

## BAB III

### PELAKSANAAN KERJA

#### 3.1 Kedudukan dan Koordinasi

Pelaksanaan periode magang di Hypernet Technologies dengan posisi sebagai IT Application System yang dilakukan selama sekitar dua bulan setengah. Program magang ini berlangsung mulai dari tanggal 02 September 2025 hingga 19 Desember 2025, metode pelaksanaan magang di Hypernet Technologies dilakukan secara full *Work from Office* (WFO). Selama pelaksanaan magang tersebut, fokus utama dalam pelaksanaan magang ini adalah pembuatan website aplikasi untuk pemesanan ruang rapat di PT Hipernet Indodata.

Selama periode magang tersebut, mahasiswa telah mengerjakan beberapa tugas antara lain adalah desain prototipe menggunakan figma untuk masing-masing halaman website, proses pengembangan pengembangan frontend menggunakan aplikasi visual studio code dengan menggunakan bahasa HTML serta didukung dengan CSS dan JavaScript yang berfungsi untuk memberikan pengalaman UI/UX. Selain itu, menjadi bagian dari divisi IT Application juga membantu dalam perjalanan pengembangan aplikasi selain aplikasi pemesanan ruang rapat yaitu membuat desain UI/UX untuk pelanggan Hypernet Technologies. Pengalaman tersebut menjadikan mahasiswa memiliki pengetahuan baru terkait interaksi serta alur koordinasi antara perusahaan dengan pelanggan.

##### 3.2.1 Kedudukan



Gambar 3. 1 Struktur Organisasi IT Service Management

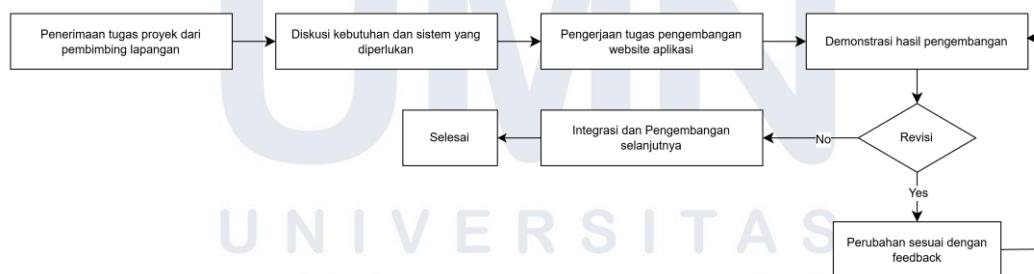
Pada Gambar 3.1 di atas merupakan struktur organisasi dari divisi *IT Service Management* (ITSM) yang dimana divisi tersebut bekerja untuk manajemen IT, dan pemantauan serta optimasi layanan IT secara keseluruhan. Selain itu, divisi ITSM memastikan layanan IT telah

tersampaikan dengan efisien sesuai dengan kebutuhan bisnis, mulai dari dukungan teknis dasar hingga strategi jangka panjang.

Mahasiswa ditempatkan pada divisi IT *Application System* di bawah pengawasan langsung oleh kepala divisi ITSM yang bertanggung jawab atas pengelolaan dan pemberi arahan atas pengembangan proyek *website* pemesanan ruang rapat ini. Struktur organisasi di atas mendukung sistem koordinasi yang terarah dan efisien, dimana setiap tugas dan tanggung jawab diberikan dari divisi ITSM hingga sampai ke divisi terkait. Selain itu, selama pelaksanaan program magang di Hypernet Technologies mahasiswa mendapatkan mentor atau *user* yang akan memberikan arahan dan tempat berdiskusi jika ada hal yang perlu ditanyakan atau menjadi

### 3.2.2 Koordinasi

Bagian ini berisikan penjelasan mengenai informasi alur koordinasi pekerjaan yang penulis lakukan di perusahaan magang. Alur koordinasi ini dibentuk setelah melewati hari pertama periode magang, karena perlu ada penyesuaian dan analisa terkait alur koordinasi yang baik agar berjalan lancar. Alur koordinasi ini dapat pula dilengkapi dengan bagan alur kerja seperti contoh di bawah ini:



Gambar 3. 2 Bagan Alur Koordinasi

Pada Gambar 3.2 di atas merupakan bagan alur koordinasi yang dilakukan selama melaksanakan program magang di Hypernet Technologies. Pada awal periode magang diberikan seluruh akses pengerjaan proyek yang dibuat dalam GitHub agar setiap proses perubahan dapat terdokumentasi dan masih memiliki *backup* jika suatu saat membutuhkan *code* yang sebelumnya.

Pengerjaan proyek pengembangan *website* aplikasi pemesanan ruang rapat ini melibatkan beberapa *stakeholder* yaitu kepala divisi ITSM hingga karyawan dari IT *Application System* selaku sebagai *user* dari mahasiswa magang. Tahap awal yang dilakukan adalah menentukan kebutuhan yang perlu ada pada aplikasi tersebut serta alur sistem yang akan dibuat. Pemberian akses tugas menggunakan GitHub untuk menyimpan dan melacak perubahan kode yang sudah dibuat, selain itu fungsi utamanya adalah untuk memfasilitasi kolaborasi tim, mengelola, dan melacak perubahan kode.

Selama pengerjaan proyek tersebut, mahasiswa melakukan koordinasi dengan *user* selaku yang memberikan proyek tersebut. Pengerjaan proyek tersebut menggunakan *tools* visual studio code dengan menggunakan bantuan *framework bootstrap* untuk memudahkan pembuatan dasar struktur *website*. Sebelum mahasiswa membuat struktur *website* dan melakukan desain UI/UX terdapat proses *brainstorming* bersama tim. Hal ini berfungsi untuk menentukan kebutuhan yang perlu tersedia seperti fitur serta aktivitas yang bisa dilihat dan dilakukan pada pengembangan *website* aplikasi pemesanan ruang rapat.

### 3.2 Tugas yang Dilakukan

Selama proses pelaksanaan kerja magang di perusahaan PT HiperNet Indodata fokus atau tugas utamanya adalah pada pengembangan *frontend website* aplikasi untuk pemesanan ruang rapat. Tugas ini bertujuan untuk meningkatkan efisiensi serta kemudahan karyawan dalam memesan ruang rapat pada perusahaan. Sebelum pelaksaan proyek ini, pemesanan ruang rapat pada perusahaan PT HiperNet Indodata melalui pengisian *form* yang setelahnya mengkonfirmasi langsung kepada resepsionis. Hal ini menjadi faktor utama dalam pengembangan *website* aplikasi ini, karena ingin beralih dari sistem yang masih manual menjadi digital sesuai dengan bidang perusahaan yang sudah berbasis teknologi.

Tahap awal pembuatan *website* aplikasi ini adalah diskusi terkait proses dan hal apa saja yang akan ada pada *website* aplikasi tersebut, lalu penentuan bisnis prosesnya dan yang terakhir adalah pengembangan lebih lanjut. Pembuatan *website*

aplikasi menggunakan *framework bootstrap* untuk memudahkan serta mempercepat pengembangan situs web. *Bootstrap* menyediakan kumpulan kode HTML, CSS, dan JavaScript yang sudah jadi, sehingga tidak perlu mengetik kode dari awal. Berikut tabel terkait detail pekerjaan yang dilakukan selama periode magang dilaksanakan:

Tabel 3. 1 Detail Pekerjaan yang Dilakukan

No.	Tanggal	Proyek	Keterangan
1.	15 – 17 September 2025	Diskusi kebutuhan serta alur kerja, dan analisis serta perancangan awal website.	Brainstorming untuk color palette yang akan digunakan, membuat use case serta activity diagram untuk 3 (tiga) pengguna, dan mencari referensi aplikasi yang serupa.
2.	22 September – 04 Oktober 2025	Pembuatan desain UI/UX untuk pembuatan website aplikasi dan pengembangan frontend halaman awal (login dan lupa password)	Membuat desain UI/UX menggunakan Figma mulai dari halaman login hingga logout (akun), serta halaman lainnya
3.	06 – 10 Oktober 2025	Pengembangan frontend halaman awal (login dan lupa password)	Percobaan awal untuk melakukan pengembangan frontend pada halaman login dan lupa password dengan menyesuaikan desain yang sudah dibuat sebelumnya
3.	06 – 31 Oktober 2025	Pengembangan frontend untuk halaman utama dan lainnya.	Setelah pengembangan halaman awal selesai, selanjutnya melakukan pengembangan untuk halaman utama dan lainnya sebagai inti dari proses pemesanan ruang rapat.
4.	03 – 07 November 2025	Testing ( <i>bug dan error</i> ) setelah pengembangan selesai	Melakukan testing sederhana seperti melakukan pemesanan ruang rapat serta pengecekan jadwal, dan proses-proses interaktif pada website aplikasi.
5.	10 November – 5 Desember 2025	Revisi terkait bug atau error yang ditemukan setelah testing	Hal yang ditemukan bug seperti waktu pemesanan yang bisa tetap submit ketika jadwal atau jam yang sudah dipesan sudah tidak tersedia, dll.

No.	Tanggal	Proyek	Keterangan
6.	08 – 19 Desember 2025	Finalisasi Pengembangan Website dan Testing lanjutan sebelum publikasi	Menyempurnakan UI/UX pada website aplikasi yang sedang dikembangkan dan testing kedua setelah finalisasi website aplikasi sudah selesai.

Pada Tabel 3.1 di atas merupakan detail pekerjaan yang dilakukan mencakup seluruh rangkaian yang dilakukan pada pembuatan proyek pengembangan *website* aplikasi pemesanan ruang rapat. Aktivitas yang dilakukan antara lain pengenalan lingkungan perusahaan, pengenalan proyek, pembuatan desain UI/UX, pengembangan *frontend*, testing, hingga revisi pengembangan.

Proses pelaksanaan proyek ini melibatkan beberapa aplikasi pendukung seperti Visual Studio Code, Figma, Draw.io, GitHub, dan Bootstrap. Aplikasi-aplikasi tersebut digunakan dengan menyesuaikan kebutuhan contohnya Visual Studio Code digunakan untuk menulis dan mengelola kode secara efisien. Berikut penjelasan terkait penggunaan aplikasi pendukung:

### 1) Visual Studio Code

Aplikasi Visual Studio Code merupakan sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows [5]. Bahasa yang digunakan dalam proyek ini antara lain ada HTML, CSS, dan JavaScript. Selain penggunaan bahasa pemrograman aplikasi visual studio code menyediakan *extension* serta *framework* untuk memudahkan developer dalam membuat sebuah *website* atau aplikasi. Pada proyek ini selain menggunakan bootstrap sebagai *framework*, mahasiswa menggunakan beberapa *extension* untuk memudahkan penggeraan proyek tersebut.

### 2) Figma

Aplikasi Figma merupakan aplikasi desain kolaboratif berbasis *cloud* yang populer untuk membuat desain antarmuka (UI/UX) aplikasi dan *website*, membuat prototipe interaktif, serta bekerja bersama tim secara *real-time* pada

satu kanvas digital, menggunakan alat vektor, komponen, dan fitur *prototyping* untuk mempercepat alur kerja desain digital [6]. Penggunaan dalam proyek ini hanya sebatas pembuatan desain UI/UX dengan *prototype* untuk menggambarkan interaksi yang akan dilakukan pada *website* aplikasi yang akan dikembangkan.

