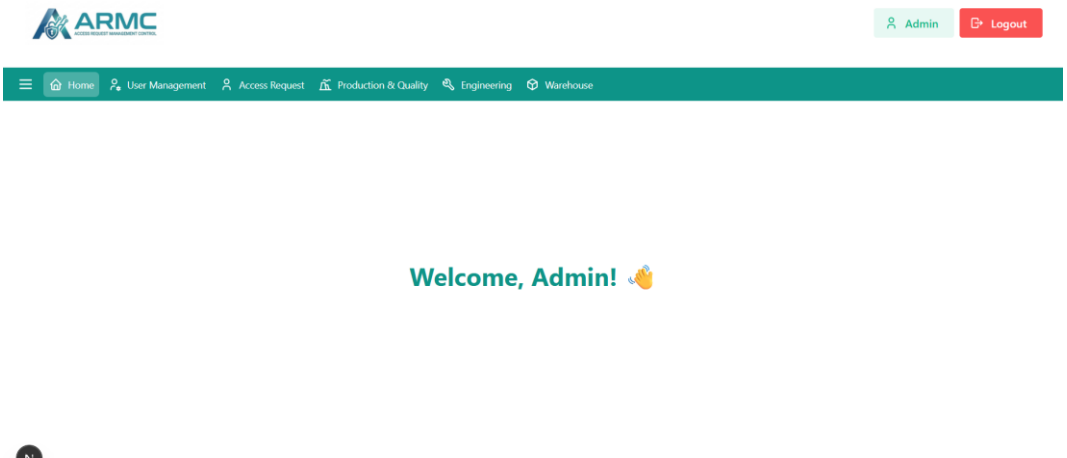
Gambar 53 Wireframe Beranda Requestor

Gambar 53 menjelaskan halaman beranda sistem setelah pengguna berhasil *login* sebagai Requestor. Sistem menampilkan pesan selamat datang dan menyediakan bar navigasi atas untuk mengakses berbagai menu utama seperti *Access Request, Production, Engineering,* dan *Warehouse*.



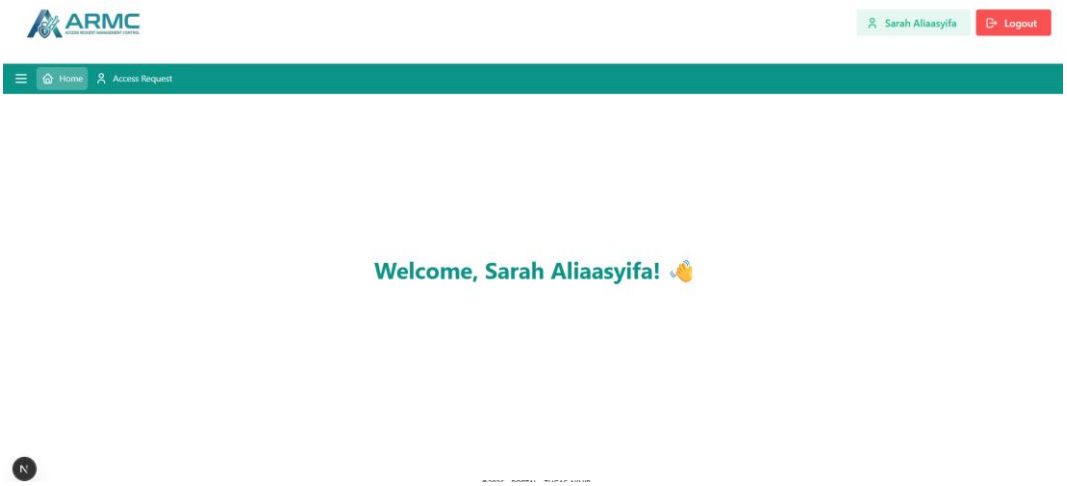
Gambar 54 Wireframe Beranda Head of Department Requestor

Gambar 54 menjelaskan halaman beranda sistem setelah pengguna berhasil login sebagai *Head of Department*. Sistem menampilkan pesan selamat datang dan menyediakan bar navigasi utama untuk mengakses menu seperti *Access Request, Production & Quality, Engineering,* dan *Warehouse.*



Gambar 55 Wireframe Beranda Administrator

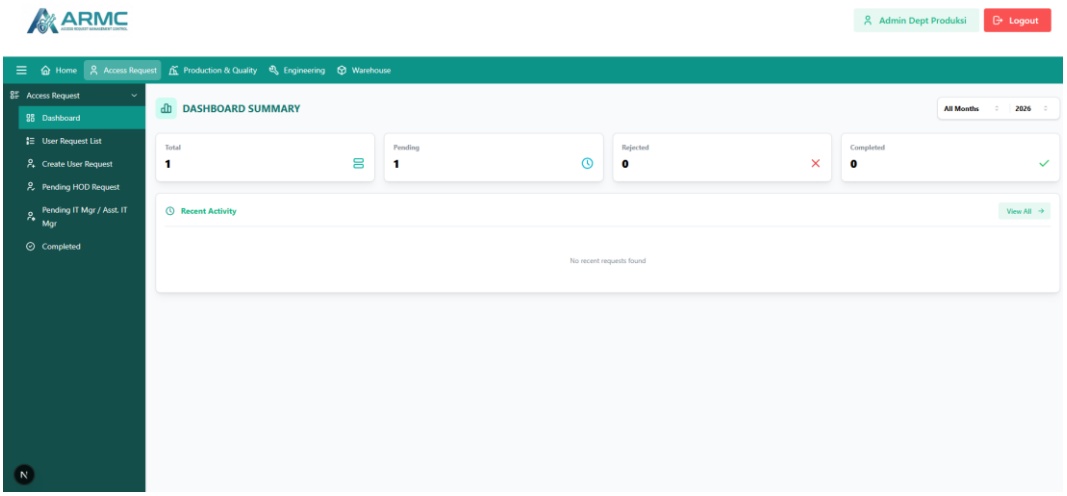
Gambar 55 menjelaskan halaman beranda sistem setelah pengguna berhasil *login* sebagai Administrator. Sistem menampilkan bar navigasi utama yang mencakup menu *User Management* untuk pengelolaan data master, serta menu operasional lainnya seperti *Access Request*, *Production*, *Engineering*, dan *Warehouse*.



Gambar 56 Wireframe Beranda Staff

Gambar 56 menjelaskan halaman beranda sistem setelah pengguna berhasil *login* sebagai *Staff*. Sistem menampilkan pesan selamat datang dan hanya memberikan akses terbatas pada navigasi, yaitu menu *Access Request*.

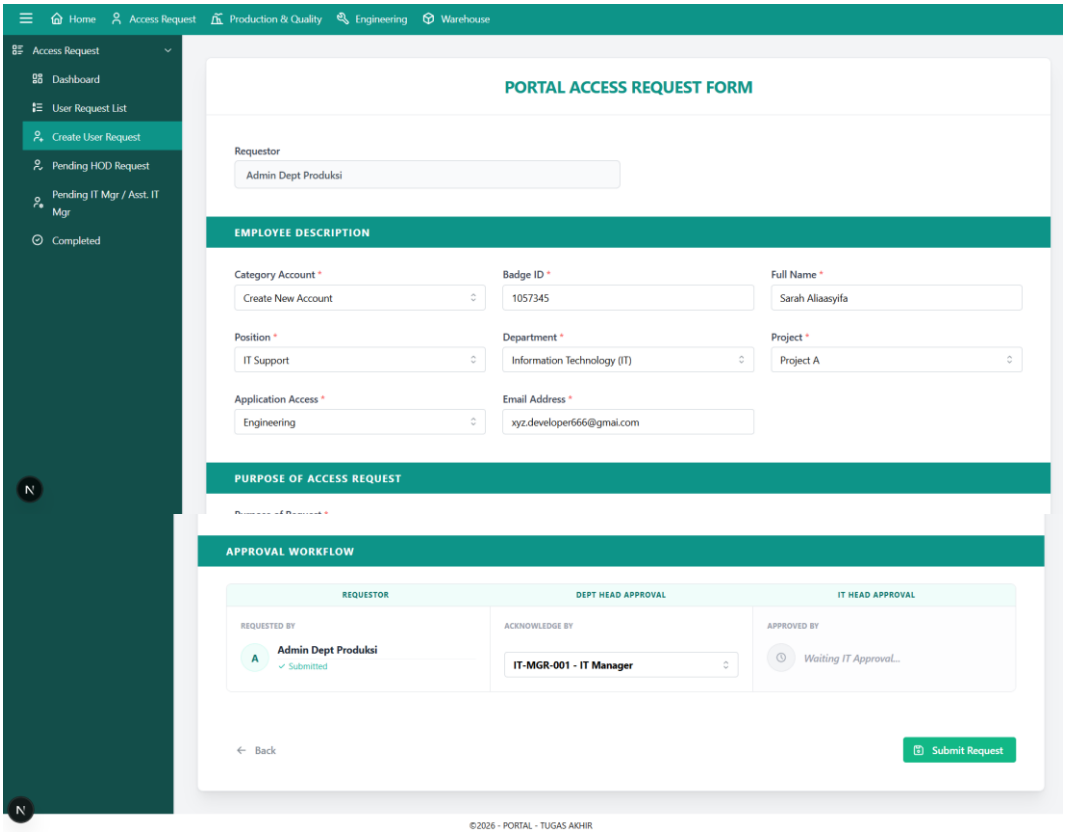
3. Halaman Dashboard Summary (Access Request)



Gambar 57 Wireframe Dashboard Summary

Gambar 57 ini menjelaskan halaman *Dashboard* Summary yang menyajikan statistik pengajuan akses secara real-time. Terdapat empat indikator utama (Total, *Pending*, *Rejected*, dan *Completed*) serta bagian *Recent Activity* untuk memantau riwayat aktivitas terbaru pada menu *Access Request*.