### 3) Draw.io

Aplikasi Draw.io merupakan sebuah aplikasi *open source* untuk membuat diagram *online* gratis yang memungkinkan pengguna membuat berbagai jenis visualisasi seperti diagram alir, UML, organisasi, jaringan, *mind map*, dan lainnya. Pada proyek ini draw.io digunakan untuk membuat *use case* diagram dan juga *activity* diagram. *Use case* diagram berfungsi untuk memvisualisasi apa yang bisa dilakukan sistem dan siapa yang berinteraksi aktor, sedangkan *activity* diagram menggambarkan bagaimana sebuah proses atau alur kerja dalam sistem berjalan langkah demi langkah [7].

### 4) GitHub

Aplikasi GitHub merupakan platform berbasis web untuk mengelola, menyimpan, dan berkolaborasi dalam proyek kode sumber menggunakan sistem kontrol versi Git, memungkinkan pengembang melacak perubahan, berbagi pekerjaan, dan bekerja sama secara efisien dengan fitur yang disediakan. Pada proyek ini menggunakan GitHub untuk menghubungkan antara satu *device* dengan *device* yang lain serta memudahkan jika ada perubahan yang sudah terpush maka masih terdapat dokumentasi *code* sebelumnya.

### 5) Bootstrap

Bootstrap merupakan *framework* untuk mempercepat pengembangan aplikasi web dengan menyediakan kerangka kerja seperti HTML, CSS, dan JavaScript yang siap pakai, sehingga memungkinkan pembuatan desain yang responsif serta interaktif. Pada proyek ini bootstrap biasanya digunakan pada struktur pembuatan tabel atau list agar lebih selaras bentuk tabel dan listnya.

## 3.3 Uraian Pelaksanaan Kerja

Selama melaksanakan program magang ditempatkan pada divisi IT *Application System* mahasiswa dilibatkan pada proyek pengembangan aplikasi untuk pemesanan ruang rapat. Bagian ini akan menjelaskan secara rinci dari pekerjaan mulai dari proses awal hingga akhir dari pengerjaan proyek tersebut.

### **3.3.1 Proses Pelaksanaan**

Langkah awal dalam pembuatan proyek ini adalah pengenalan kembali pada struktur dari code HTML, CSS dan JavaScript. Proses pengembangan ini dibantu dengan framework bootstrap untuk digunakan sebagai dasar membangun sistem.

Pada tahap ini menggunakan metode desain sprint untuk mengatasi tantangan desain, memecahkan masalah, dan mengembangkan solusi inovatif dalam waktu singkat. Sehingga proses pengerjaan proyek ini yang bisa dikatakan singkat hanya dalam beberapa bulan dapat menghasilkan aplikasi yang dapat berguna ke depannya. Berikut penjelasan terkait penggunaan desain sprint pada proyek pengembangan aplikasi pemesanan ruang rapat:

#### **1) *Understand***

Pada tahap ini merupakan tahap pertama dari desain sprint, dilakukan identifikasi masalah serta target pengguna. Hasilnya adalah proses pemesanan ruang rapat masih dilakukan secara manual dengan konfirmasi langsung kepada resepsionis perusahaan sehingga proses pemesanan dan konfirmasi tersedia ruangan menjadi lama serta tidak efisien sehingga target pengguna pengembangan aplikasi pemesanan ruang rapat berbasis web ini adalah seluruh karyawan baik di kantor pusat maupun kantor cabang.

#### **2) *Diverge***

Pada tahap ini baik pembimbing lapangan dan mahasiswa magang melakukan pemberian ide atau solusi terkait permasalahan yang sudah teridentifikasi sebelumnya. Hasilnya adalah pembimbing lapangan memberikan ide atau solusi untuk membuat aplikasi pemesanan ruang

rapat berbasis *website* yang dapat digunakan dan di akses secara langsung.

### 3) *Decide*

Pada tahap ini solusi yang dibuat pada tahap sebelumnya yaitu pembuatan aplikasi berbasis web untuk pemesanan ruang rapat, selain itu membuat diagram untuk menggambarkan alur kerja dan alur sistem yang akan dibuat.

### 4) *Prototype*

Pada tahap ini setelah selesai membuat diagram yang menggambarkan alur kerja dan alur sistem menggunakan *use case* diagram dan *activity* diagram, selanjutnya membuat desain UI/UX sebagai gambaran dari hal-hal yang sudah diidentifikasi dan diskusikan pada tahap-tahap sebelumnya. Sehingga hasil dari desain UI/UX ini dapat digunakan sebagai acuan desain pada proses pengembangan *frontend* nantinya.

### 5) *Validate*

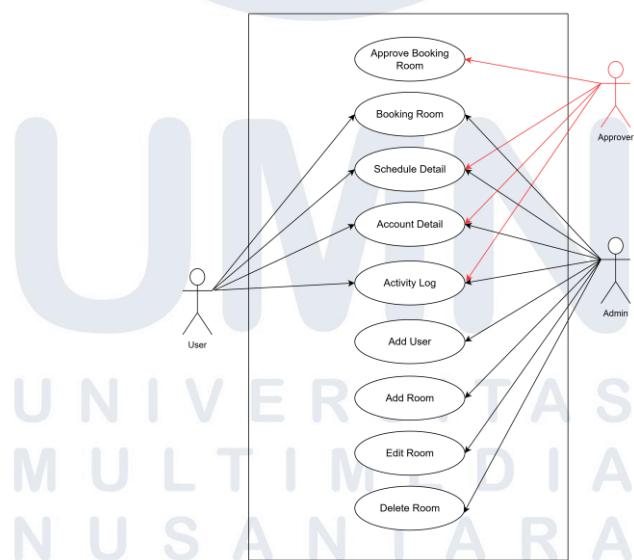
Setelah selesai tahap prototipe hal yang dilakukan selanjutnya adalah mempresentasikan hasil desain UI/UX kepada pembimbing lapangan, menanyakan beberapa hal terkait interaksi pengguna apakah sudah sesuai dengan kebutuhan atau masih perlu beberapa perbaikan. Pada penjelasan desain sprint di atas hanya penjelasan singkat, sehingga berikut adalah penjelasan detail terkait tahap-tahap dari desain *sprint* yang dibuat sebelumnya:

#### 3.3.1.1 Pembuatan *Use Case* dan *Activity Diagram*

Pada tahap ini masuk pada tahap *decide* yang dimana dilakukan tahap awal dalam pengembangan yaitu analisis kebutuhan dengan membuat *use case diagram* dan *activity diagram*. Sebelum membuat 2 (dua) diagram tersebut pada awal minggu pertama periode magang berlangsung, mahasiswa diperkenalkan dengan tujuan serta

pemberian akses proyek seperti *download* aplikasi visual studio *code* menggunakan laptop yang diberikan perusahaan sebagai aset selama melaksanakan kegiatan magang di Hypernet.

*Use case diagram* merupakan diagram yang menggambarkan hubungan antara aktor dengan sistem. *Use case diagram* bisa mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat [8]. Pada *use case diagram* terdapat 3 (tiga) komponen penting yang ada pada *use case*, pertama adalah sistem yang merupakan batasan dalam relas dengan aktor-aktor yang menggunakan dan fitur-fitur yang disediakan oleh sistem, kedua adalah aktor yang merupakan segala hal di luar sistem yang akan menggunakan sistem tersebut untuk melakukan sesuatu, dan yang terakhir *use case* itu sendiri merupakan gambaran fungsional dari sebuah sistem. Berikut gambar *use case diagram* pada penggerjaan proyek pengembangan *website* aplikasi pemesanan ruang rapat:



Gambar 3. 3 Use Case Diagram Aplikasi Pemesanan Ruang Rapat

Pada Gambar 3.3 merupakan *use case diagram* yang telah dibuat dan menggambarkan interaksi aktor pada sistem yang dibuat pada

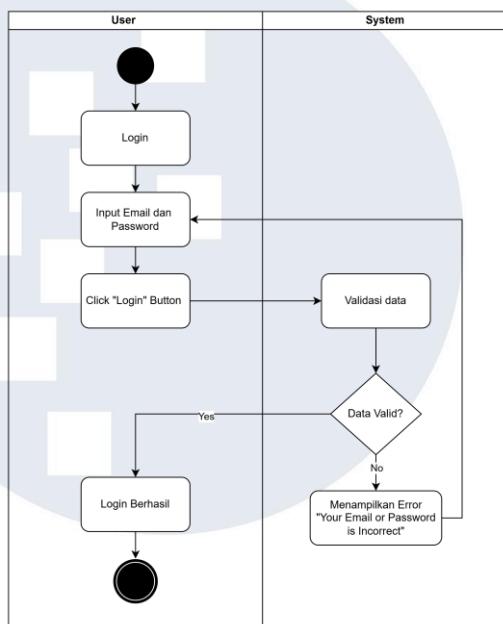
pengembangan *website* aplikasi pemesanan ruang rapat. Berikut penjelasan detail mengenai use case diagram di atas:

Aktor admin merupakan pihak yang memiliki hak akses penuh dalam pengelolaan sistem, baik dari sisi membantu pemesanan ruang rapat, pengaturan pengguna, maupun monitoring aktivitas pengguna. Selain itu, terdapat aktor *user* yang memiliki akses untuk melakukan pemesanan ruang rapat. *User* disini hanya dapat mengakses sistem untuk melakukan pemesanan, melihat status pemesanan, dan melihat jadwal. Terakhir ada aktor *approver* yang bertugas sebagai pemegang hak untuk menentukan status pemesanan apakah tersedia ataupun tidak tersedia.

Selain pembuatan *use case* diagram, terdapat pembuatan *activity* diagram. *Activity* diagram merupakan rancangan aliran aktivitas atau aliran kerja dalam sebuah sistem yang akan dijalankan. *Activity* Diagram juga digunakan untuk mendefinisikan atau mengelompokan aluran tampilan dari sistem tersebut [9].

*Activity* diagram memiliki 5 (lima) komponen yang terdapat pada *activity* diagram, pertama adalah *start point* atau *initial state* yang merupakan status awal dimulainya suatu aliran kerja pada *activity* diagram, pada satu *activity* diagram hanya memiliki satu *start point* atau *initial state*. Yang kedua adalah bagian *activity* merupakan aktivitas atau pekerjaan yang dilakukan dalam setiap aliran kerja, selanjutnya ada *decision* atau percabangan. *Decision* atau percabangan biasanya digunakan untuk menggambarkan pilihan kondisi dimana ada kemungkinan perbedaan transisi untuk memastikan bahwa aliran kerja dapat mengalir lebih dari satu jalur. Yang keempat adalah *merge* berfungsi untuk menggabungkan kembali aliran kerja yang sebelumnya telah dipisah oleh *decision*, dan yang terakhir adalah *end point* atau *final state* merupakan bagian akhir dari suatu aliran kerja pada sebuah *activity* diagram.

Pembuatan *activity* diagram ini dilakukan menggunakan *tools* draw.io yang dimana draw.io menyediakan beberapa bagian diagram menyesuaikan dengan kebutuhan dan penggunaan diagram yang diperlukan. Berikut beberapa *activity* diagram yang menggambarkan pada penggerjaan proyek pengembangan *website* aplikasi pemesanan ruang rapat:

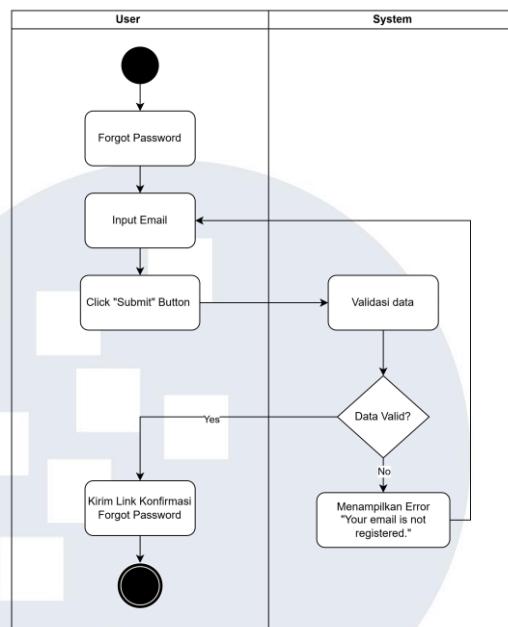


Gambar 3. 4 Activity Diagram Login

Pada Gambar 3.4 merupakan *activity* diagram yang menggambarkan alur proses login untuk pengguna aplikasi tersebut. Proses awal dimulai ketika pengguna berhasil mengakses halaman login, kemudian memasukkan email dan *password* lalu menekan tombol login. Aktivitas sistem akan memvalidasi data yang diinput serta memastikan informasi tersebut benar, jika data yang dimasukkan valid maka pengguna berhasil masuk ke dalam sistem (login berhasil).

Ketika data tidak valid, sistem akan menampilkan pesan kesalahan berupa kalimat “*Your Email or Password is Incorrect*” dan pengguna diminta untuk mengulangi proses login. Diagram ini menunjukkan interaksi antara pengguna dan sistem dalam

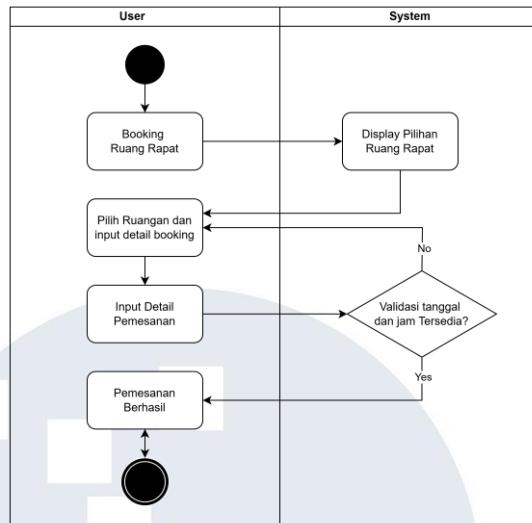
memastikan proses autentikasi berjalan dengan benar sebelum pengguna dapat mengakses fitur aplikasi.



Gambar 3. 5 Activity Diagram Forgot Password

Pada Gambar 3.5 merupakan activity diagram yang menggambarkan alur proses lupa kata sandi (*forgot password*) pada aplikasi pemesanan ruang rapat. Proses ini dimulai ketika pengguna memilih menu *forgot password* pada halaman *login*, kemudian memasukkan alamat email yang terdaftar dan menekan tombol *submit*.

Selanjutnya, sistem akan melakukan validasi alamat email yang terdaftar. Jika email terdaftar maka sistem akan mengirimkan *link* konfirmasi *forgot password* dan pengguna dapat membuat *password* baru. Namun, apabila email tidak terdaftar sistem akan menampilkan pesan kesalahan berupa kalimat “*Your email is not registered.*” dan pengguna diminta untuk mengulangi proses.

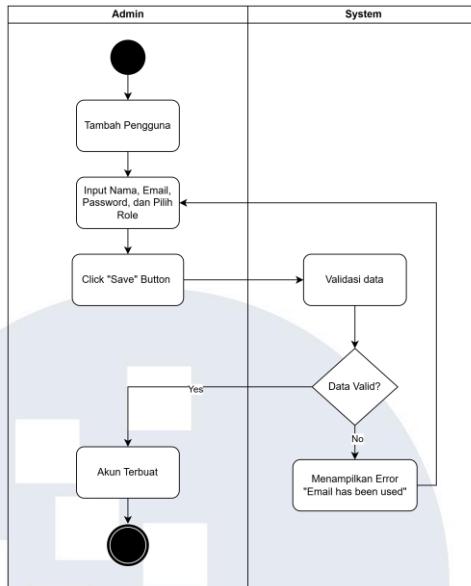


Gambar 3. 6 Activity Diagram Booking Meeting Room

Pada Gambar 3.6 merupakan *activity* diagram yang menggambarkan alur proses pemesanan ruang rapat. Proses ini dimulai dari memilih menu pemesanan ruang rapat, kemudian sistem akan menampilkan daftar pilihan ruang rapat yang tersedia.

Setelah pengguna memilih ruangan proses selanjutnya adalah pengguna mengisi detail pemesanan, seperti nama, divisi, nama acara, tanggal pemesanan, serta jam mulai dan selesai penggunaan ruangan. Jika jadwal tersedia, maka pemesanan dinyatakan berhasil. Namun, jika jadwal tidak tersedia, pengguna perlu menyesuaikan kembali detail pemesanan.

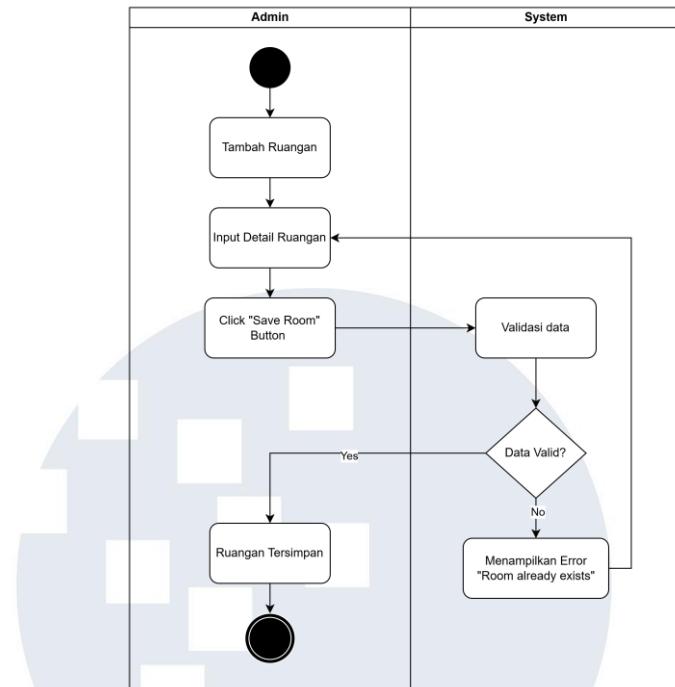
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3. 7 Activity Diagram Add New User

Pada Gambar 3.7 merupakan *activity* diagram yang menggambarkan alur proses penambahan pengguna baru oleh admin pada aplikasi pemesanan ruang rapat. Proses ini dimulai ketika admin memilih menu tambah pengguna (*add user*), kemudian mengisi detail data pengguna baru berupa nama, email, *password*, serta *role* yang sesuai lalu menekan tombol ‘*save*’. *Role* pada saat tambah pengguna baru ada 3 (tiga) jenis yaitu sebagai *admin*, *approver*, dan *user*.

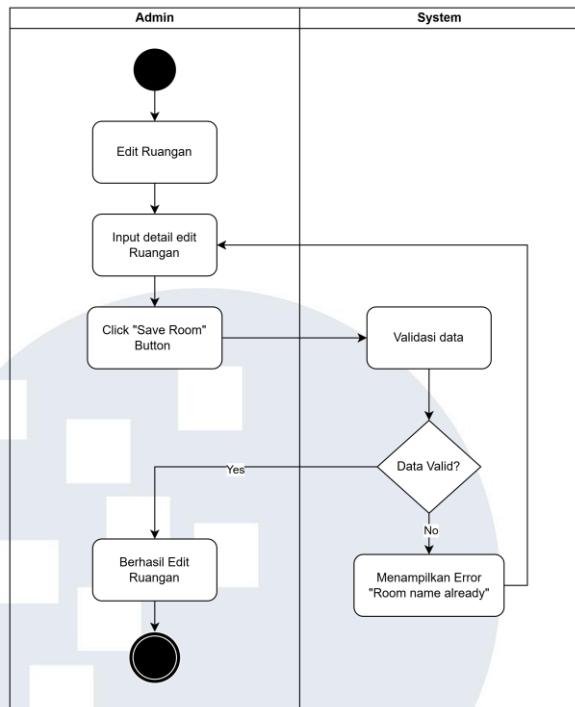
Selanjutnya, sistem melakukan validasi data. Jika data yang dimasukkan valid dan email belum digunakan maka sistem akan membuat akun pengguna dan proses dinyatakan berhasil. Namun, ketika email atau data tidak valid seperti kriteria *password* yang tidak sesuai maka sistem akan menampilkan sistem kesalahan berupa kalimat “*Email has been used*” dan admin diminta untuk melakukan perbaikan data.



Gambar 3. 8 Activity Diagram Add New Meeting Room

Pada Gambar 3.8 merupakan *activity* diagram yang menggambarkan alur proses penambahan ruangan pada aplikasi pemesanan ruang rapat. Proses ini dimulai ketika admin memilih menu tambah ruangan (*add room*), kemudian admin mengisi informasi pada detail ruangan seperti nama ruangan, jumlah kapasitas, lokasi, gambar ruangan dan menekan tombol ‘*save*’.

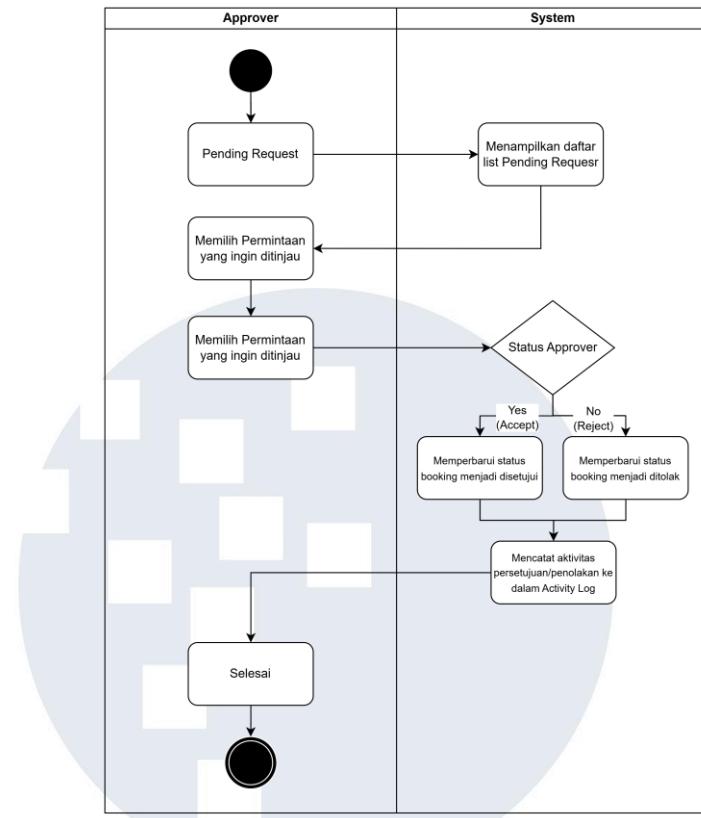
Selanjutnya, sistem akan menerima data dan melakukan validasi data. Jika data valid diartikan ruangan yang ingin ditambahkan belum tersedia maka proses berlanjut ke status “Ruangan Tersimpan”, namun jika data tidak valid diartikan bahwa ruangan tersebut telah tersedia maka sistem akan menampilkan pesan kesalahan dengan kalimat “*Room already exists*” dan sistem meminta untuk kembali ke tahap input detail ruangan agar admin memperbaiki data.