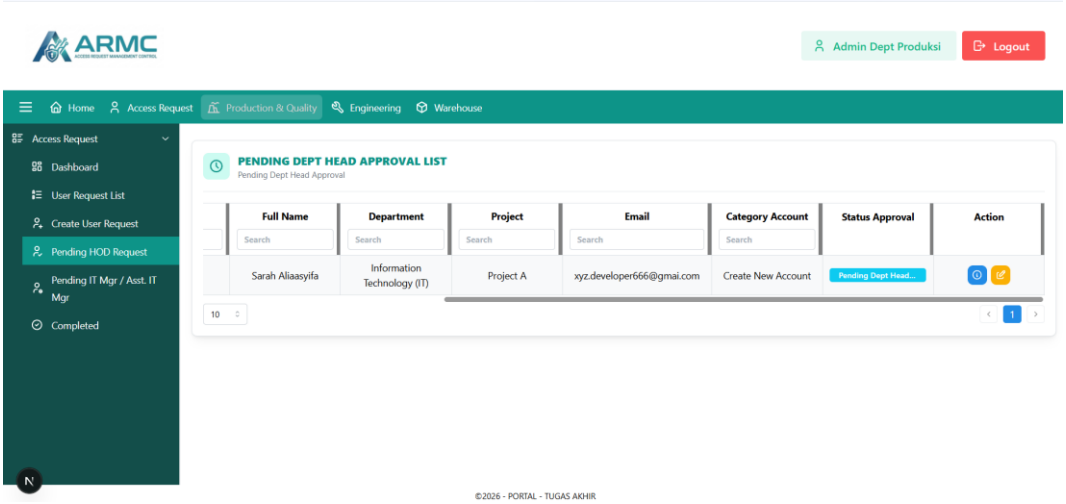
4. *Halaman Formulir Pengajuan Akses*



Gambar 58 Wireframe Formulir Pengajuan Akses

Gambar 58 menjelaskan halaman *Access Request* Form yang digunakan untuk mengajukan akses sistem bagi karyawan. *Formulir* ini mencakup detail profil karyawan (seperti *Badge ID*, *Full Name*, dan *Position*), tujuan pengajuan akses, serta alur persetujuan (*Approval Workflow*) yang melibatkan Requestor, *Dept Head*, hingga *IT Head* sebelum menekan tombol "*Submit Request*"

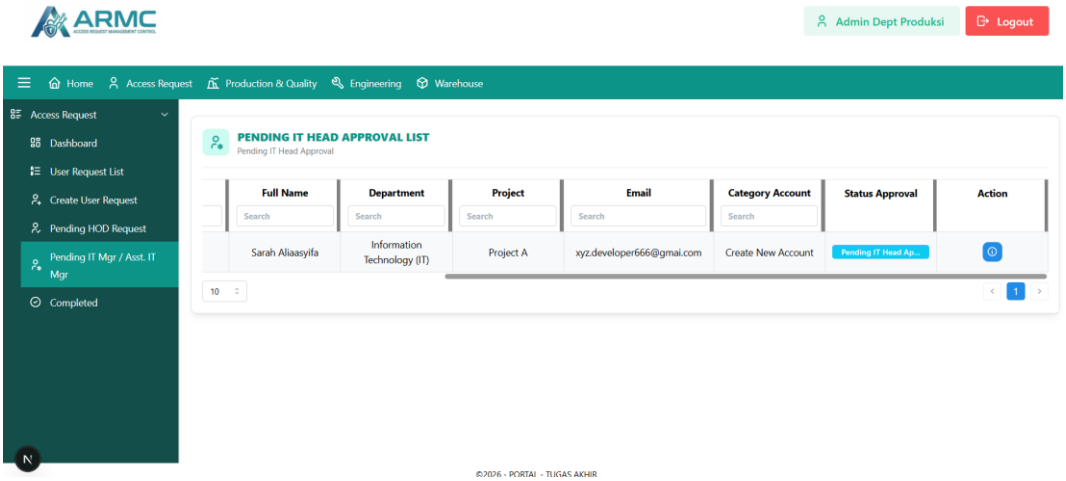
5. *Halaman Pending Head of Department Requestor*



Gambar 59 Wireframe Pending Head of Department Requestor

Gambar 59 menjelaskan halaman *Pending Dept Head Approval List* yang berisi daftar pengajuan akses yang sedang menunggu persetujuan *Head of Department* terkait.

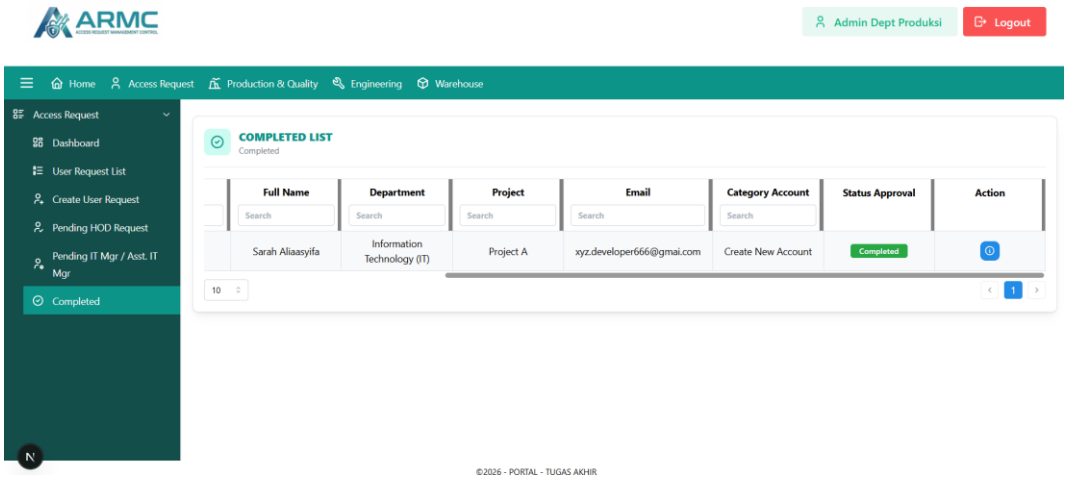
6. *Halaman Pending Head of Department IT*



Gambar 60 Wireframe Pending Head of Department IT

Gambar 60 menjelaskan halaman *Pending IT Head Approval List* yang menampilkan daftar pengajuan akses yang sedang menunggu persetujuan dari *IT Mgr / Asst IT Mgr.*

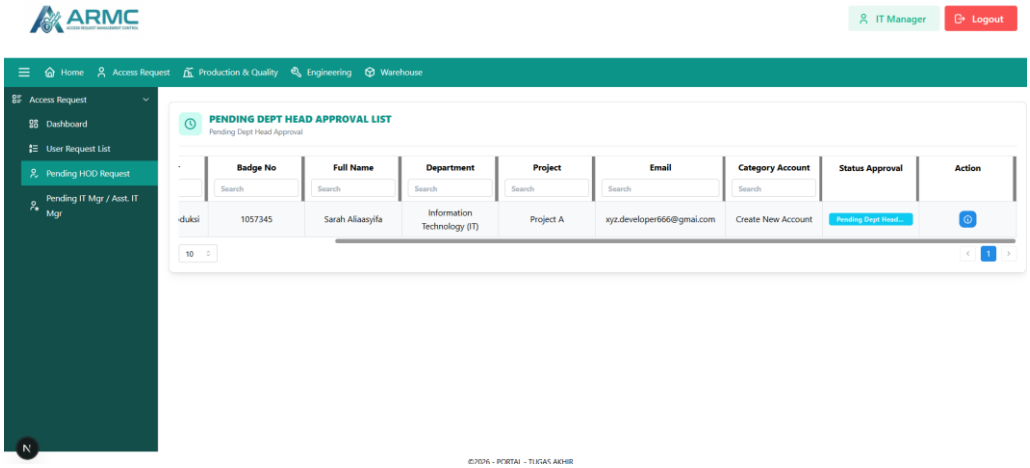
7. *Halaman Completed List*



Gambar 61 Wireframe Completed List

Gambar 61 menjelaskan halaman *Completed List* yang menampilkan daftar seluruh pengajuan akses yang telah berhasil diselesaikan atau disetujui sepenuhnya.

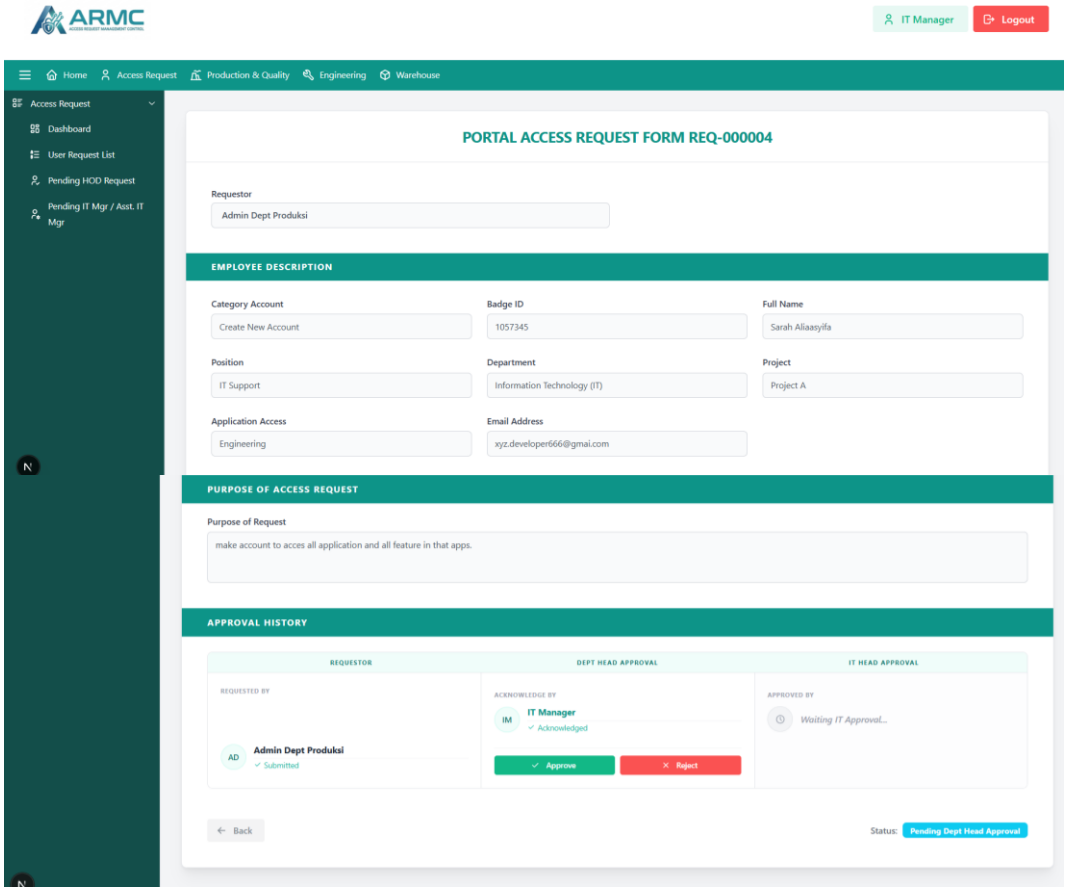
8. *Halaman Pending Head of Department Requestor*



Gambar 62 Wireframe Pending Head of Department Requestor

Gambar 62 menunjukkan halaman *Pending Dept Head Approval List* khusus untuk peran *Head of Department* yang berisi daftar pengajuan akses staf departemennya untuk divalidasi dan diproses melalui halaman detail.