Gambar 3. 9 Activity Diagram Edit Meeting Room

Pada Gambar 3.9 merupakan *activity* diagram yang menggambarkan alur proses edit ruangan pada aplikasi pemesanan ruang rapat. Proses dimulai ketika admin menekan tombol edit pada ruangan tertentu, kemudian admin mengubah dan memasukkan data baru pada detail edit ruangan. Setelah detail sudah di rubah dan merasa cukup maka admin dapat menekan tombol ‘*save*’.

Selanjutnya, sistem akan melakukan validasi data. Jika data valid maka sistem akan otomatis menyimpan perubahan dan proses selesai. Namun, ketika data tidak valid maka sistem akan menampilkan pesan kesalahan berupa kalimat “*Room name already*” dan sistem meminta admin untuk kembali ke tahap input agar admin bisa memperbaiki data tersebut.



Gambar 3. 10 Diagram Activity Approver Booking Request

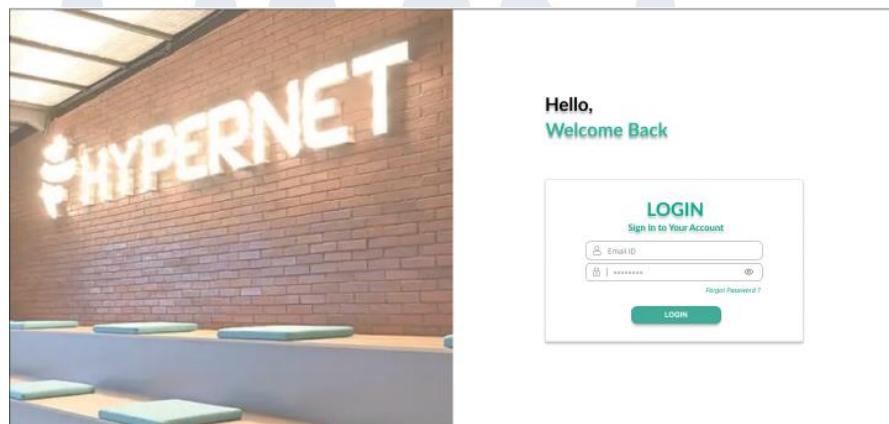
Pada Gambar 3.10 merupakan *activity diagram* untuk *approver* melakukan tindakan menyetujui atau menolak dari *request booking* yang masuk ke dalam sistem aplikasi. Proses dimulai ketika *Approver* mengakses menu *Pending Request* dan *System* merespons dengan menampilkan daftar lengkap permintaan yang menunggu tinjauan. *Approver* kemudian memilih permintaan spesifik yang ingin ditinjau. Setelah permintaan dipilih, *System* menunggu tindakan persetujuan dari *Approver* melalui *Decision Node* "Status Approver". Jika *Approver* memilih *Yes (Accept)*, *System* akan memperbarui status booking menjadi disetujui, dan jika *Approver* memilih *No (Reject)*, *System* akan memperbarui status *booking* menjadi ditolak. Kedua jalur tindakan (setuju atau tolak) kemudian mengarah pada langkah pencatatan aktivitas, di mana *System* mencatat hasil

persetujuan/penolakan ke dalam *Activity Log*, sebelum proses dinyatakan Selesai.

### 3.3.1.2 Pembuatan Desain UI/UX

Langkah selanjutnya setelah membuat *use case* diagram dan *activity* diagram adalah masuk ke tahap prototipe pembuatan desain UI/UX untuk tampilan *website*. Tujuan dari pembuatan desain ini sebagai jembatan antara kebutuhan pengguna dengan implementasi teknis, tujuan teknisnya adalah memberikan panduan visual yang presisi kepada tim.

Pembuatan desain UI/UX ini mencakup desain tata letak (*layout*), *style guide* (*color palette*, tipografi dan *spacing*), dan interaksi (prototipe). Selain itu, tujuan lainnya adalah membuat simulasi aplikasi yang mendekati aslinya. Hal ini memungkinkan tim atau *stakeholder* melihat dan mencoba alur aplikasi secara nyata tanpa perlu menunggu proses coding selesai terlebih dahulu. Berikut hasil dari pembuatan desain UI/UX untuk halaman-halaman yang akan ada pada *website* aplikasi pemesanan ruang rapat PT Hipernet Indodata:

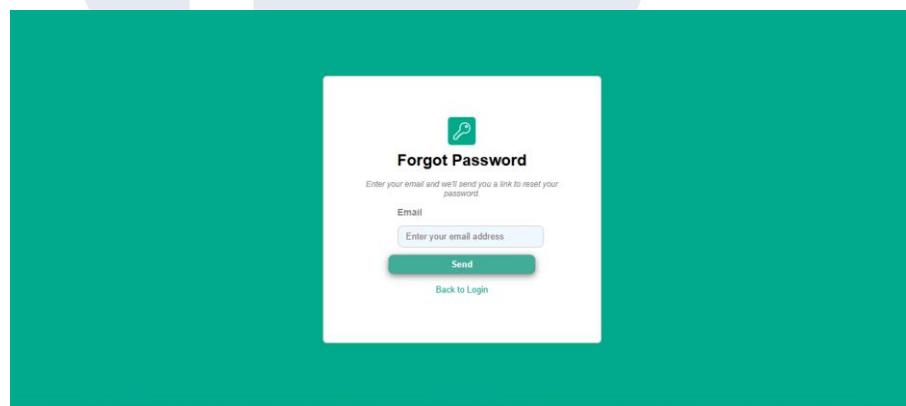


Gambar 3. 11 Desain UI Halaman Login

Pada Gambar 3.11 merupakan desain untuk halaman login. Pada sisi kiri halaman terdapat latar belakang visual dengan logo Hypernet,

tujuan dari latar belakang ini untuk memperkuat *branding* serta memberikan estetika yang modern atau kasual pada halaman *login*.

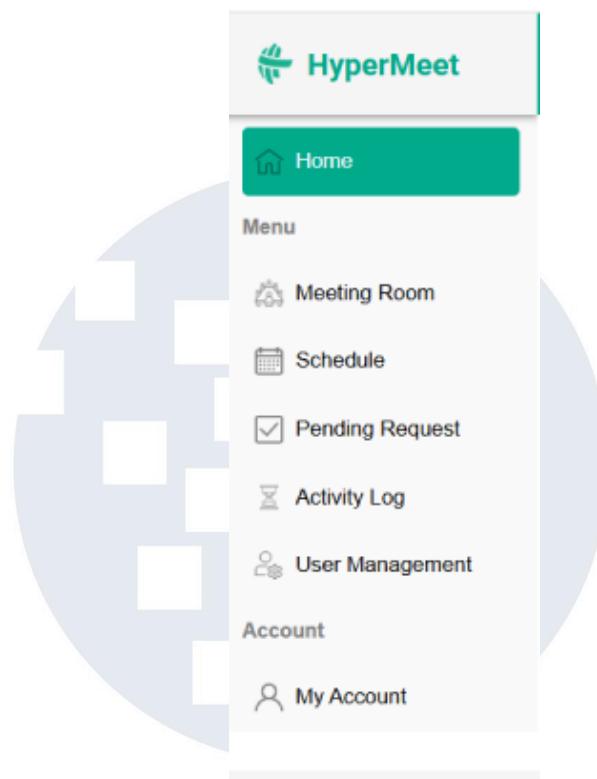
Selanjutnya, pada sisi kanan merupakan area utama untuk input detail pengguna. Area utama menggunakan *background* berwarna putih yang bersih agar pengguna fokus pada proses login tersebut. Selain itu, terdapat *header* berupa kalimat sapaan “*Hello, Welcome Back*” untuk memberikan kesan personal serta menjadi tanda untuk melakukan login terlebih dahulu. Form input yang terdapat pada halaman login adalah input email dan password, selanjutnya aksi yang dapat dilakukan adalah menekan tombol utama login dengan *background* tombol berwarna #41AC98 (tosca) serta dapat melakukan *forgot password* untuk navigasi ke halaman *forgot password*.



Gambar 3. 12 Desain UI Forgot Password

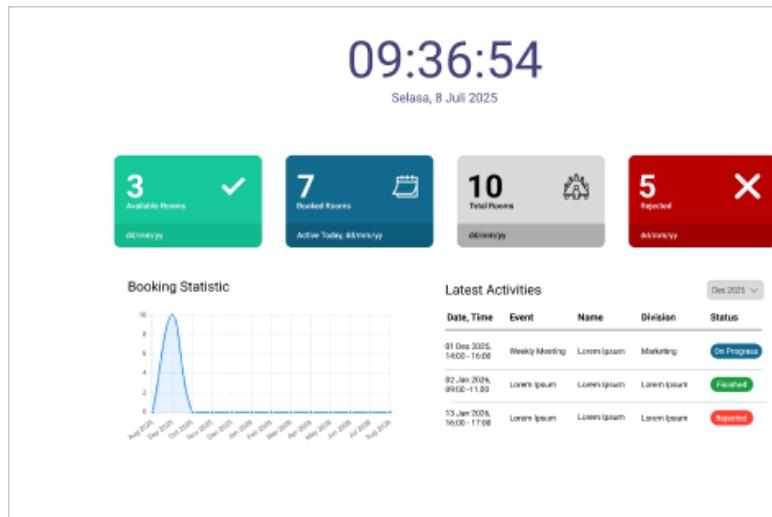
Pada Gambar 3.12 merupakan desain tampilan untuk halaman *forgot password*. Halaman ini menampilkan desain ketika pengguna menggunakan ingin melakukan tindakan pemulihian kata sandi ketika pengguna lupa dengan kata sandi. Halaman ini memiliki judul utama “*Forgot Password*” yang disertai ikon kunci untuk menggambarkan konteks keamanan. Hal yang perlu diisi pada halaman ini adalah email agar sistem mengirimkan tautan untuk mengatur ulang kata sandi. Di bagian bawah, terdapat tautan “*Back to Login*” yang dimana tautan atau navigasi ini akan mengarahkan pengguna dapat kembali ke halaman login jika pengguna mengingat kata sandi mereka. Secara

keseluruhan, desain ini minimalis dan berfokus pada satu tujuan, yaitu memulai proses reset kata sandi.



Gambar 3. 13 Desain UI Side Bar

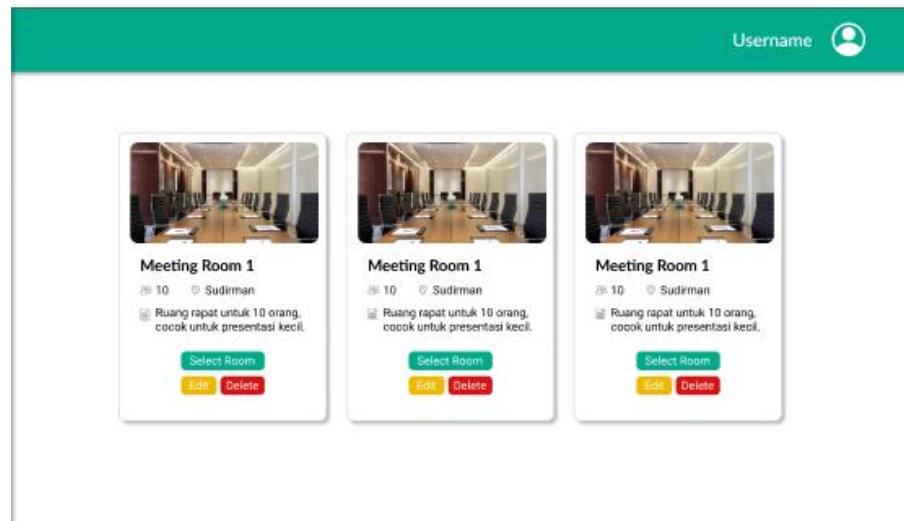
Pada Gambar 3.13 merupakan tampilan untuk *side bar* yang berisikan menu-menu yang akan ada pada website aplikasi pemesanan ruang rapat. Desain kedua adalah Sidebar Navigasi untuk aplikasi bernama HyperMeet. Sidebar ini dirancang untuk mempermudah navigasi dalam sistem, dibagi menjadi beberapa bagian. Bagian utama menampilkan menu "Home" yang disorot (menandakan halaman saat ini). Bagian Menu mencakup fungsionalitas inti aplikasi seperti *Meeting Room*, *Schedule*, *Pending Request*, *Activity Log*, dan *User Management*, yang menunjukkan bahwa aplikasi ini berfokus pada manajemen ruangan atau pertemuan. Terakhir, bagian Account hanya berisi *My Account*, untuk mengelola profil pengguna.



Gambar 3. 14 Desain UI Halaman Utama (Dashboard)

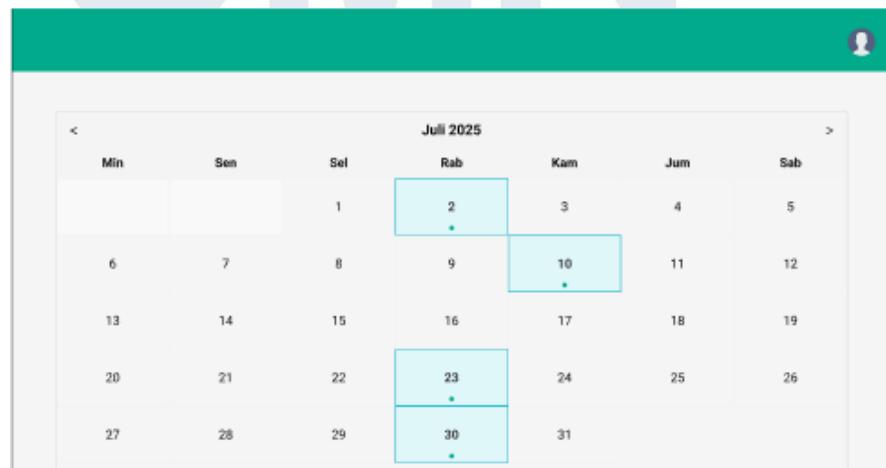
Pada Gambar 3.14 merupakan tampilan halaman utama pada menu navigasi pertama side bar “*Home*”. Halaman ini menampilkan *dashboard* yang berfungsi sebagai pusat kontrol secara real-time untuk memberikan informasi. Bagian atas menampilkan waktu dan tanggal saat pengguna masuk atau login ke *website* pemesanan ruang rapat, lalu bagian bawahnya terdapat kartu ringkasan yang menunjukkan status serta total aktivitas ruangan seperti ketersediaan ruangan, ruangan yang telah dipesan, dll.

Selain terdapat tanggal dan waktu serta kartu ringkasi, terdapat juga visualisasi data historis menggunakan *booking static (line chart)* berupa grafik bulanan yang menunjukkan aktivitas pemesanan ruangan. Terakhir, yang ada dibagian kanan *booking static* adalah bagian *latest activities* yang mencatat kegiatan terbaru. menampilkan detail seperti Tanggal, Waktu, Event, Nama, Divisi, dan Status, di mana status ditandai dengan label seperti "On Progress" (misalnya pada 01 Des 2025), "Finished" (misalnya pada 02 Jan 2026), atau "Rejected" (misalnya pada 13 Jan 2026). Tampilan halaman utama ini akan ada pada semua role (*admin*, *user*, dan *approver*), karena halaman ini merujuk pada operasional pemesanan ruangan.



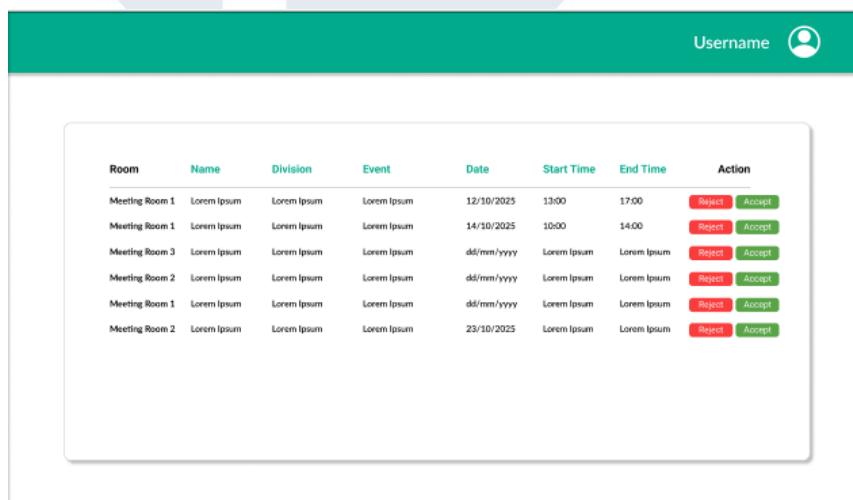
Gambar 3. 15 Desain UI Halaman Meeting Room

Pada Gambar 3.15 merupakan *user interface* dari halaman “*meeting room*”. Halaman ini menampilkan pilihan ruangan yang tersedia dengan format kartu yang rapi. Setiap bagian kartu menampilkan informasi seputar ruangan, seperti foto ruangan, nama ruangan, kapasitas (ikon orang), lokasi ruangan (ikon pin lokasi), deskripsi ruangan. Tampilan *user interface* ini berlaku untuk admin, sedangkan untuk user tampilan aksi hanya terdapat tombol “*Select room*” untuk memilih ruangan dan melanjutkan proses pemesanan ruangan. Tombol edit dan *delete* berlaku untuk admin dalam mengelola detail dan inventaris ruangan.



Gambar 3. 16 Desain UI Halaman Schedule

Pada gambar 3.16 merupakan tampilan untuk halaman “*schedule*”. Halaman ini menampilkan kalender bulanan, yang berfokus untuk visualisasi detail pemesanan ruangan di hari tersebut. Tampilan kalender standar menunjukkan hari dalam seminggu dan tanggal-tanggal pada bulan tersebut. Fitur utamanya adalah penandaan detail pemesanan pada tanggal tertentu, jika pada tanggal tersebut terdapat lingkaran kecil di bawah tanggal tertentu menunjukkan bahwa ditanggal tersebut terdapat aktivitas pemesanan ruangan. Kalender ini memungkinkan pengguna untuk melihat status jadwal secara sekilas per bulan, dan diasumsikan pengguna dapat menekan tanggal yang disorot untuk melihat detail pemesanan ruangan yang lebih rinci. Diharapkan ketika user atau pengguna ketika ingin melakukan pemesanan ruangan dapat melihat detail pemesanan ruangan di hari yang dituju.



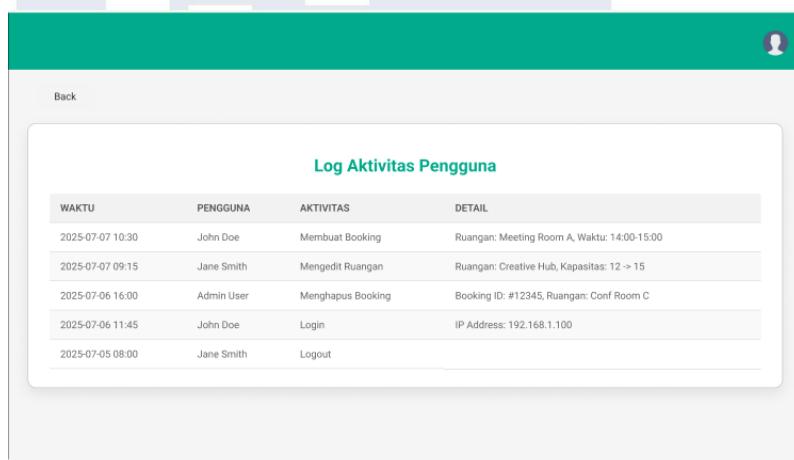
Room	Name	Division	Event	Date	Start Time	End Time	Action
Meeting Room 1	Lorem Ipsum	Lorem Ipsum	Lorem Ipsum	12/10/2025	13:00	17:00	<span>Reject</span> <span>Accept</span>
Meeting Room 1	Lorem Ipsum	Lorem Ipsum	Lorem Ipsum	14/10/2025	10:00	14:00	<span>Reject</span> <span>Accept</span>
Meeting Room 3	Lorem Ipsum	Lorem Ipsum	Lorem Ipsum	dd/mm/yyyy	Lorem Ipsum	Lorem Ipsum	<span>Reject</span> <span>Accept</span>
Meeting Room 2	Lorem Ipsum	Lorem Ipsum	Lorem Ipsum	dd/mm/yyyy	Lorem Ipsum	Lorem Ipsum	<span>Reject</span> <span>Accept</span>
Meeting Room 1	Lorem Ipsum	Lorem Ipsum	Lorem Ipsum	dd/mm/yyyy	Lorem Ipsum	Lorem Ipsum	<span>Reject</span> <span>Accept</span>
Meeting Room 2	Lorem Ipsum	Lorem Ipsum	Lorem Ipsum	23/10/2025	Lorem Ipsum	Lorem Ipsum	<span>Reject</span> <span>Accept</span>

Gambar 3. 17 Desain UI Pending Request

Pada Gambar 3.17 merupakan tampilan untuk halaman *pending request*, halaman ini berfungsi sebagai pusat persetujuan atau penolakan dari permintaan pemesanan ruangan. Tampilan utama pada halaman ini berbentuk tabel yang terstruktur, serta berisi semua permintaan yang masuk dan sedang menunggu tindakan persetujuan atau penolakan. Kolom-kolom tabel memberikan detail lengkap

mencakup *room* (nama ruangan), *name* (nama ruangan), *division* (divisi pemesan), *event* (nama acara), *date* (tanggal pemesanan, misalnya 01/10/2025), *start time* (waktu mulai), dan *end time* (waktu selesai).