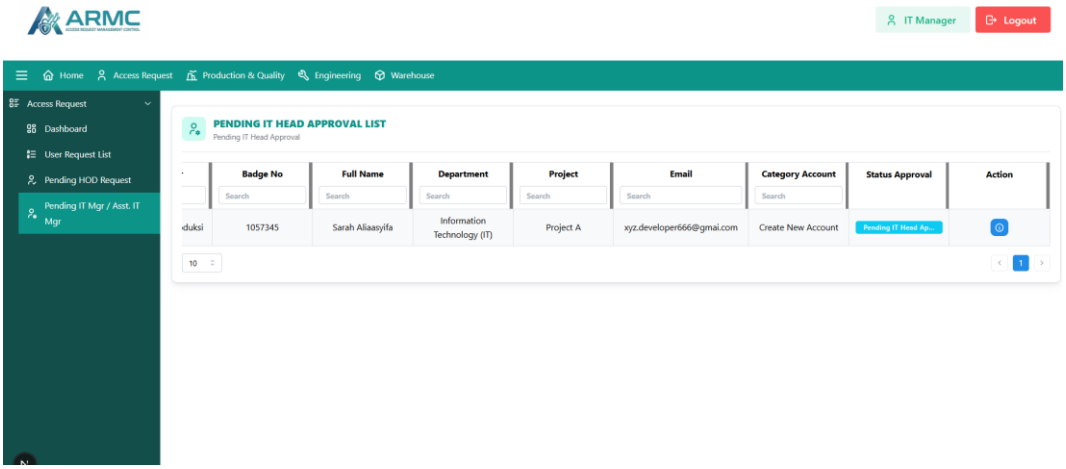
9. *Halaman Persetujuan Head of Department Requestor*



Gambar 63 Wireframe Persetujuan Head of Department Requestor

Gambar 63 menampilkan tampilan *Access Request Form* dari sisi *Head of Department*. *Head of Department* dapat meninjau detail pengajuan akses karyawan, termasuk deskripsi tujuan pengajuan, lalu mengambil tindakan melalui opsi "*Approve*" atau "*Reject*" pada bagian alur persetujuan.

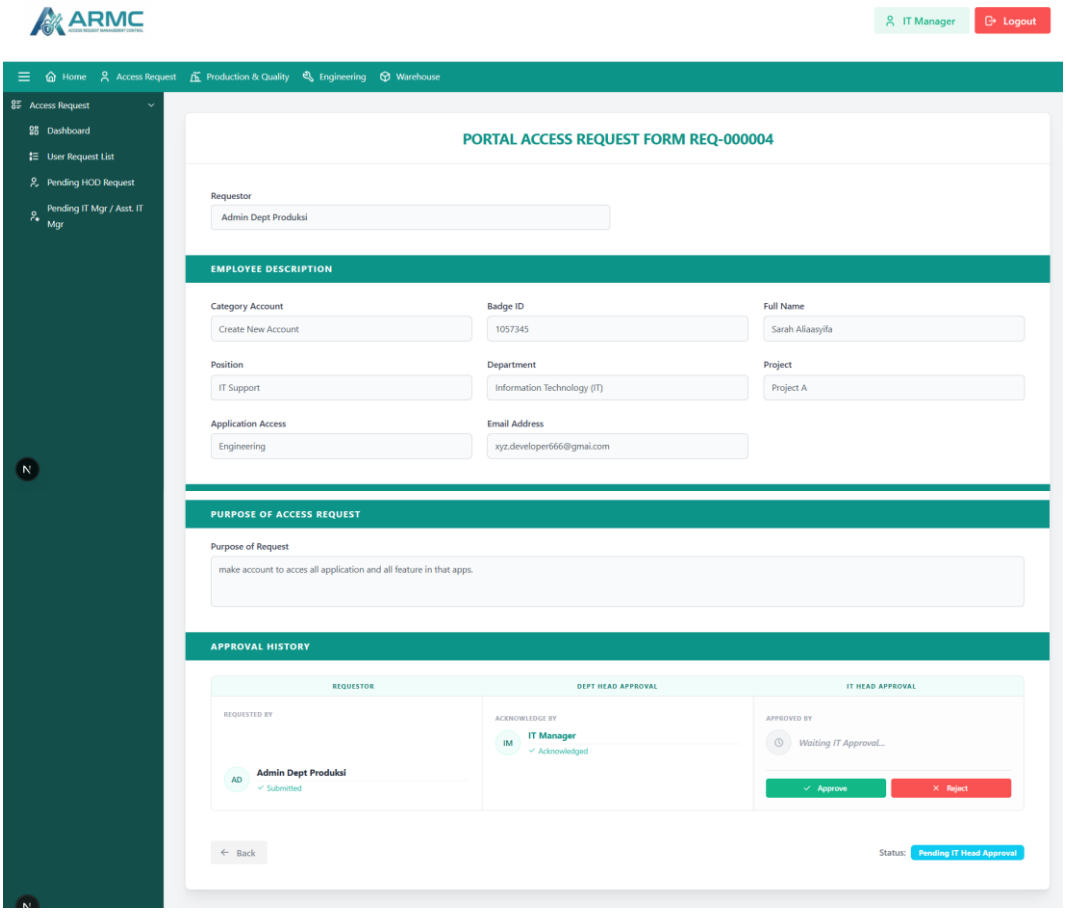
10. Halaman Pending Head of Department IT



Gambar 64 Wireframe Pending Head of Department IT

Gambar 64 menunjukkan halaman *Pending IT Head Approval List* khusus untuk peran *Head of Department* IT yang berisi daftar pengajuan akses semua departemen untuk divalidasi dan diproses melalui halaman detail.

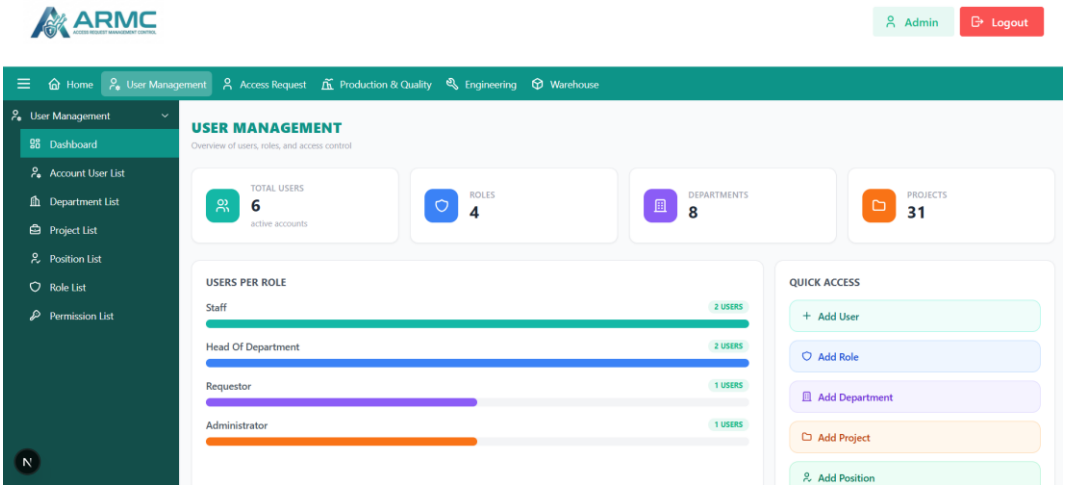
11. Halaman Persetujuan Head of Department IT



Gambar 65 Wireframe Persetujuan Head of Department IT

Gambar 65 menjelaskan tampilan *Access Request Form* dari sisi *Head of Department* IT. Pada tahap ini, pengajuan telah disetujui oleh Head of Department Requestor, sehingga *Head of Department* IT dapat melakukan tinjauan akhir dan mengambil tindakan melalui tombol "*Approve*" atau "*Reject*".

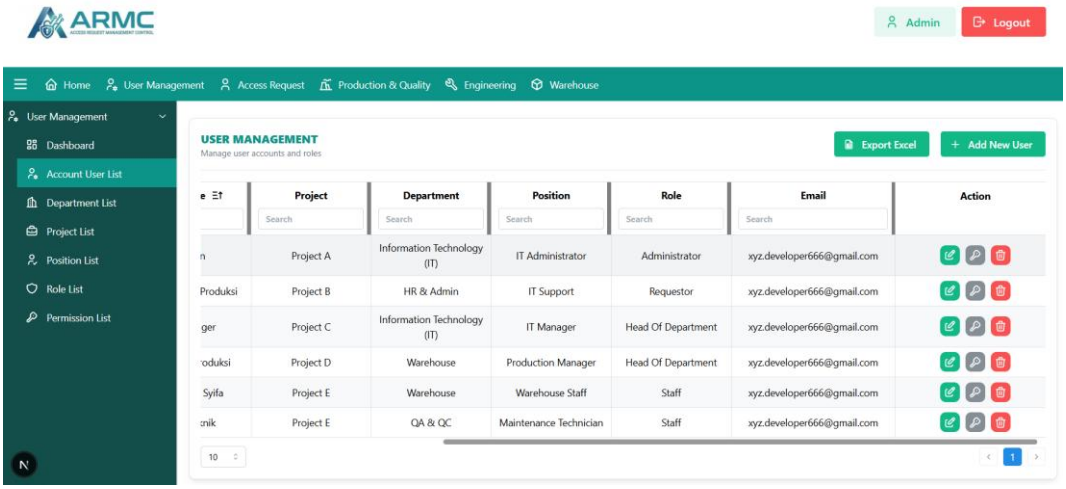
12. Halaman Dashboard User Management



Gambar 66 Wireframe Dashboard User Management

Gambar 66 menjelaskan halaman *Dashboard User Management* yang menyajikan ringkasan statistik data master seperti jumlah pengguna, peran, departemen, dan proyek. Halaman ini dilengkapi fitur *Quick Access* untuk penambahan data cepat.

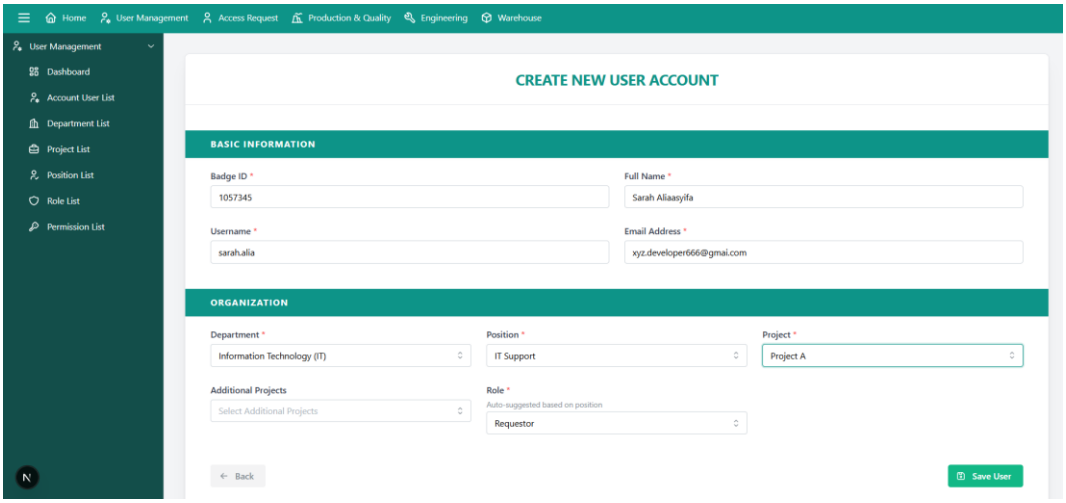
13. Halaman Account User List



Gambar 67 Wireframe Account User List

Gambar 67 menjelaskan halaman *Account User List* yang berfungsi untuk mengelola data akun pengguna dalam sistem. Halaman ini menyediakan tabel informasi detail karyawan, fitur *Export Excel* untuk laporan, serta tombol *Add New User* untuk mendaftarkan akun baru.