Kolom terakhir yaitu *action* adalah kolom interaktif untuk *approver* dapat memilih tindakan untuk *reject* (tolak, jika ruangan atau jam tidak tersedia) atau *accept* (setuju, jika ruangan atau jam tersedia). Desain tabel ini memastikan bahwa *approver* memiliki semua informasi terkait detail untuk membuat keputusan persetujuan secara cepat dan efisien.

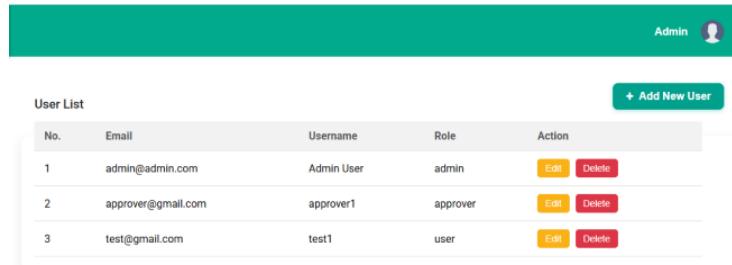


Gambar 3. 18 Desain UI Log Activity

Pada Gambar 3.18 merupakan tampilan dari halaman *log activity*, halaman ini berfungsi untuk mencatat semua tindakan pengguna dalam sistem, menyediakan detail historis yang penting. Data log aktivitas menggunakan format tabel yang jelas dengan empat kolom utama yaitu waktu, pengguna, aktivitas, dan detail. Aktivitas yang akan dicatat mencakup fungsionalitas pengguna, seperti *login* serta *logout*, membuat pemesanan ruangan, mengedit ruangan, dan sebagainya.

Kolom Detail memberikan konteks spesifik dari setiap aktivitas, misalnya mencantumkan *Booking ID*, ruangan yang terpengaruh, atau

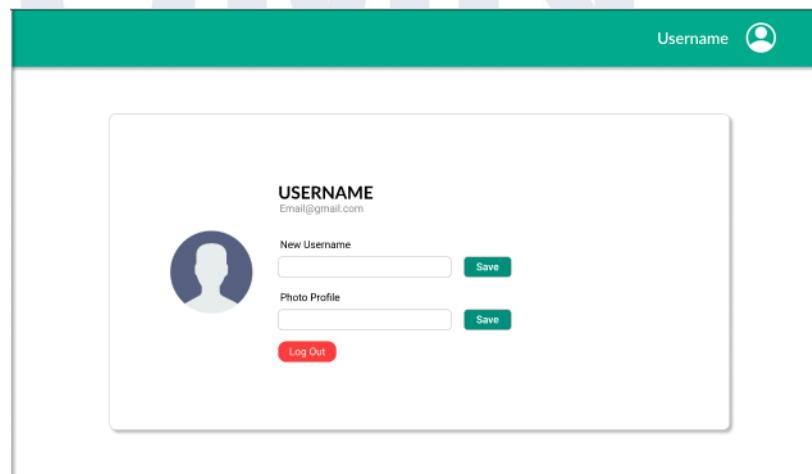
*IP Address* saat pengguna *login*. Meskipun tidak bersifat *real-time* murni, log ini menyediakan catatan kronologis yang akurat dari semua interaksi pengguna untuk keperluan pemantauan dan keamanan.



User List				
No.	Email	Username	Role	Action
1	admin@admin.com	Admin User	admin	<button>Edit</button> <button>Delete</button>
2	approver@gmail.com	approver1	approver	<button>Edit</button> <button>Delete</button>
3	test@gmail.com	test1	user	<button>Edit</button> <button>Delete</button>

Gambar 3. 19 Desain UI User Management

Pada Gambar 3.19 merupakan tampilan untuk halaman *user management*, halaman ini berfungsi khusus untuk admin dalam mengelola daftar pengguna yang memiliki akses. Data pengguna disajikan dalam format tabel berjudul “*user list*”. Setiap baris tabel mencantumkan detail terkait email pengguna, *username*, *role*, dan *action* yang bisa dilakukan yaitu *edit* dan *delete*. Selain itu, admin memiliki fitur untuk menambahkan pengguna ke dalam sistem yang letaknya pada tampilan berada di sisi kanan atas. Secara keseluruhan, halaman ini memberikan kontrol penuh kepada Administrator untuk mengelola hak akses dan identitas pengguna dalam aplikasi.



Gambar 3. 20 Desain UI My Account

Pada Gambar 3.20 merupakan tampilan untuk halaman *my account*, halaman ini berfungsi untuk detail informasi dari pengguna. Halaman ini menampilkan detail identitas pengguna, dengan *username* ditampilkan dengan huruf kapital besar dan alamat email yang terdaftar di bawah dari *username*. Terdapat ikon untuk menampilkan foto profil pengguna, selain itu pada sisi kanan terdapat formulir sederhana yang memungkinkan pengguna dapat melakukan dua modifikasi yaitu mengubah *username* dan mengunggah foto *profile* baru. Pada setiap kolom *input* memiliki aksi tombol *save* sendiri untuk menyimpan perubahan, selain itu terdapat tombol *logout* berwarna merah untuk keluar dari sesi akun.

The image shows a mobile application interface for editing a meeting room. At the top, there is a green header bar with a user icon on the right. Below the header, a back button is on the left. The main content area is titled "Form Edit Ruang Meeting". It contains four text input fields: "Nama Ruangan", "Kapasitas Ruangan", "Lokasi", and "Deskripsi (opsional)". Below these is a "Foto Ruangan" section with a file input field ("Choose File No file chosen") and a "Reset Foto" button. To the right of this section is a "Preview Foto Ruangan" placeholder. At the bottom is a large green "Simpan Ruangan" button.

Gambar 3. 21 Desain UI Form Edit Meeting Room

Pada Gambar 3.21 merupakan tampilan untuk halaman edit ruang meeting. Halaman ini memiliki struktur yang sama dengan halaman pemesanan serta penambahan ruangan. Hanya dibedakan dari form input yang tersedia pada setiap halaman.

Desain dari halaman ini berbentuk formular dan berisikan kolom-kolom input untuk detail yang akan dimasukkan. Contohnya pada halaman *edit meeting room input detail* yang dapat dirubah meliputi nama ruanganm kapasitas ruangan, Lokasi, deskripsi dan dapat

memilih foto ruangan serta reset foto ruangan. Penjelasan dari tampilan desain UI/UX yang sudah dibuat di atas berfungsi untuk gambaran terhadap pembuatan struktur website yang akan dibuat.

### 3.3.1.3 Pengembangan Frontend Website Aplikasi

Setelah melakukan pembuatan desain UI/UX, proses selanjutnya adalah masuk ke tahap yang penting dalam proyek pengembangan website aplikasi pemesanan ruang rapat menggunakan HTML, CSS, dan JavaScript. Berikut detail dari proses pengembangan website aplikasi pemesanan ruang rapat yang sudah dibuat:

#### 1) Halaman Login

Setelah membuat desain tampilan UI/UX menggunakan figma, desain tersebut digunakan untuk membuat struktur website pada halaman login. Struktur website aplikasi ini dibagi menjadi 2 (dua) bagian yaitu *left-panel* dan *right-panel*.

```
<div class="left-panel">
</div>
<div class="right-panel">
  <div class="welcome-text">
    <h1>Hello,</h1>
    <h2>Welcome Back</h2>
    <!-- <h3>Please enter your details</h3> -->
  </div>

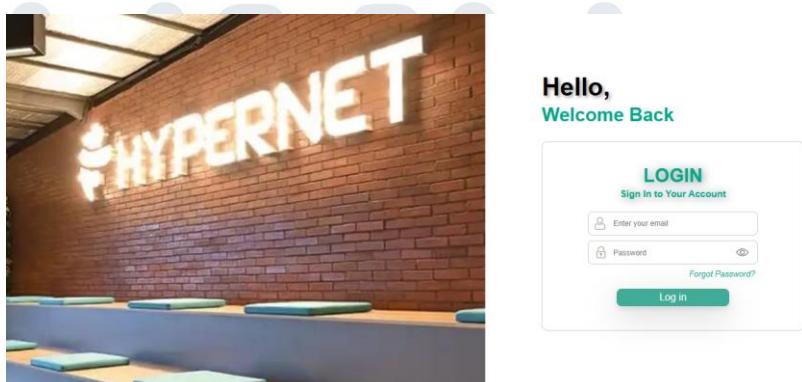
  <div class="container">
    <div class="login-header">
      <h3>LOGIN</h3>
      <div class="login-sign">
        <h3>Sign In to Your Account</h3>
        </div>
    </div>
    <form method="POST" action="{{ route('login.post') }}">
      @csrf

      <input type="email" name="email" id="email" placeholder="Enter your email" required>
      <br>
      <!-- update hide and seek password -->
      <div class="password-container">
        <input type="password" name="password" id="pass" placeholder="Password" required>
        
      </div>
    </form>
  </div>
</div>
```

Gambar 3. 22 Struktur HTML Halaman Login

Pada Gambar 3.22 merupakan *overview* dari struktur HTML yang dibuat sebagai halaman login. Sesuai yang dijelaskan pada paragraf di atas bahwa struktur pada halaman login ini dibagi menjadi 2 (dua) bagian. Pada panel kanan (*right-panel*) berisikan teks sambutan, dan dibawah kata sambutan terdapat *container* yang menjadi wadah formulir login sesungguhnya.

Formulir ini meminta dua input wajib (required), yaitu Email (bertipe email) dan Password (bertipe password, dilengkapi fitur toggle untuk melihat karakter). Formulir menggunakan metode POST dan memiliki penanganan kesalahan (@if (@Errors->any())) untuk menampilkan pesan validasi backend jika input tidak valid. Selain itu, disediakan tautan "*Forgot Password?*" untuk memfasilitasi pemulihan akun, dan diakhiri dengan tombol submit bertuliskan "*Log in*" untuk otentikasi. Fitur *toggle password* atau untuk melihat karakter dari *password* didukung dengan menggunakan JavaScript untuk menyimpan seluruh logika dan skenario untuk menampilkan karakter dari kata sandi. Berikut hasil dari pembuatan struktur website yang sudah diselesaikan secara keseluruhan:



N U M M A A  
Gambar 3. 23 Tampilan Website Halaman Login

Pada Gambar 3.23 merupakan tampilan dari website setelah melakukan pengembangan menggunakan HTML, CSS, dan JavaScript. Pada halaman ini belum menggunakan bootstrap karena halaman ini dibentuk secara manual, desain dari halaman ini terdapat perbedaan dari desain UI/UX yang dibuat sebelumnya yaitu terdapat placeholder di form input untuk email dan password.