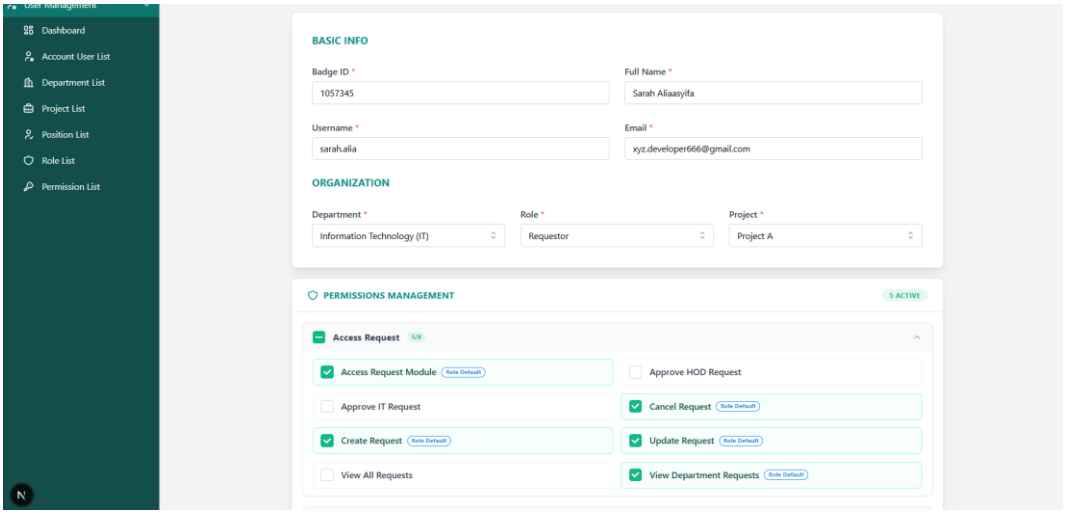
14. Halaman Add New User



Gambar 68 Wireframe Add New User

Gambar 68 menjelaskan halaman *Create New User Account* untuk pendaftaran akun baru melalui pengisian data *Basic Information* dan *Organization*. Administrator dapat menetapkan identitas serta peran pengguna sebelum menyimpan data dengan tombol *Save User.*

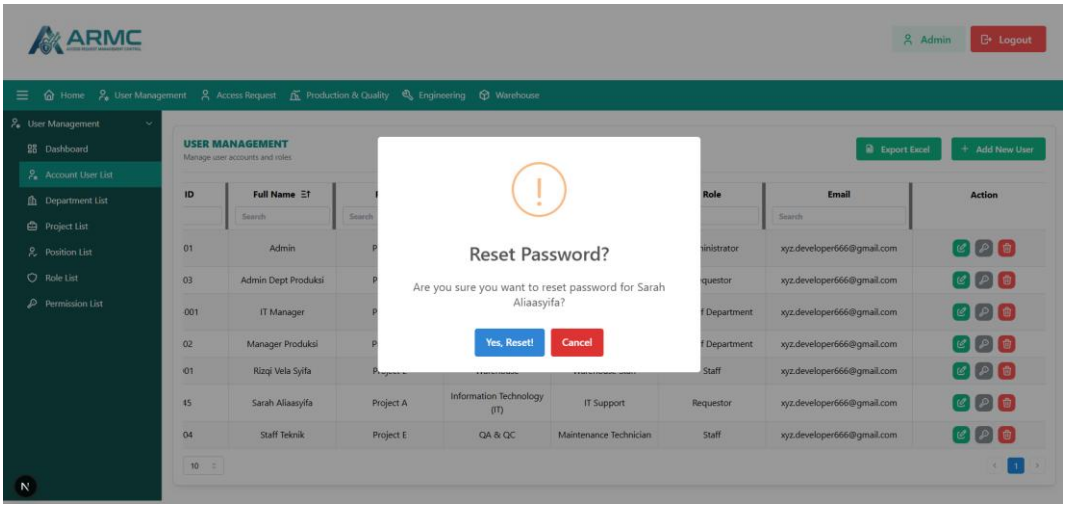
15. Halaman Edit User



Gambar 69 Wireframe Edit User

Gambar 69 menjelaskan halaman *Edit User* yang memungkinkan Administrator memperbarui data profil serta menambahkan *Permission* tambahan secara spesifik untuk pengguna tersebut.

16. Halaman Administrator Reset Password



Gambar 70 Wireframe Administrator Reset Password

Gambar 70 menjelaskan tampilan *Pop-up Reset Password* yang muncul sebagai konfirmasi saat Administrator ingin mengatur ulang kata sandi pengguna. Admin dapat memilih *Yes, Reset!* untuk memproses pengaturan ulang atau *Cancel* untuk membatalkan tindakan tersebut.

17. Halaman Export Excel

ARMC SYSTEM - USER ACCOUNT LIST

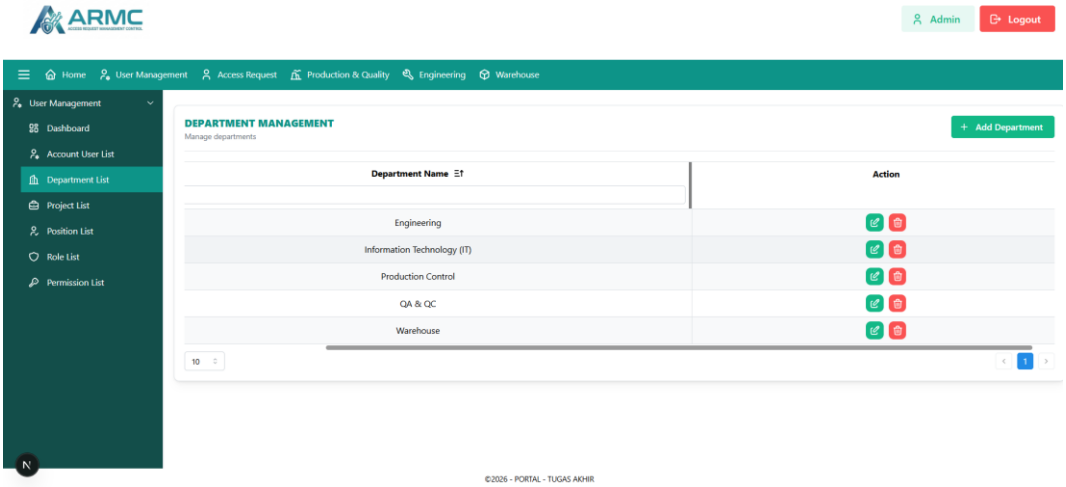
Generated: 5/13/2026, 8:02:17 PM | Total Records: 7

1	admin.it	TEST-001	Admin	Project A	Information Technology (IT)	IT Administrator	Administrator
2	manager	IT-MGR-001	IT Manager	Project C	Information Technology (IT)	IT Manager	Head Of Department
3	manager.project	EMP-002	Manager Produksi	Project D	Warehouse	Production Manager	Head Of Department
4	admin.dept	EMP-003	Admin Dept Produksi	Project B	HR & Admin	IT Support	Requestor
5	staff	EMP-004	Staff Teknik	Project E	QA & QC	Maintenance Technician	Staff
6	vela	Test-0001	Rizqi Vela Syifa	Project E	Warehouse	Warehouse Staff	Staff
7	sarah.alia	1057345	Sarah Aliaasyifa	Project A	Information Technology (IT)	IT Support	Requestor

Gambar 71 Export Excel

Gambar 71 sistem menyediakan fitur *Export to Excel* yang memungkinkan Administrator mengunduh seluruh data daftar akun pengguna

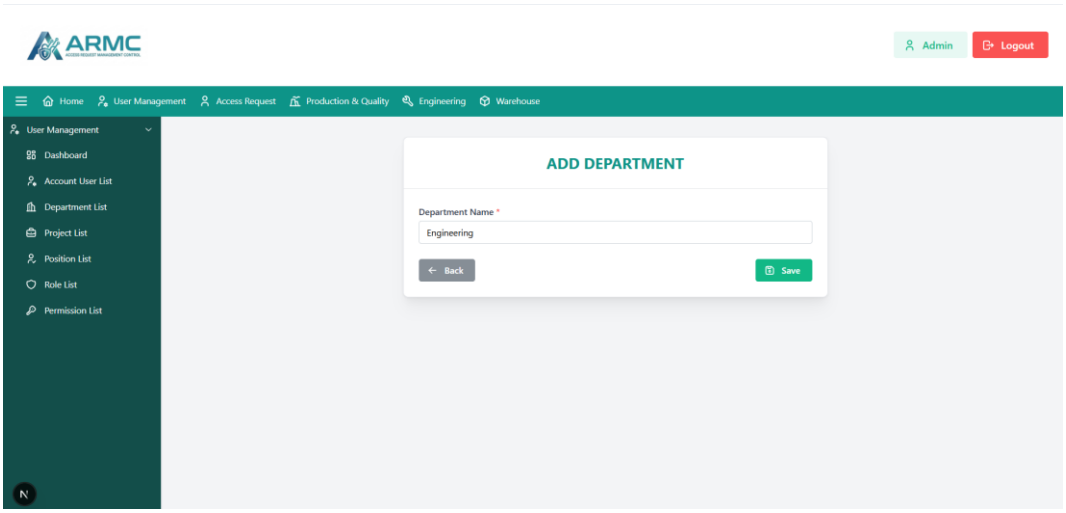
18. Halaman Department List



Gambar 72 Implementasi Department List

Gambar 72 menjelaskan halaman *Department Management* yang digunakan Administrator untuk mengelola departemen. Halaman ini menampilkan daftar departemen terdaftar serta menyediakan tombol *Add Department* untuk menambah data baru.

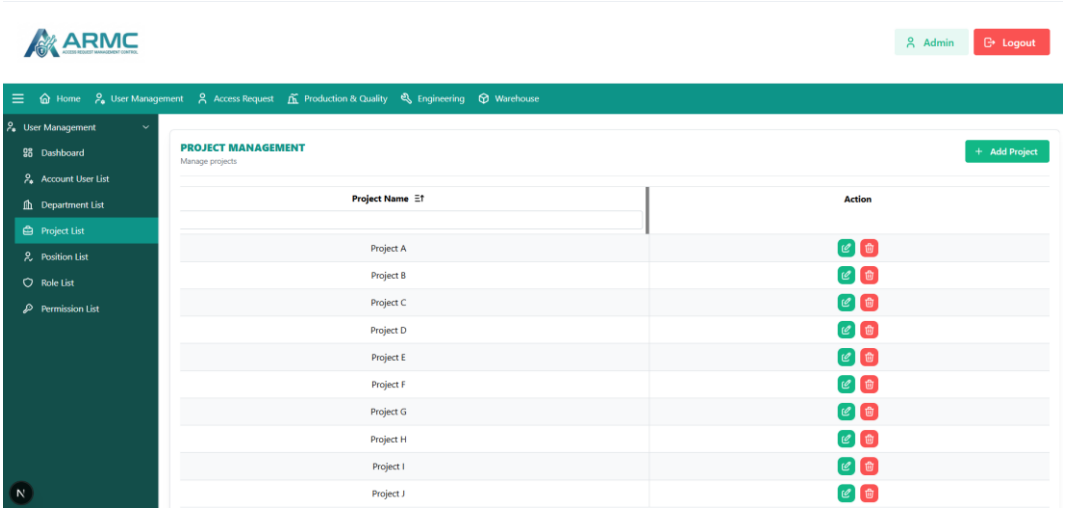
19. Halaman Add Department



Gambar 73 Implementasi Add Department

Gambar 73 memiliki kolom input *Department Name* serta tombol *Save* untuk menyimpan data dan tombol *Back* untuk kembali ke halaman daftar.

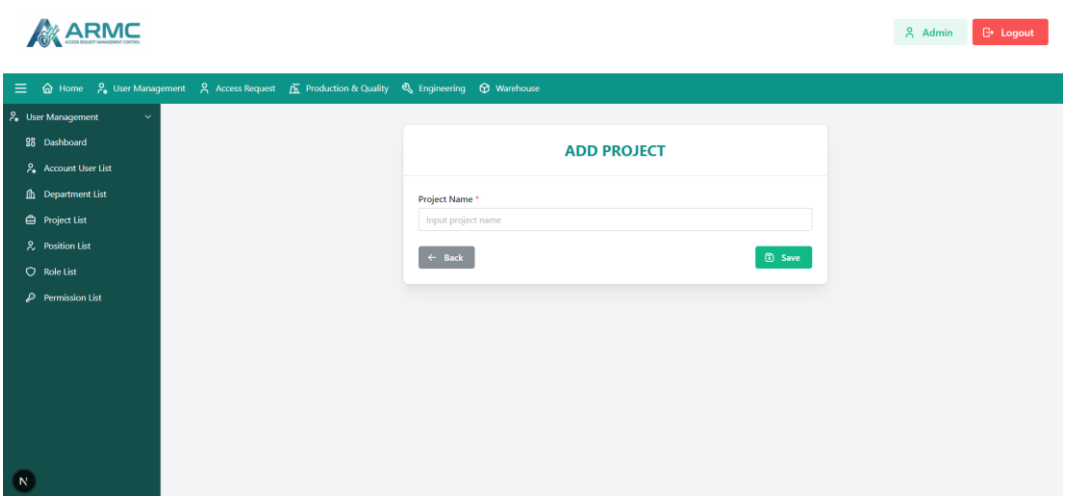
20. Halaman Project List



Gambar 74 Implementasi Project List

Gambar 74 Administrator dapat menambah data melalui tombol *Add Project* serta mengelola proyek yang ada menggunakan fitur pada kolom *Action*.

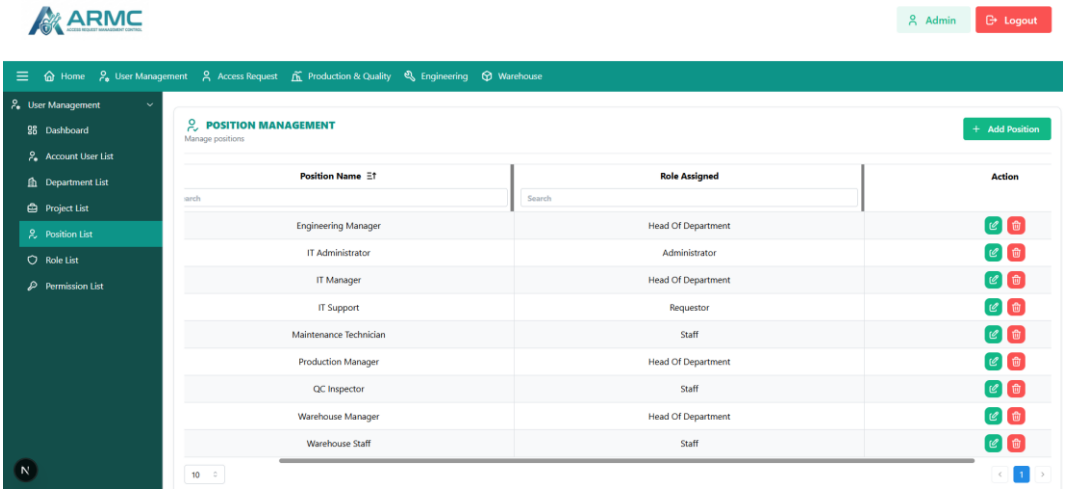
21. Halaman Add Project



Gambar 75 Implementasi Add Project

Gambar 75 memiliki kolom input *Project Name* serta tombol *Save* untuk menyimpan data dan tombol *Back* untuk kembali ke halaman daftar.

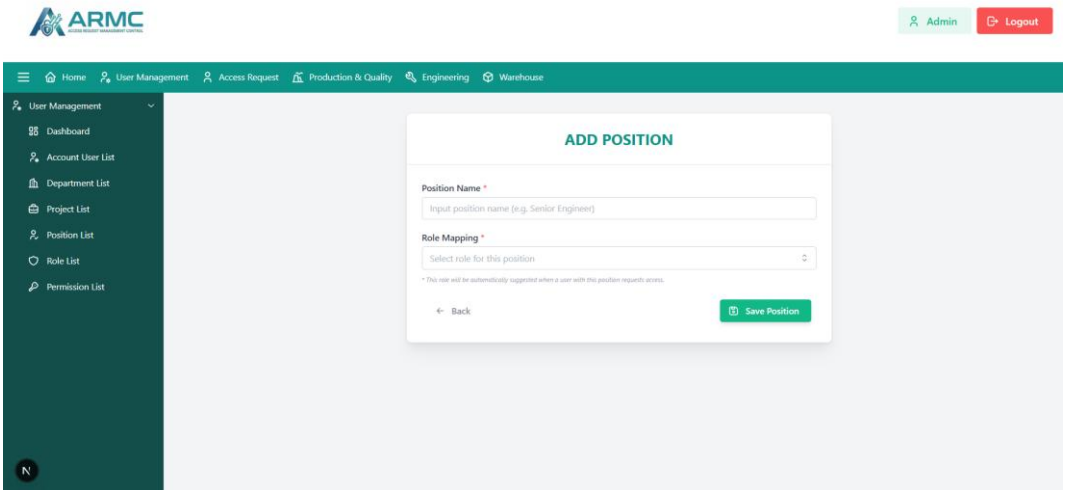
22. Halaman Position List



Gambar 76 Implementasi Position List

Gambar 76 menjelaskan halaman *Position Management* yang menampilkan daftar jabatan beserta peran yang ditetapkan dalam sistem. Administrator dapat mendaftarkan jabatan baru melalui tombol *Add Position* serta mengelola data yang ada melalui fitur pada kolom *Action*.

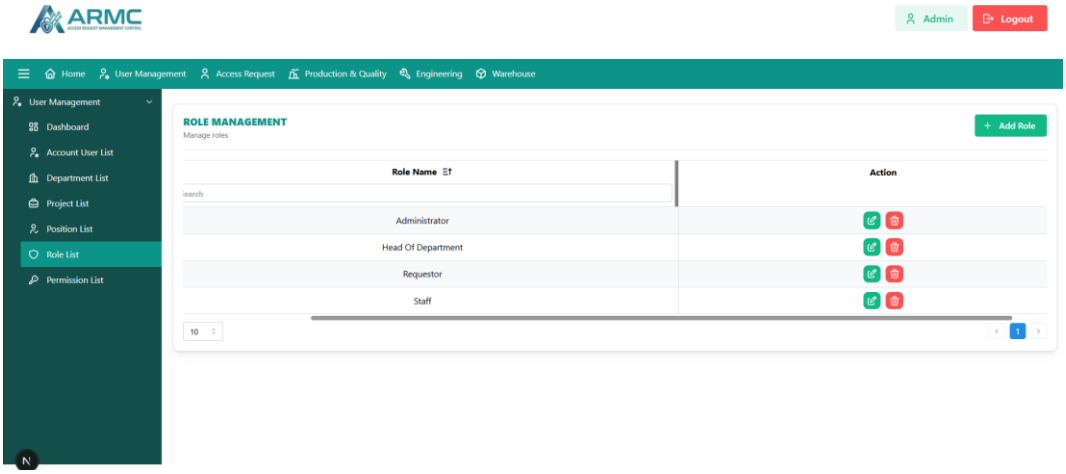
23. Halaman Add Position



Gambar 77 Implementasi Add Position

Gambar 78 memiliki kolom input *Position Name* serta *Mapping Role* untuk menghubungkan jabatan dengan peran tertentu.

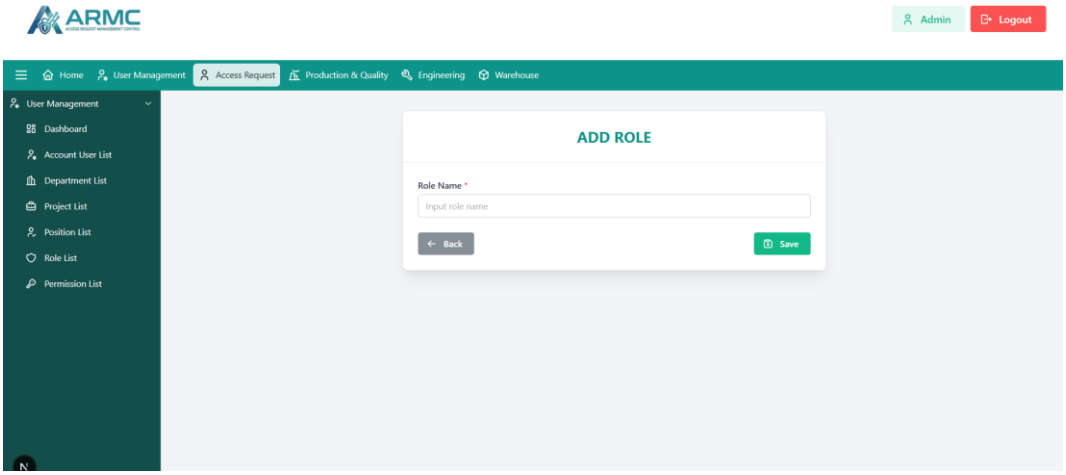
24. Halaman Role List



Gambar 78 Implementasi Role List

Gambar 78 menjelaskan halaman *Role Management* yang menampilkan daftar peran (*role*) yang tersedia dalam sistem. Administrator dapat menambahkan peran baru melalui tombol *Add Role* serta mengelola data yang ada menggunakan fitur pada kolom *Action*.

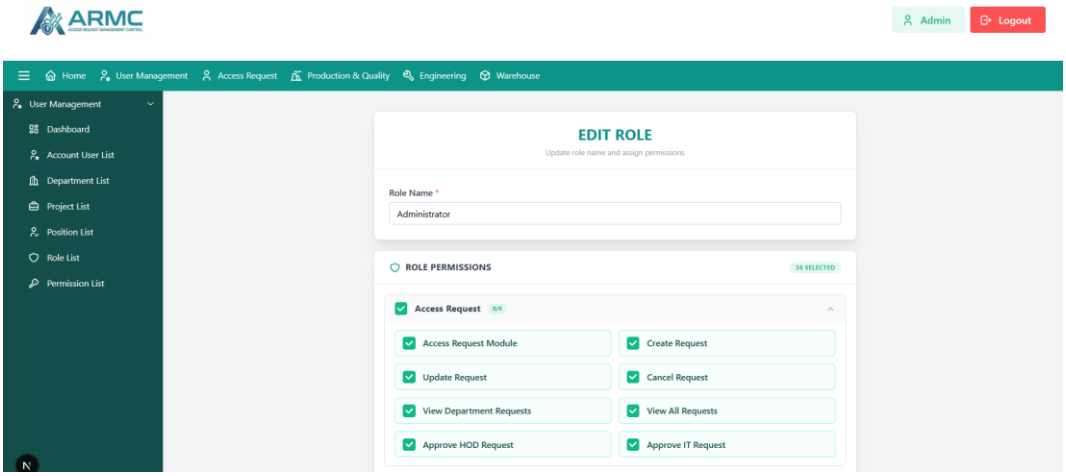
25. Halaman Add Role



Gambar 79 Implementasi Add Role

Gambar 79 memiliki kolom input *Role Name* untuk mendaftarkan nama peran baru ke dalam sistem. Terdapat juga tombol *Save* untuk menyimpan data dan tombol *Back* untuk kembali ke halaman daftar peran.

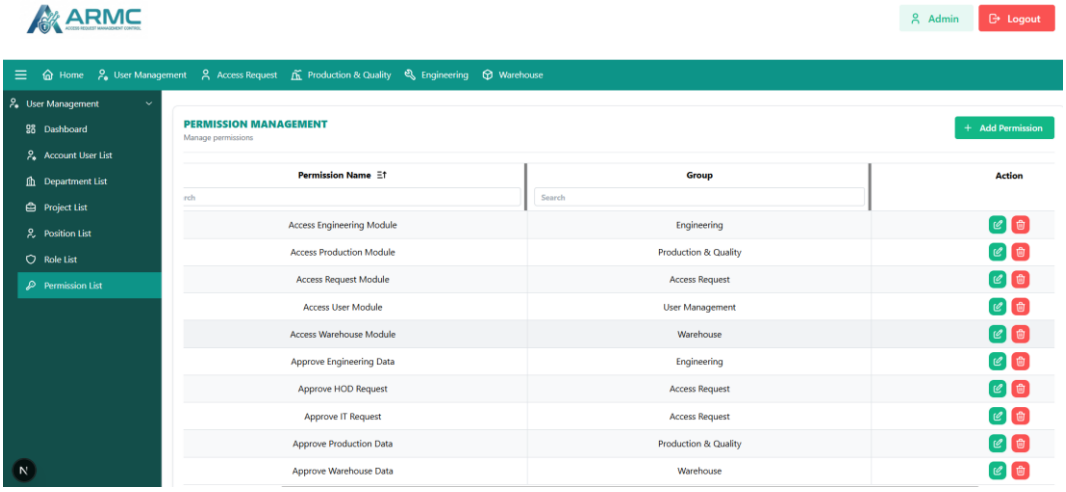
26. Halaman Edit Role



Gambar 80 Implementasi Edit Role

Gambar 80 menjelaskan halaman *Edit Role* yang digunakan untuk memperbarui nama peran serta mengatur *Role Permission*

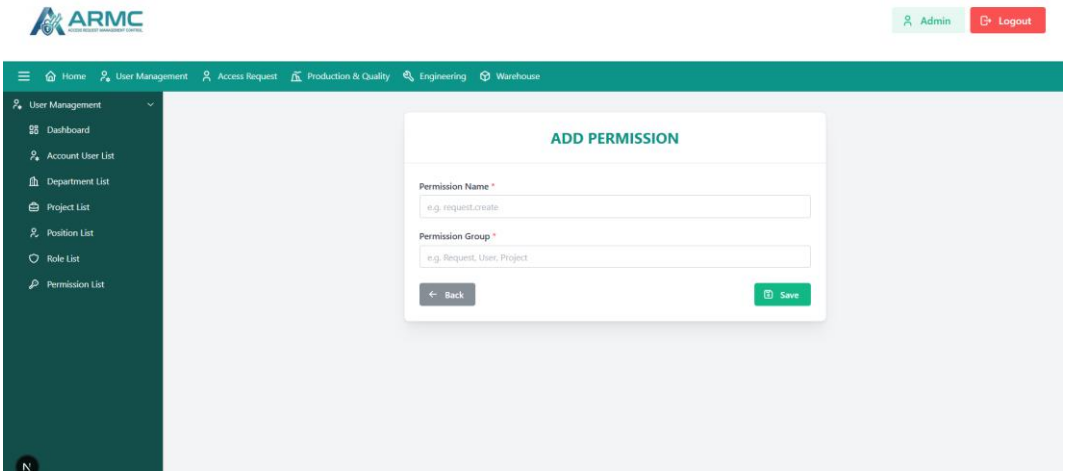
27. Halaman Permission List



Gambar 81 Implementasi Permission List

Gambar 81 menjelaskan halaman *Permissions Management* yang menampilkan daftar hak akses spesifik beserta grup menunya dalam sistem. Administrator dapat menambah hak akses baru melalui tombol *Add Permission* serta mengelola data yang ada melalui kolom *Action*.

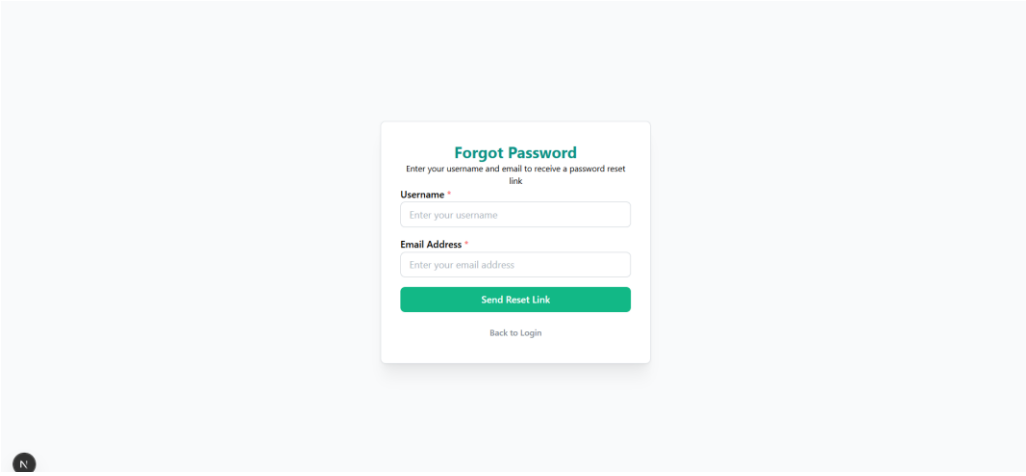
28. Halaman Add Permission



Gambar 82 Implementasi Add Permission

Gambar 82 memiliki kolom input *Permission Name* dan *Permission Group* untuk mendaftarkan hak akses baru berdasarkan kategori menunya.

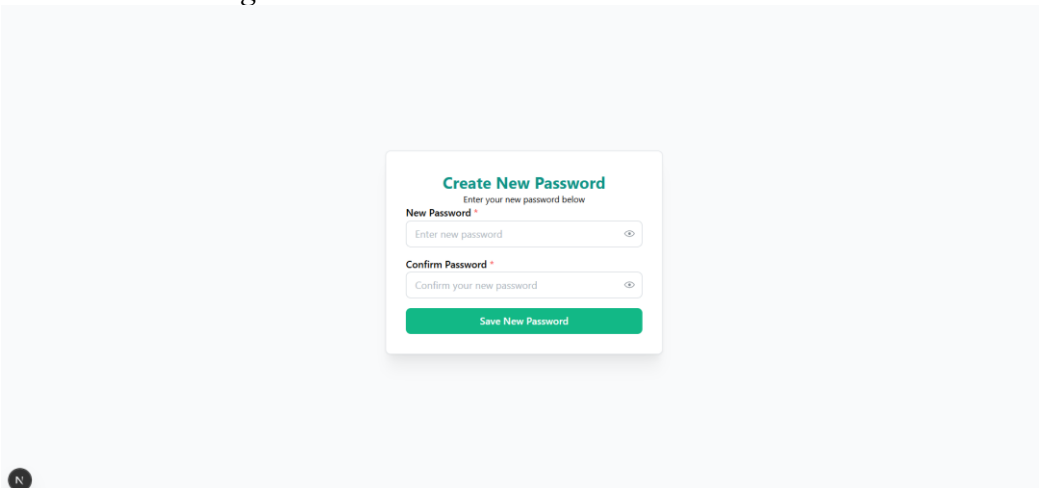
29. Halaman Forgot Password



Gambar 83 Implementasi Forgot Password

Gambar 83 menjelaskan halaman *Forgot Password* untuk pemulihan akun melalui input *Username* dan *Email*. Pengguna dapat mengeklik *Send Reset Link* untuk menerima tautan pengaturan ulang kata sandi atau memilih *Back to Login* untuk kembali.

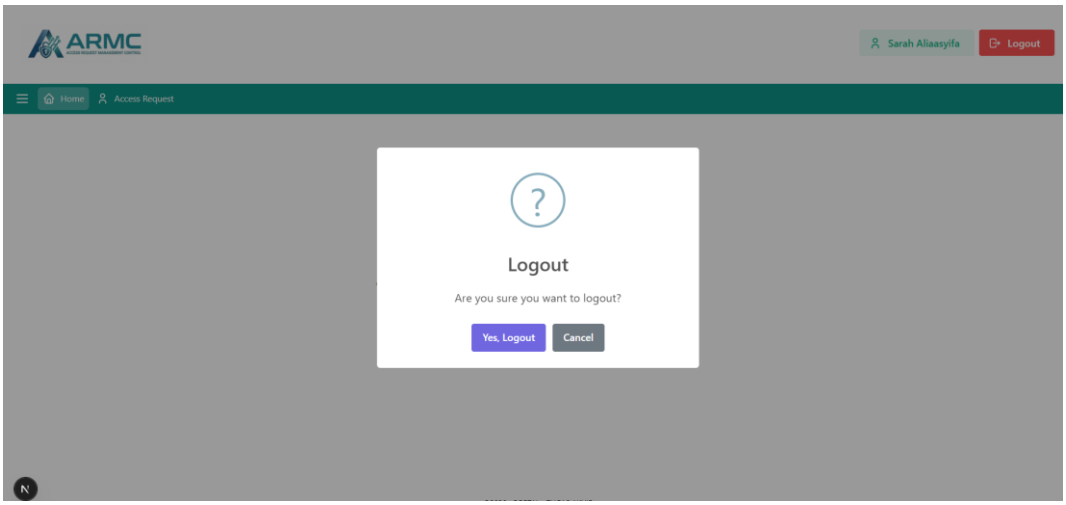
30. Halaman Mengatur Kata Sandi



Gambar 84 Implementasi Mengatur Kata Sandi

Gambar 84 pengguna perlu menginput kata sandi baru pada kolom *New Password* dan melakukan verifikasi di kolom *Re-type Password* sebelum menyimpannya melalui tombol *Save New Password*.

31. Halaman Logout



Gambar 85 Implementasi Logout

Gambar 85 menjelaskan tampilan *Pop-up Logout* yang muncul sebagai konfirmasi keamanan saat pengguna mengeklik tombol keluar. Pengguna dapat memilih tombol *Yes*, *Logout* untuk mengakhiri sesi atau *Cancel* untuk kembali ke halaman sebelumnya.

4.2 Pengujian User Acceptance Testing (UAT)

Pengujian *User Acceptance Testing* (UAT) telah dilaksanakan untuk memverifikasi bahwa sistem berjalan sesuai dengan kebutuhan pengguna dan spesifikasi yang telah ditentukan. Pengujian dilakukan oleh pihak terkait dengan detail pelaksanaan sebagai berikut:

Tanggal : Selasa, 31 Maret 2026

Tempat : PT. XYZ

Penguji : Ari Kurniawan, IT Programmer

Tabel 4. 1 User Acceptance Testing

No	Skenario Pengujian	Cara Pengujian	Ekspektasi Hasil	Hasil Aktual	Status
1	*Login* dengan data valid	Masukkan *username* dan *password* yang terdaftar, kemudian klik tombol *Login*.	Sistem berhasil menampilkan halaman *dashboard* sesuai dengan *role* pengguna.	Sesuai	Pass
2	*Login* dengan kredensial yang salah	Masukkan *username* yang terdaftar namun gunakan *password* yang salah, lalu klik *Login*.	Sistem menolak akses dan menampilkan pesan *username* atau *password* tidak valid.	Sesuai	Pass
3	*Logout* dari sistem	Klik menu *Logout* pada halaman sistem.	Pengguna berhasil keluar dari sistem dan diarahkan ke halaman *Login*.	Sesuai	Pass
4	Requestor membuat *Request* akun baru dengan data lengkap	*Login* sebagai Requestor, klik menu *Create Request*, isi seluruh data termasuk tipe *request*, *position*, dan data akun, kemudian klik *Save*.	Sistem menyimpan Request dengan status *Pending*.	Sesuai	Pass

No	Skenario Pengujian	Cara Pengujian	Ekspektasi Hasil	Hasil Aktual	Status
5	Requestor membuat *Request* dengan data tidak lengkap	*Login* sebagai Requestor, klik *Create Request*, kosongkan salah satu kolom wajib, kemudian klik *Save*.	Sistem menggagalkan penyimpanan dan menampilkan pesan *Please fill in all required fields.*	Sesuai	Pass
6	Requestor mengubah *Request* yang masih *Pending*	Login sebagai Requestor, pilih *Request* berstatus *Pending*, klik Edit, ubah data, kemudian klik *Update*.	Data Request berhasil diperbarui dengan status tetap *Pending*.	Sesuai	Pass
7	Requestor mengubah *Request* yang sudah disetujui *Head of Department*	*Login* sebagai Requestor, pilih *Request* yang sudah disetujui *Head of Department* Requestor, kemudian klik Edit.	Sistem menolak aksi edit.	Sesuai	Pass
8	*Head of Department* Requestor melihat detail *Request*	Login sebagai *Head of Department* Requestor, klik menu *Penging Hod Request*, pilih salah satu *Request*, klik Details.	Sistem menampilkan detail lengkap *Request* dari *Department* yang sama.	Sesuai	Pass
9	*Head of Department* Requestor menyetujui *Request*	Login sebagai *Head of Department* Requestor, buka detail *Request* berstatus *Pending*, pilih opsi *Approved*.	Status *Request* berhasil diperbarui menjadi *Approved* dan diteruskan ke *Head of Department* IT.	Sesuai	Pass
10	*Head of Department*	Login sebagai *Head of*	Status *Request* berhasil	Sesuai	Pass

No	Skenario Pengujian	Cara Pengujian	Ekspektasi Hasil	Hasil Aktual	Status
	Requestor menolak *Request* dengan catatan	*Department* Requestor, buka detail *Request*, pilih opsi *Rejected*, isi catatan alasan, klik *Save*.	diperbarui menjadi *Rejected* dan catatan tersimpan.		
11	*Head of Department* Requestor menolak *Request* tanpa catatan	*Login* sebagai *Head of Department* Requestor, pilih opsi *Rejected*, kosongkan kolom catatan, klik *Save*.	Sistem menolak perubahan status dan meminta catatan alasan diisi.	Sesuai	Pass
12	*Head of Department* IT melihat detail *Request*	Login sebagai *Head of Department*, klik menu *Pending It Mgr/Asst IT Mgr*, pilih salah satu *Request*, klik *Details*.	Sistem menampilkan detail lengkap seluruh *Request* dari semua *Department*.	Sesuai	Pass
13	*Head of Department* IT menyetujui *Request*	Login sebagai *Head of Department* IT, buka detail *Request* berstatus *Pending*, pilih opsi *Approved*.	Status *Request* berhasil diperbarui menjadi *Approved*.	Sesuai	Pass
14	*Head of Department* IT menolak *Request* dengan catatan	Login sebagai *Head of Department* IT, buka detail *Request*, pilih opsi *Rejected*, isi catatan alasan, klik *Save*.	Status *Request* berhasil diperbarui menjadi *Rejected* dan catatan tersimpan.	Sesuai	Pass
15	*Head of Department* IT menolak	Login sebagai *Head of Department* IT, pilih opsi	Sistem menolak perubahan status dan meminta	Sesuai	Pass

No	Skenario Pengujian	Cara Pengujian	Ekspektasi Hasil	Hasil Aktual	Status
	Request tanpa catatan	*Rejected*, kosongkan kolom catatan, klik *Save*.	catatan alasan diisi.		
16	Administrator membuat akun pengguna baru dengan data lengkap	*Login* sebagai Administrator, buka halaman *User Management*, klik *Add New*, isi seluruh data termasuk *position* dan *role*, klik *Save*.	Akun pengguna baru berhasil dibuat dan tersimpan di sistem.	Sesuai	Pass
17	Administrator membuat akun dengan data tidak lengkap	*Login* sebagai Administrator, klik *Add New*, kosongkan salah satu kolom wajib, klik *Save*.	Sistem menggagalkan penyimpanan dan menampilkan pesan validasi.	Sesuai	Pass
18	Administrator mengedit akun pengguna	*Login* sebagai Administrator, pilih akun dari daftar, klik Edit, ubah data yang diperlukan, klik *Save*.	Data akun pengguna berhasil diperbarui.	Sesuai	Pass
19	Administrator menghapus akun pengguna	*Login* sebagai Administrator, pilih akun dari daftar, klik Hapus, konfirmasi penghapusan.	Akun pengguna berhasil dihapus dari sistem.	Sesuai	Pass
20	Administrator mereset kata sandi pengguna	*Login* sebagai Administrator, pilih akun pengguna, klik Reset Password,	*Reset link* terkirim ke email	Sesuai	Pass
21	Administrator mengunduh	*Login* sebagai Administrator,	Sistem berhasil mengunduh	Sesuai	Pass

No	Skenario Pengujian	Cara Pengujian	Ekspektasi Hasil	Hasil Aktual	Status
	daftar akun pengguna	buka halaman *User Management*, klik tombol *Export Excel*.	daftar akun pengguna dalam format file *Excel*.		
22	Administrator membuat *Role* baru	*Login* sebagai Administrator, buka halaman *Role* Management, klik *Add Role*, isi nama *Role* dan pilih *permission*, klik Save.	Role baru berhasil dibuat dan tersimpan di sistem.	Sesuai	Pass
23	Administrator membuat *Role* dengan nama duplikat	*Login* sebagai Administrator, klik *Add Role*, masukkan nama *Role* yang sudah ada, klik *Save*.	Sistem menolak penyimpanan dan menampilkan pesan error duplikasi.	Sesuai	Pass
24	Administrator mengedit *Role*	*Login* sebagai Administrator, pilih *Role* dari daftar, klik Edit, ubah nama atau *permission*, klik *Update*.	Data *Role* berhasil diperbarui.	Sesuai	Pass
25	Administrator menghapus *Role*	*Login* sebagai Administrator, pilih *Role* dari daftar, klik *Delete*, konfirmasi penghapusan.	*Role* berhasil dihapus dari sistem.	Sesuai	Pass
27	Administrator membuat *Permission* baru	*Login* sebagai Administrator, buka halaman *Permission Management*, klik *Add Permission*, isi nama	*Permission* baru berhasil dibuat dan tersimpan di sistem.	Sesuai	Pass

No	Skenario Pengujian	Cara Pengujian	Ekspektasi Hasil	Hasil Aktual	Status
		Permission, klik *Save*.			
28	*Administrator* membuat *Permission* dengan nama duplikat	*Login* sebagai Administrator, klik *Add Permission*, masukkan nama *Permission* yang sudah ada, klik *Save*.	Sistem menolak penyimpanan dan menampilkan pesan error duplikasi.	Sesuai	Pass
29	Administrator mengedit *Permission*	*Login* sebagai Administrator, pilih *Permission* dari daftar, klik Edit, ubah nama, klik *Update*.	Data *Permission* berhasil diperbarui.	Sesuai	Pass
30	Administrator menghapus Permission	*Login* sebagai Administrator, pilih Permission dari *daftar*, klik *Delete*, konfirmasi penghapusan.	*Permission* berhasil dihapus dari sistem.	Sesuai	Pass
31	Administrator membuat *Department* baru	*Login* sebagai Administrator, buka halaman *Department Management*, klik *Add Department*, isi nama *Department*, klik *Save*.	*Department* baru berhasil dibuat dan tersimpan di sistem.	Sesuai	Pass
32	Administrator membuat *Department* dengan nama duplikat	*Login* sebagai Administrator, klik *Add Department*, masukkan nama *Department* yang sudah ada, klik *Save*.	Sistem menolak penyimpanan dan menampilkan pesan *error* duplikasi.	Sesuai	Pass

No	Skenario Pengujian	Cara Pengujian	Ekspektasi Hasil	Hasil Aktual	Status
33	Administrator mengedit *Department*	*Login* sebagai Administrator, pilih *Department* dari daftar, klik Edit, ubah nama, klik *Update*.	Data *Department* berhasil diperbarui.	Sesuai	Pass
34	Administrator menghapus *Department*	*Login* sebagai Administrator, pilih *Department* dari daftar, klik *Delete*, konfirmasi penghapusan.	*Department* berhasil dihapus dari sistem.	Sesuai	Pass
35	Administrator membuat *Project* baru	*Login* sebagai Administrator, buka halaman *Project Management*, klik *Add Project*, isi nama *Project* dan pilih *Project*, klik *Save*.	*Project* baru berhasil dibuat dan tersimpan di sistem.	Sesuai	Pass
36	Administrator membuat *Project* dengan nama duplikat	*Login* sebagai Administrator, klik *Add Project*, masukkan nama *Project* yang sudah ada, klik *Save*.	Sistem menolak penyimpanan dan menampilkan pesan *error* duplikasi.	Sesuai	Pass
37	Administrator mengedit *Project*	*Login* sebagai Administrator, pilih *Project* dari daftar, klik Edit, ubah nama, klik *Update*.	Data *Project* berhasil diperbarui.	Sesuai	Pass
38	Administrator menghapus *Project*	*Login* sebagai Administrator, pilih *Project* dari daftar, klik *Delete*, konfirmasi penghapusan.	*Project* berhasil dihapus dari sistem.	Sesuai	Pass

No	Skenario Pengujian	Cara Pengujian	Ekspektasi Hasil	Hasil Aktual	Status
39	Administrator membuat *Position* baru	*Login* sebagai Administrator, buka menu *Position Management*, klik *Add Position*, isi nama *Position*, pilih peran pada *Role Mapping,* klik *Save*.	*Position* dan *Role* baru berhasil dibuat dan tersimpan di sistem.	Sesuai	Pass
40	Administrator membuat *Position* dengan nama duplikat	*Login* sebagai Administrator, klik *Add Position*, masukkan nama *Position* yang sudah ada, klik *Save*.	Sistem menolak penyimpanan dan menampilkan pesan error duplikasi.	Sesuai	Pass
41	Administrator mengedit *Position*	*Login* sebagai Administrator, pilih *Position* dari daftar, klik Edit, ubah nama, klik *Update*.	Data *Position* berhasil diperbarui.	Sesuai	Pass
42	Administrator menghapus *Position*	*Login* sebagai Administrator, pilih *Position* dari daftar, klik *Delete*, konfirmasi penghapusan.	*Position* berhasil dihapus dari sistem.	Sesuai	Pass

BAB V KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil pengembangan dan pengujian yang telah dilakukan, berikut ini kesimpulan dari proyek akhir pengembangan fitur manajemen hak akses pengguna pada sistem internal PT. XYZ menggunakan metode *Role-Based Access Control* (RBAC)

1. Sistem berhasil dikembangkan menggunakan *Next.js*, *NestJS*, dan *PostgreSQL*, serta mampu mengelola proses *request*, persetujuan berlapis, dan pembuatan akun pengguna secara digital menggantikan proses manual sebelumnya.
2. Penerapan RBAC berhasil mendefinisikan *role* berdasarkan struktur organisasi PT. XYZ, sehingga setiap pengguna memperoleh hak akses sesuai fungsi dan tanggung jawabnya dan risiko *over-privilege* dapat diminimalisasi.
3. Hasil *User Acceptance Testing* (UAT) menunjukkan seluruh 42 skenario pengujian berstatus *Pass*, membuktikan sistem telah memenuhi kebutuhan fungsional yang dirancang.

5.2 Saran

Setelah melaksanakan pengembangan sistem ini, penulis memiliki beberapa saran untuk pengembangan lebih lanjut:

1. Sistem saat ini hanya mencakup karyawan internal PT. XYZ, sehingga disarankan agar sistem diperluas untuk mengakomodasi pengguna eksternal seperti subkontraktor dan klien.
2. Disarankan agar sistem dilengkapi dengan fitur *audit log* atau riwayat aktivitas yang mencatat seluruh tindakan pengguna di dalam sistem. Hal ini penting agar setiap perubahan hak akses dapat terlacak dengan jelas dan dipertanggungjawabkan.
3. Sistem selanjutnya disarankan untuk dilengkapi dengan fitur otomatisasi pembuatan (*create account*). Fitur ini nantinya dapat

menarik data secara langsung dari *master data* atau formulir *request* yang sudah disetujui. Hal ini bertujuan agar Administrator tidak perlu repot menginput data ulang secara manual, sehingga proses operasional menjadi jauh lebih efisien dan terhindar dari risiko *human error*.

DAFTAR PUSTAKA

Zikra, A. A., Darnilasari, A., Yanuary, R., & Sari, K. (2025). *Design of web-based project management system with multi-level role-based access control*. *Journal of Computer Science and Informatics Engineering, 4*(3), 201–215. https://doi.org/10.55537/cosie.v4i3.1179

Setiawan, H., Hanaputra, R. R., Anggoman, C. R., Putra, I. G. M., Wijayanti, R. A., & Hindami, A. L. A. (2024). *Rancang bangun secure document management system (DBMS) menggunakan metode Agile-SSDLC.* INSERT: Information System and Emerging Technology Journal, 5(1), 73-84. https://doi.org/10.23887/insert.v5i1.75244

Putrawan, P., & Muliani Harahap, A. (2024). *Implementasi metode role-based access control pada aplikasi e-raport di MIN 15 Langkat berbasis Android. Jurnal Teknologi Informasi dan Ilmu Komputer (JTISUT),* 9(1), 1-12. https://ejournal.ust.ac.id/index.php/JTIUST/article/view/3815/2924?utm_source=chatgpt.com

Prasetia, Y. A. (2024). *Role-Based Access Control (RBAC) untuk sistem otorisasi terpusat berbasis Flask: Studi kasus PT. XYZ* [Skripsi, Universitas Kristen Satya Wacana]. *Repository Univeristas Kristen Satya Wacana. https://repository.uksw.edu/bitstream/123456789/33562/1/T1_672020193_Judul.pdf*

Penggalih, V. G., & Silmnina, E. P. (2025). *Implementasi dashboard proyek dengan sistem role-bases access control pada PT XYZ.* JATI (Jurnal Mahasiswa Teknik Informatika), 9(2). https://doi.org/10.36040/jati.v9i2.12978

Jayanti, F. N., & Aslamiyah, S. (2024). Pengembangan Sistem Informasi Untuk Pembayaran Dengan Studi Kasus Di LKP Seiza Pesanggaran. *INTEKNA Jurnal Informasi Teknik Dan Niaga*, 24(1), 62–76. https://ejurnal.poliban.ac.id/index.php/intekna/issue/view/142

Alim, M., Rasyid Munthe, I., & Putra Juledi, A. (2024). Evaluasi Keamanan Sistem Informasi dalam Lingkungan Bisnis Digital. *Jurnal Ilmu Komputer Dan Sistem Informasi (JIKOMSI)*, 7(1), 328–332. http://repository.ulb.ac.id/id/eprint/772

Sagita, L., Nafi'ah, S. F., Risnaeni, N. A., Suhadi, A., Pratama, M. A., & Yuningrum, H. (2026). Implementasi Sistem Pengendalian Internal Sebagai Upaya Peningkatan Kinerja Pada Balatkop UKM & Kewirausahaan Provinsi Jawa Tengah. *Jurnal Ilmiah Manajemen Ekonomi Dan Akuntansi (JIMEA)*, 3(3), 124-135. https://doi.org/10.62017/jimea.v3i3.7388

Yuricha, Y., & Phan, I. K. (2023). Penerapan Role Based Access Control dalam Sistem Supply Chain Management Berbasis Cloud. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 3(2). https://doi.org/10.57152/malcom.v3i2.1259

Gumilar, M. D., & Firmansyah, B. (2023). Perancangan Sistem Pelayanan Kuliah Pengganti di Institut Bisnis dan Informatika Kosgoro 1957 Berbasis Website. *Jurnal Nasional Informatika (JUNIF)*, 3(1), 12–19. https://doi.org/10.55122/junif.v4i1.944

Vercel. (2025). *About React and Next.js.* https://nextjs.org/docs

NestJS. (2025). *NestJS documentation.* https://docs.nestjs.com/

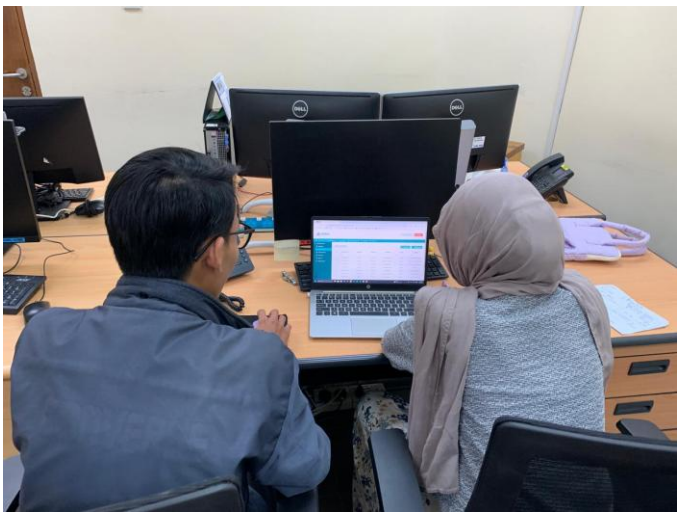
PostgreSQL Global Development Group. (2025). *PostgreSQL about.* https://www.postgresql.org/docs/

Dewi, M. M., Farida, L. D., & Nuraminudin, M. (2024). Implementasi sistem informasi antrian berbasis website dengan metodologi Scrum. *Journal of Information System Management (JOISM),* *5*(2), 233–238. https://doi.org/10.24076/joism.2024v5i2.1442

DAFTAR LAMPIRAN

Lampiran 1 Proses Requirement

Lampiran 2 Dokumen Pengujian UAT



Lampiran 3 Link Produk

Berikut link GitHub aplikasi website pembuatan dan pengelolaan *purchase request:*
Backend: https://github.com/reezqiii/armc-backend.git
Frontend: https://github.com/reezqiii/armc_frontend.git

Berikut link demo aplikasi website *Armc:*
https://youtu.be/nW8iHbwgTqA?si=GOJX48UIJjuq4Tsa