## 2) Halaman *Forgot Password*

Tahap selanjutnya adalah pembuatan halaman *forgot password*, untuk ke halaman ini perlu menambahkan referensi yang mengacu kepada halaman *forgot password* ini dari halaman *login*. `<a href="{{ url('/forgot-password') }}>Forgot Password?</a>`, merupakan jalur navigasi yang memungkinkan pengguna mengklik teks “*Forgot Password?*” dan secara otomatis akan dibawa ke halaman *forgot password*. Berikut struktur *website* dari halaman *forgot password* yang sudah dibuat:

```
<div class="container">
  <form action="">
    
    <h2>Forgot Password</h2>
    <p>Enter your email and we'll send you a link to reset your password.</p>
    <div class="input-email">
      <label for="email">Email</label><br><br>
      <input type="email" name="" id="email" placeholder="Enter your email address">
    </div>
    <br><br>
    <button type="submit" class="sub">Send</button>
    <br><br>
    <a href="http://127.0.0.1:8000/">Back to Login</a>
  </form>
</div>
```

Gambar 3. 24 Struktur HTML Halaman Forgot Password

Pada Gambar 3.24 merupakan struktur dari halaman *forgot password*, halaman tersebut menggunakan sebuah *container* yang di dalamnya terdapat formulir (`<form>`). Hasil desain yang ditampilkan dari halaman *forgot password* tidak ada yang berbeda dengan desain UI/UX yang sudah dibuat.

### 3) Side Bar

Selanjutnya setelah berhasil *login* ke *website* aplikasi pemesanan ruang rapat, tahap lanjutnya ke pengembangan *side bar*. *Side bar* ini dibuat untuk navigasi dalam berpindah ke halaman-halaman yang akan ada pada aplikasi ini. Berikut struktur *website* untuk *side bar*:

```
<div class="dashboard-menu">
  <div class="tombol" id="home">
    
    <h3>Home</h3>
  </div>
  <p>Menu</p>
  <div class="tombol" id="ruang-meeting">
    
    <a href="{{ route('ruangan.index') }}"><h3>Meeting Room</h3></a>
  </div>

  <div class="tombol" id="jadwal">
    
    <a href="{{ route('jadwal') }}"><h3>Schedule</h3></a>
  </div>

  <div class="tombol" id="pending-request">
     {{-
    <a href="{{ route('booking.waiting') }}"}><h3>Pending Request</h3></a>
  </div>

  <div class="tombol" id="log-activity">
     {{-- Assuming you
    <a href="{{ route('admin.activity-log') }}"}><h3>Log Activity</h3></a>
  </div>

  <div class="tombol" id="user-manage">
    
    <a href="{{ route('user.index') }}"><h3>User Management</h3></a>
  </div>
```

Gambar 3. 25 Struktur HTML Side Bar

Pada Gambar 3.25 merupakan susunan daftar menu di dalam sebuah elemen kontainer (*dashboard-body*). Setiap item menu dibungkus dalam div dengan kelas tombol dan menyertakan ikon (*<img>*) serta tautan (*<a>*) yang menggunakan *framework routing* (seperti *{{ route('...') }}*) untuk mengarahkan pengguna ke halaman yang relevan.

#### 4) Halaman Utama (*Home*)

Pada bagian ini menampilkan struktur HTML dari halaman utama yang berisikan *dashboard* untuk melihat segala aktivitas yang terdapat pada *website* aplikasi pemesanan ruang rapat.

```
<div class="main-body">
  <div id="jam">00:00:00</div>
  <div id="tanggal">Hari, 00, 00, 0000</div>
</div>

<!-- ===== Bagian Cards ===== -->
<div class="main-cards">
  <div class="card available">
    <div class="card-inner">
      <h2>3</h2>
      <span class="material-icons-outlined">check</span>
    </div>
    <h1>Available Rooms</h1>
    <span>dd/mm/yy</span>
  </div>

  <div class="card booked">
    <div class="card-inner">
      <h2>{{ $requestBookings }}</h2>
      <span class="material-icons-outlined">lock</span>
    </div>
    <h1>Booking Requests</h1>
    <span>dd/mm/yy</span>
  </div>
</div>
```

Gambar 3. 26 Struktur HTML Halaman Utama (Header dan Card)

Pada Gambar 3.26 merupakan struktur dari halaman utama terutama pada header yang berisikan jam dan ringkasan dari aktivitas ruangan. Bagian struktur HTML ini dimulai dengan menggunakan kontainer main-header yang menampilkan nama pengguna serta foto profil yang telah diatur pada halaman ‘*My Account*’ dan dapat pindah ke halaman ‘*My Account*’ dengan tautan yang sudah diatur.

Setelah itu, pada bagian bawah dari header username terdapat bagian main-body yang dikhususkan untuk menampilkan waktu dan tanggal *real-time*. Selanjutnya, pada bagian cards (*main-cards*) yang menampilkan visual utama *dashboard*.

```

<!-- ===== Bagian Chart & Tabel ===== -->
<div class="dashboard-content">
    <div class="chart-section">
        <h2>Booking Statistic</h2>
        <select id="year-select">
            <option value="2025" selected>2025</option>
            <option value="2026">2026</option>
            <option value="2027">2027</option>
        </select>
        <canvas id="bookingChart"></canvas>
    </div>

    <div class="activity-section">
        <div class="activity-header">
            <h2>Latest Activities</h2>
            <select name="" id="select-box">
                <option>Oct 2025</option>
                <option>Nov 2025</option>
                <option>Des 2025</option>
                <option>Jan 2026</option>
                <option>Feb 2026</option>
            </select>
        </div>
    </div>

```

Gambar 3. 27 Struktur HTML Halaman Utama (Chart dan Tabel)

Pada Gambar 3.27 merupakan struktur dari halaman utama khususnya pada bagian chart dan tabel, kode HTML ini berfokus pada penyajian data analitis tiap tahun dan aktivitas terbaru pengguna. Bagian *chart-section* akan menampilkan visualisasi data menggunakan *line chart* yang berjudul “*Booking Static*” yang selanjutnya pada bagian bawah judul terdapat elemen dropdown yang memungkinkan pengguna website untuk memilih tahun yang diinginkan misalnya tahun 2025 hasil dari memilih tahun yang diinginkan akan menampilkan filter statistic per tahunnya.

Selanjutnya, terdapat satu bagian untuk menampilkan aktivitas terbaru dari pengguna yang tercatat oleh sistem. Bagian ini berjudul “*Latest Activities*”, pada bagian ini terdapat fitur dropdown seperti chart sebelumnya untuk menampilkan filter berdasarkan bulan dan tahun

## 5) Halaman *Meeting Room*

```
<div class="container-ruangan">
  @foreach ($ruangans as $ruangan)
    <div class="card">
      nama }}"/>
      <h3>{{ $ruangan->nama_ruangan }}</h3>
      <div class="info-row">
        <span class="material-icons-outlined">person</span>
        <p>{{ $ruangan->kapasitas }}</p>
      </div>
      <div class="info-row">
        <span class="material-icons-outlined">location_on</span>
        <p>{{ $ruangan->lokasi }}</p>
      </div>
      <div class="info-row">
        <span class="material-icons-outlined">description</span>
        <p>{{ $ruangan->deskripsi }}</p>
      </div>
      <form method="GET" action="{{ route('ruangan.booking.form', ['id' => $ruangan->id, 'nama' => Str::slug($ruangan->nama_ruangan)] )}}">
        <button class="pilih" type="submit">Select Room</button>
      </form>
    </div>
  @endforeach
</div>
```

Gambar 3. 28 Struktur HTML Halaman Meeting Room

Pada Gambar 3.28 merupakan struktur dari halaman meeting room, menghasilkan daftar ruangan yang tersedia dalam format card yang dapat berulang, di mana setiap card memvisualisasikan satu ruangan secara rinci. Struktur kode menggunakan *loop* (*@foreach*) untuk menampilkan setiap ruangan (\$ruangan). Setiap card dimulai dengan menampilkan foto ruangan, diikuti oleh nama ruangan sebagai judul (*<h3>*). Detail fungsional ruangan kemudian disajikan dalam baris-baris informasi (*info-row*) yang dilengkapi dengan ikon untuk kapasitas (orang), lokasi, dan deskripsi. Akhirnya, di bagian bawah setiap card, terdapat tombol "Select Room" yang memungkinkan pengguna memilih ruangan tersebut dan diarahkan ke formulir pemesanan (*booking form*) yang spesifik.

## 6) Halaman *Schedule*

```
<main class="box-2">
  <div class="main-header">
    <a href="{{ route('admin.akun') }}><p class="username"></p></a>
    <a href="{{ route('admin.akun') }}></a>
  </div>
  <div class="main-body">
    <div class="container-kalender">
      <div class="kalender">
        <div class="header-kalender">
          <button id="prevMonth"><span class="material-icons">arrow_back_ios_new</span></button>
          <span id="monthYear"></span>
          <button id="nextMonth"><span class="material-icons">arrow_forward_ios</span></button>
        </div>
        <div class="days-name">
          <div>Mon</div>
          <div>Tue</div>
          <div>Wed</div>
          <div>Thu</div>
          <div>Fri</div>
          <div>Sat</div>
          <div>Sun</div>
        </div>
        <div class="days" id="days"></div>
      </div>
    </div>
  </div>
```

Gambar 3. 29 Struktur HTML Halaman Schedule

Pada Gambar 3.29 merupakan struktur dari halaman schedule, menghasilkan susunan antarmuka kalender interaktif untuk menampilkan jadwal. Struktur utama kalender (container-kalender dan kalender) dimulai dengan sebuah header (header-kalender) yang berisi tombol navigasi "prevMonth" dan "nextMonth" untuk berpindah bulan, serta span untuk menampilkan bulan dan tahun yang sedang dilihat (monthYear). Di bawah header ini, terdapat baris statis yang menampilkan nama-nama hari dalam seminggu, dari Mon (Senin) hingga Sun (Minggu), menggunakan elemen div sederhana. Terakhir, elemen div dengan id="days" berfungsi sebagai placeholder dinamis, di mana tanggal-tanggal kalender dan jadwal pemesanan yang sebenarnya akan diisi menggunakan script JavaScript terpisah.

## 7) Halaman *Pending Request*

```
<div class="pending-card">
<!--  -->
<h3> Pending Request</h3>
<table class="booking-details">
<thead>
<tr>
<th>Room Name</th>
<th>Name</th>
<th>Division</th>
<th>Event</th>
<th>Date</th>
<th>Start Time</th>
<th>Finish Time</th>
<th>Action</th>
</tr>
</thead>
<tbody>
@forelse($bookings as $booking)
<tr>
<td>{{ $booking->ruangan->nama_ruangan ?? 'Ruang Rapat' }}</td>
<td>{{ $booking->nama_pemesan }}</td>
<td>{{ $booking->divisi }}</td>
<td>{{ $booking->event }}</td>
<td>{{ \Carbon\Carbon::parse($booking->tanggal)->format('d/m/Y') }}</td>
<td>{{ \Carbon\Carbon::parse($booking->jam_mulai)->format('H:i') }}</td>
<td>{{ \Carbon\Carbon::parse($booking->jam_selesai)->format('H:i') }}</td>
<td>
```

NUSANTARA

Gambar 3. 30 Struktur HTML Halaman Pending Request

Pada Gambar 3.30 merupakan struktur dari halaman *pending request*, menghasilkan tampilan untuk seluruh daftar pemesanan ruangan yang masih memerlukan persetujuan. Konten utama disajikan dalam sebuah kartu (*pending-card*) dengan judul "*Pending Request*" (<h3>). Data pemesanan ditampilkan dalam format tabel (<table> dengan kelas *booking-details*) yang mudah dibaca. Baris header tabel (<thead>) secara jelas mengidentifikasi kolom-kolom informasi yang akan disajikan, yaitu: Room Name, Name (Pemesan), Division, Event, Date, Start Time, Finish Time, dan Action (untuk persetujuan/penolakan). Isi tabel (<tbody>) kemudian diisi secara dinamis menggunakan loop (@forelse) yang akan mengambil detail pemesanan (\$booking) dari backend dan menampilkan data spesifik seperti nama ruangan, nama pemesan, divisi, dan nama acara.

## 8) Halaman *Activity Log*

```

<main class="box-2">
  <div class="main-header">
    <a href="{{ route('admin.akun') }}><p class="username"></p></a>
    <a href="{{ route('admin.akun') }}></a>
  </div>

  <div class="main-body">
    <div class="log-activity-container">
      <h2>User Log Activity</h2>
      <!-- <div class="log-table-wrapper"> -->
      <table class="log-table-wrapper">
        <thead>
          <tr>
            <th>Date</th>
            <th>User</th>
            <th>Activity</th>
            <th>Detail</th>
          </tr>
        </thead>
        <tbody>
          @forelse ($logs as $log)
            <tr>
              <td>{{ $log->created_at->format('d-m-Y H:i') }}</td>
              <td>{{ $log->user_name ?? 'System' }}</td>
              <td>{{ $log->activity }}</td>
              <td>{{ $log->details }}</td>
            </tr>
          @endforelse
        </tbody>
      </table>
    </div>
  </div>
</main>

```

Gambar 3. 31 Struktur HTML Halaman Activity Log

Pada Gambar 3.31 merupakan struktur dari halaman *activity log*, menghasilkan tampilan riwayat tindakan yang dilakukan oleh pengguna dalam sistem. Halaman ini memiliki judul "*User Log Activity*" (<h2>) dan menyajikan data aktivitas dalam format tabel.

*Header* tabel (*<thead>*) mendefinisikan kolom-kolom informasi yang jelas, yaitu: *Date* (Tanggal), *User* (Pengguna), *Activity* (Aktivitas), dan *Detail* (Rincian). Baris-baris data aktivitas (*<tbody>*) diisi secara dinamis menggunakan loop (*@forelse*) yang mengambil data log (\$logs) dari *backend*. Setiap baris menampilkan waktu pembuatan log (*created\_at*), nama pengguna yang melakukan aksi (*user\_name*), jenis aktivitas, dan rincian spesifik dari aktivitas tersebut. Jika tidak ada log aktivitas yang ditemukan, tabel akan menampilkan pesan "*No activity log*".

## 9) Halaman *User Management*

```

<div class="main-body">
    <!-- <h2>User Management</h2> -->
    <div class="user-management-actions">
        <button class="add-user-button" onclick="openAddUserModal()">
            <span class="plus-icon-button">+</span> Add New User
        </button>
    </div>

    <div class="user-list-container">
        <h3>User List</h3>
        <table class="user-table">
            <thead>
                <tr>
                    <th>No.</th>
                    <th>Email</th>
                    <th>Username</th>
                    <th>Role</th>
                    <th>Action</th>
                </tr>
            </thead>
            <tbody>
                @forelse($users as $index => $user)
                    <tr id="user-row-{{ $user->id }}">
                        <td>{{ $index + 1 }}</td>
                        <td>{{ $user->email }}</td>
                        <td>{{ $user->name }}</td>
                        <td>{{ $user->role }}</td>
                        <td>
                            ...
                        </td>
                    </tr>
                @forelse
            </tbody>
        </table>
    </div>

```

Gambar 3. 32 Struktur HTML Halaman User Management

Pada Gambar 3.32 merupakan struktur dari halaman *user management*, menghasilkan tampilan daftar seluruh pengguna sistem dalam format tabel yang terorganisir. Konten dibungkus dalam sebuah wadah utama (*user-list-container*) dengan judul "*User List*" (*<h3>*).

Informasi pengguna disajikan melalui elemen tabel (<table>) yang memiliki bagian header (<thead>) untuk mengidentifikasi kolom data secara berurutan, mulai dari nomor, alamat email, nama pengguna, hingga peran atau role pengguna tersebut. Bagian isi tabel (<tbody>) diproses secara dinamis menggunakan logika perulangan (@forelse) yang mengambil data pengguna dari sistem. Setiap baris data tidak hanya menampilkan informasi identitas pengguna, tetapi juga dilengkapi dengan kolom Action yang berisi tombol navigasi atau fungsi seperti tombol "Edit" yang memicu munculnya modal untuk melakukan pembaruan data pengguna secara spesifik.

## 10) Halaman *My Account*

```
<main class="box-2">
  <div class="main-header">
    <a href="{{ route('admin.akun') }}><p class="username"></p></a>
    <a href="{{ route('admin.akun') }}></a>
  </div>
  <div class="main-body">
    <div class="container-akun">
      <!--  -->
      <!-- FOTO PROFIL -->
      

      <!-- POPUP VIEWER -->
      <div id="popup-profile" class="popup-overlay" onclick="closePopup()">
        <div class="popup-content">
          <img id="popup-image" src="" alt="Profile Detail">
        </div>
      </div>
      <div class="info-akun">
        <h1 class="username"></h1>
        <label for="username">New Username</label>
        <div class="username-group">
          <input type="text" id="input-username" placeholder="Enter new username">
          <button class="btn-save" onclick="saveName()">Save</button>
        </div>
        <label for="file">Photo Profile</label>
        <div class="Photo-profile">
          <input type="file" id="input-file">
          <button class="btn-save" onclick="savePhoto()">Save</button>
        </div>
      </div>
    </div>
  </div>
</main>
```

Gambar 3. 33 Struktur HTML Halaman My Account

Pada Gambar 3.34 merupakan struktur dari halaman *my account*, menghasilkan tampilan melihat informasi profil sekaligus melakukan pembaruan identitas akun mereka. Struktur halaman dibagi menjadi dua bagian utama, yaitu bagian *header* yang menampilkan informasi singkat seperti nama pengguna dan foto profil kecil, serta bagian konten utama yang difokuskan pada pengelolaan detail profil pengguna secara mendalam. Pada bagian konten, terdapat elemen

untuk menampilkan foto profil dalam ukuran besar yang dilengkapi dengan fitur penampil *popup* untuk melihat rincian foto secara lebih detail. Di bawah informasi visual tersebut, tersedia kolom input teks yang memungkinkan pengguna memasukkan nama pengguna baru dan sebuah tombol simpan untuk memperbarui data tersebut ke dalam sistem.

#### 3.3.1.4 Testing

Pada bagian ini mahasiswa melakukan 2 (dua) tahap testing, tahap pertama testing pada saat seluruh HTML, CSS, dan JavaScript sudah selesai dikembangkan serta seluruh fitur-fitur utama yang sudah bisa melakukan interaksi. Pengujian testing ini dilakukan pada tim IT Application System termasuk user atau supervisor mahasiswa.

Hasil dari testing pertama ditemukan beberapa *bug* atau *error* pada halaman *user management*, *bug* atau *error* yang ditemukan pada halaman tersebut adalah pada fungsi edit. Ketika admin (*testing*) ingin mengubah *role* misalnya dari *user* menjadi *approver*, pada saat admin menekan tombol *save* hasil di halaman user management *role* tidak berubah dan ketika mencoba login sesuai dengan akun yang sudah dibuat tetap menjurus ke halaman *role* yang awal.

Selanjutnya setelah menemukan *bug* atau *error* yang ditemukan dilakukan revisi pada bagian-bagian tertentu yang berhubungan dengan *bug* atau *error*. Tahap lanjutannya setelah selesai melakukan tahap revisi, tim IT Application System melakukan testing terakhir untuk memastikan bahwa fitur dari *bug* atau *error* yang ditemukan sudah terselesaikan dan dapat berfungsi sesuai dengan fungsinya.

#### 3.3.2 Kendala yang Ditemukan

Selama periode magang berlangsung, terdapat beberapa kendala dan kesulitan yang dihadapi, baik dari aspek teknis maupun non-teknis. Kendala-kendala ini memengaruhi kelancaran proses dalam menyelesaikan tanggung

jawab. Meskipun demikian, kendala yang ditemukan menjadi bagian penting dalam pengembangan karakter untuk mahasiswa selama periode magang berlangsung. Berikut terkait kendala yang ditemukan selama pelaksanaan magang:

**1) Pemahaman terhadap proyek yang sudah berjalan pada bulan Juli 2025.**

Pada pelaksanaan proyek pengembangan website aplikasi pemesan ruang rapat ini sebelumnya sudah jalan pada bulan Juli 2025, tetapi hanya sampai struktur dasar dari website. Contohnya seperti halaman login yang masih berbentuk struktur standar ada input email dan password serta lupa password. Hal ini menjadikan UI/UX dari website tersebut kurang menarik serta interaksi yang kurang user friendly. Kondisi ini menghambat pemahaman dalam proyek yang akan dihasilkan.

**2) Alur proses bisnis pada pengembangan proyek website aplikasi yang belum terdokumentasi.**

Pada proses ini belum adanya dokumentasi alur proses bisnis secara sistematis, seperti pembuatan use case diagram dan activity diagram. Hal ini menyebabkan proses identifikasi kebutuhan, alur interaksi pengguna, serta tahapan aktivitas sistem belum tergambarkan secara jelas dalam bentuk standar pengembangan sistem. Dokumen visualisasi diagram yang seharusnya dibuat menggunakan tools draw.io belum tersedia, sehingga mahasiswa membuat dari awal dengan diskusi bersama user terkait.

### **3.3.3 Solusi atas Kendala yang Ditemukan**

Langkah berikutnya setelah menemukan kendala-kendala yang ditemukan selama pelaksanaan kegiatan magang adalah mahasiswa mencari solusi dan mengevaluasi dari kendala yang ditemukan. Setelah mengevaluasi dari kendala yang ditemukan, mahasiswa menemukan solusi atas kendala tersebut. Berikut solusi yang diambil untuk menghadapi kendala tersebut:

## **1) Koordinasi dan review menyeluruh terhadap struktur dasar website yang telah dibuat.**

Solusi dari kendala yang ditemukan pertama adalah melakukan koordinasi dengan user atau supervisi dalam memahami alur kerja, tujuan pengembangan, serta ekspetasi mengenai hasil akhir. Selain melakukan koordinasi, mahasiswa melakukan perbaikan terhadap desain UI/UX menggunakan aplikasi figma agar lebih terdokumentasi dan terupdate secara real time serta dapat dipantau oleh satu tim. Perbaikan desain UI/UX ini mengacu pada prinsip desain antarmuka dan contoh referensi aplikasi serupa agar tampilan dan interaksi pengguna menjadi lebih baik dan mudah dipahami oleh pengguna. Proses ini membantu dalam mempercepat adaptasi terhadap pengerjaan proyek yang sedang berjalan.

## **2) Diskusi dan membuat alur proses bisnis menggunakan tools draw.io**

Proses pembuatan use case diagram dan activity diagram melalui diskusi serta koordinasi dengan user dan tim yang terlibat. Diskusi ini memastikan bahwa setiap kebutuhan dan interaksi dalam sistem teridentifikasi dengan benar. Diagram yang dibuat kemudian akan dijadikan dasar untuk tahapan pengembangan berikutnya agar alur kerja sistem menjadi lebih terstruktur, jelas, dan dapat dijadikan acuan pada proses pengembangan kedepannya.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA