

BAB III

PELAKSANAAN KERJA PROFESI

3.1 Bidang Kerja

Bidang kerja yang dijalankan oleh Praktikan selama kerja profesi berkaitan dengan Informasi dan Teknologi, khususnya *software development*. Praktikan banyak bersinggungan dengan aktivitas-aktivitas *software development* seperti pengumpulan dan finalisasi kebutuhan pengguna, menentukan dan mendesain arsitektur dari aplikasi, melakukan pengkodean terhadap aplikasi yang dijalankan, serta melakukan pengetesan terhadap aplikasi yang sedang dikembangkan.

Sesuai dengan jabatan dan posisi saya sebagai *Junior Web Developer* di PT. Perkasa Pilar Utama, adapun tugas dan lingkup pekerjaan yang saya kerjakan adalah sebagai berikut :

1. Mengembangkan perangkat lunak sesuai dengan pedoman dan standar perusahaan
2. Mengkomunikasikan setiap kendala dan kesulitan yang dihadapi selama proses pengembangan
3. Memastikan perangkat lunak yang dikembangkan sudah sesuai dengan kebutuhan pengguna
4. Memastikan perangkat lunak minim *bug*
5. Menampung, mendiskusikan, serta mengimplementasikan apa yang menjadi saran untuk perbaikan dari pengguna terkait aplikasi yang sedang dikembangkan
6. Mempelajari dan melakukan eksplorasi seputar teknologi terbaru khususnya di bidang *Web App Development* dan *Mobile App Development*
7. Mampu memahami dan menangani *error* dan *bug* yang muncul dari aplikasi yang sedang dikembangkan.

3.2 Pelaksanaan Kerja

3.2.1. Tentang Platform dan Aplikasi

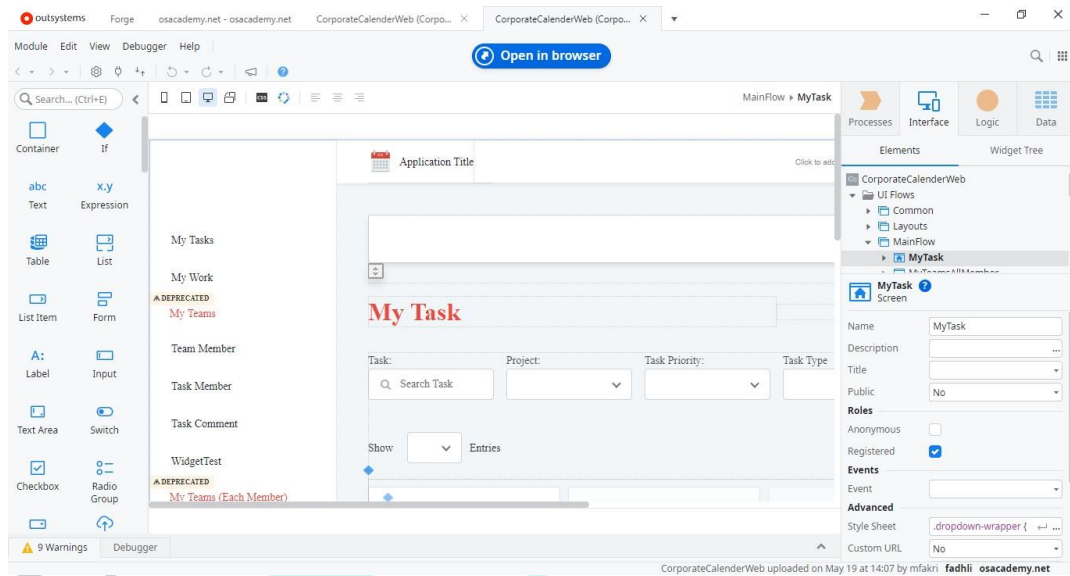
a. Tentang Platform

Dalam masa menjalani KP, Praktikan menggunakan platform *low-code* (rendah pengkodean) yang bernama Outsystems. Outsystems sendiri telah ada dan dikembangkan sejak tahun 2001 oleh seseorang bernama Paulo Rosado. Outsystems dibuat dengan tujuan untuk memangkas waktu pengembangan agar lebih cepat dan menyesuaikan penggunaan metode pengembangan perangkat lunak pada saat ini. Platform ini sendiri bersifat *close-source*, namun menyediakan akun gratis untuk para pengembang yang ingin mencoba menggunakan platform Outsystems untuk membuat aplikasi. Akun gratis tersebut sudah memiliki kapasitas basis data sampai dengan 2 GB.

Outsystems sendiri dapat berjalan di dua jenis *environment* yaitu *on-cloud* dan *on-premise*. PT. Perkasa Pilar Utama sendiri memiliki dua *environment* yang digunakan dalam pengembangan aplikasi. *Environment* pertama menggunakan IIS Windows Server yang berjalan secara *on-cloud* dan dikenal dengan domain net paser. Net paser sendiri digunakan sebagai *Development Environment*. Satu server lagi berjalan secara *on-premise* yang dikelola secara internal oleh PT. Perkasa Pilar Utama dan digunakan sebagai *Production Environment*.

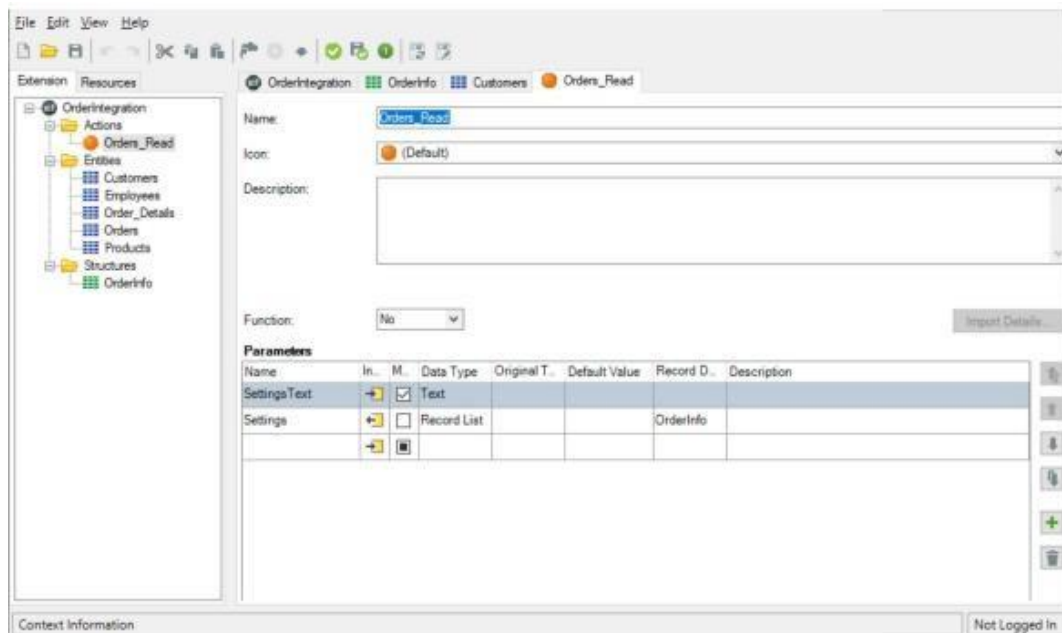
Outsystems memiliki empat alat yang digunakan. Keempat alat pengembangan tersebut yaitu :

1. *Service Studio* : merupakan sebuah *Integrated Development Environment* (IDE) yang khusus digunakan untuk mengembangkan aplikasi menggunakan Outsystems. Dengan *service studio*, pengembang bisa mengembangkan beberapa jenis aplikasi seperti *Reactive Web App*, *Tablet App* dan *Mobile App*.



Gambar 3.1 Service Studio
 Sumber : Hasil dokumentasi Praktikan

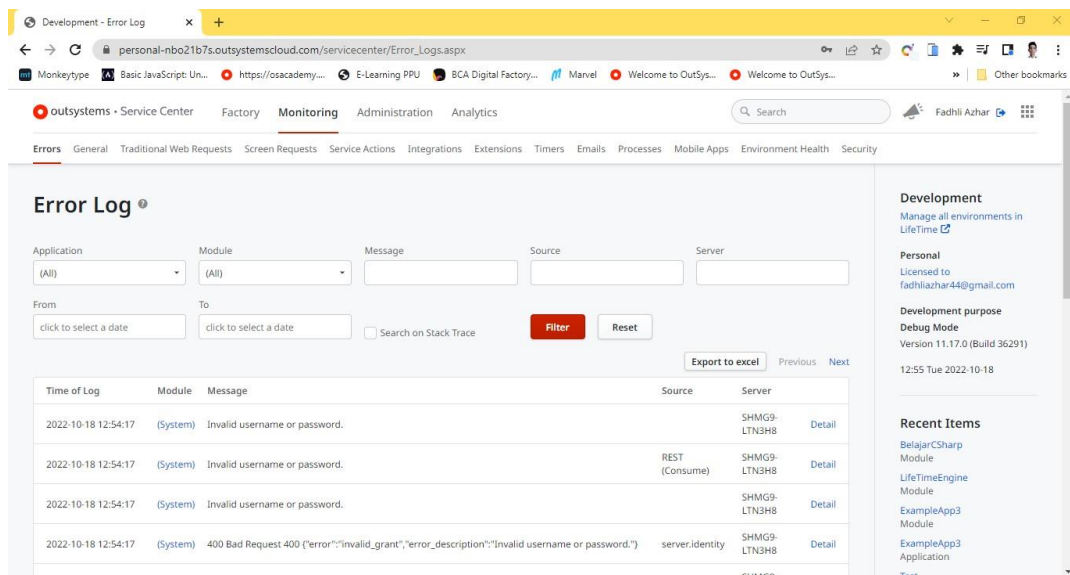
2. *Integration Studio* : merupakan perangkat lunak yang digunakan untuk membuat kustom komponen dan melakukan integrasi dengan basis data eksternal. Kustom komponen dan basis data eksternal yang sudah diintegrasikan melalui *integration studio* merupakan sebuah modul ekstensi dan bisa digunakan oleh aplikasi di *service studio*.



Gambar 3.2 Integration Studio

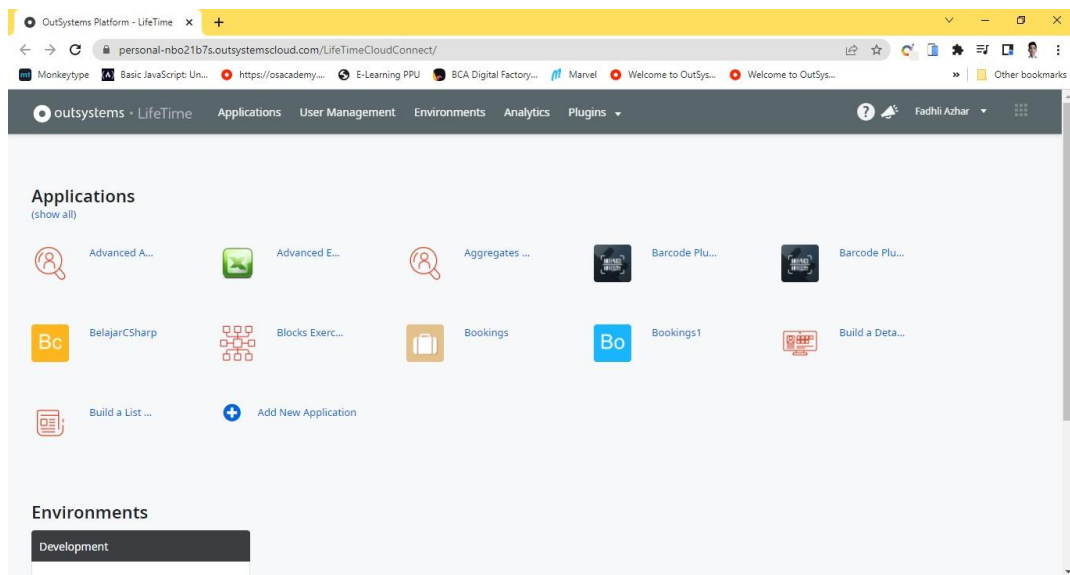
Sumber : Hasil dokumentasi Praktikan

3. *Service Center* : merupakan salah satu alat dari platform Outsystems yang digunakan untuk memantau aplikasi-aplikasi yang sudah kita buat. Service center bisa diakses melalui browser dengan akun Outsystems. Di dalam *service center*, kita bisa melihat *log* dari setiap aplikasi, membuat dan menjalankan *timer* (sebuah pewaktu untuk mengeksekusi sebuah aksi di dalam aplikasi dalam waktu yang sudah ditentukan), mempersiapkan koneksi ke eksternal basis data, mempersiapkan konektivitas *e-mail*, dan masih banyak fitur administratif yang bisa dilihat di dalam *service center*.



Gambar 3.3 Service Center
 Sumber : Hasil dokumentasi Praktikan

4. *Lifetime* : merupakan sebuah konsol terpadu untuk melihat semua lingkungan infrastruktur yang ada. *Lifetime* sendiri bisa diakses melalui browser dengan akun Outsystems. Beberapa fitur yang bisa digunakan di dalam *lifetime* antara lain penerapan dan konfigurasi aplikasi, pengembangan dan penerapan perbaikan terbaru (*hotfix & deployment*), *User Management*, membuat *plugin*, menganalisa performa dari setiap *environment* serta konfigurasi keamanan terhadap *environment* dan aplikasi-aplikasi yang ada.



Gambar 3.4 Lifetime

Sumber : Hasil dokumentasi Praktikan

Outsystems sendiri memiliki lima jenis modul yang dapat digunakan dalam mengembangkan aplikasi. Masing-masing dari kelima modul tersebut memiliki fungsi yang berbeda-beda di dalam aplikasi. Berikut ini lima modul yang disediakan oleh Outsystems :

1. **Reactive Web App** : merupakan modul dengan komponen yang paling lengkap. Di dalam modul sudah tersedia beberapa *UI Template* yang bisa digunakan pengembang untuk mengembangkan aplikasi. Modul ini dianjurkan oleh Outsystems sebagai modul *front-end* aplikasi dan pengembang biasanya menggunakan modul ini untuk merancang tampilan dari aplikasi.
2. **Blank** : merupakan modul yang isinya sama persis dengan modul *Reactive Web App*. Hanya saja modul ini tidak dilengkapi *UI Template*. Jadi pengembang bisa melakukan dan membuat tampilan dari kosongan. Biasanya modul ini digunakan oleh pengembang yang suka melakukan kustomisasi komponen tampilan pada aplikasi.
3. **Service** : merupakan modul yang disediakan Outsystem untuk pengembang melakukan perancangan basis data. Sebetulnya hal ini bisa dilakukan di dalam modul *Reactive Web App* dan modul *Blank*. Namun hal ini tidak dianjurkan oleh Outsystems karena akan menyulitkan perbaikan dan larangan untuk mengonsumsi atau melakukan dependensi ke *UI Module (Reactive Web App & Blank Module)*. Di dalam modul ini pengembang bisa membuat basis data, membuat table, menggambar sebuah ERD dari setiap tabel, serta membuat struktur. Selain konfigurasi basis data, pengembang

bisa membuat *logic & function*, mengonsumsi servis eksternal (REST API & SOAP) serta melakukan integrasi dengan sistem SAP.

4. **Library** : merupakan modul yang disediakan oleh Outsystem untuk pengembang membuat sebuah *custom widget*. Nantinya *custom widget* ini bisa digunakan secara berulang oleh aplikasi-aplikasi lain dengan cara menghubungkan dependensi dengan *library module* tersebut. Namun di dalam modul ini tidak tersedia fitur untuk melakukan konfigurasi basis data seperti
5. **Extension** : modul ini nantinya akan dibuat di dalam *Integration Studio*. Ketika pengembang memilih jenis modul ini di dalam *service studio*, maka *service studio* secara otomatis membuka *integration studio*. *Extension module* ini disediakan oleh Outsystems untuk pengembang membuat *custom logic* dan *custom function* serta melakukan integrasi dengan basis data eksternal. Untuk membuat *custom logic & custom function*, pengembang harus menguasai bahasa pemrograman C# (C-sharp). Hal itu dikarenakan aplikasi yang dibuat oleh Outsystems didukung oleh bahasa pemrograman C# disisi *back-end*. Pengembang juga diharuskan melakukan instalasi Visual Studio untuk membuat *custom logic & custom function*.

Nantinya kelima modul ini bisa dikonsumsi oleh aplikasi yang ingin menggunakan fungsi dan komponen yang ada di dalam modul tersebut.

b. Tentang Aplikasi

Aplikasi yang dirancang oleh Praktikan tentunya juga menggunakan platform Outsystems. Aplikasi yang dirancang saat Pratikkan melaksanakan KP adalah aplikasi yang berfungsi untuk manajemen proyek yang sedang berjalan dan sedang dikerjakan di PT. Perkasa Pilar Utama.

Tujuan dari perancangan aplikasi ini adalah agar perusahaan tidak lagi menggunakan dan menyewa aplikasi manajemen proyek eksternal. Aplikasi ini dinamakan *Corporate Calendar*. *Corporate Calendar* sendiri merupakan aplikasi berbasis web yang dirancang dan dibangun menggunakan Outsystems serta terdiri dari lima modul di dalamnya. Berikut penjelasan dari modul-modul yang ada di dalam aplikasi *Corporate Calendar* :

1. **CorporateCalendar_CS (Service Module)**

Modul ini dibuat dan digunakan untuk melakukan konfigurasi basis data dari aplikasi.

2. **CorporateCalendar_TH (Reactive Web App Module)**

Modul ini dibuat dan digunakan untuk membuat dan melakukan *setup* tema dari aplikasi. Tema yang dibuat menggunakan CSS untuk mempercantik tampilan dari aplikasi. Nantinya modul ini akan didependensikan dan digunakan oleh modul CorporateCalendarWeb.

3. **CorporateCalendarBackOffice (Reactive Web App Module)**

Modul ini berisikan fungsi-fungsi yang dibuat untuk menjalankan setiap proses bisnis yang ada di dalam aplikasi. Nantinya modul ini juga akan didependensikan dan digunakan oleh modul CorporateCalendarWeb.

4. **CorporateCalendarTEST (Reactive Web App Module)**

Modul ini merupakan modul yang digunakan untuk fungsi-fungsi serta logika yang masih dalam tahap eksplorasi, namun belum tentu akan diimplementasikan ke dalam modul CorporateCalendarWeb.

5. **CorporateCalendarWeb (Reactive Web App Module)**

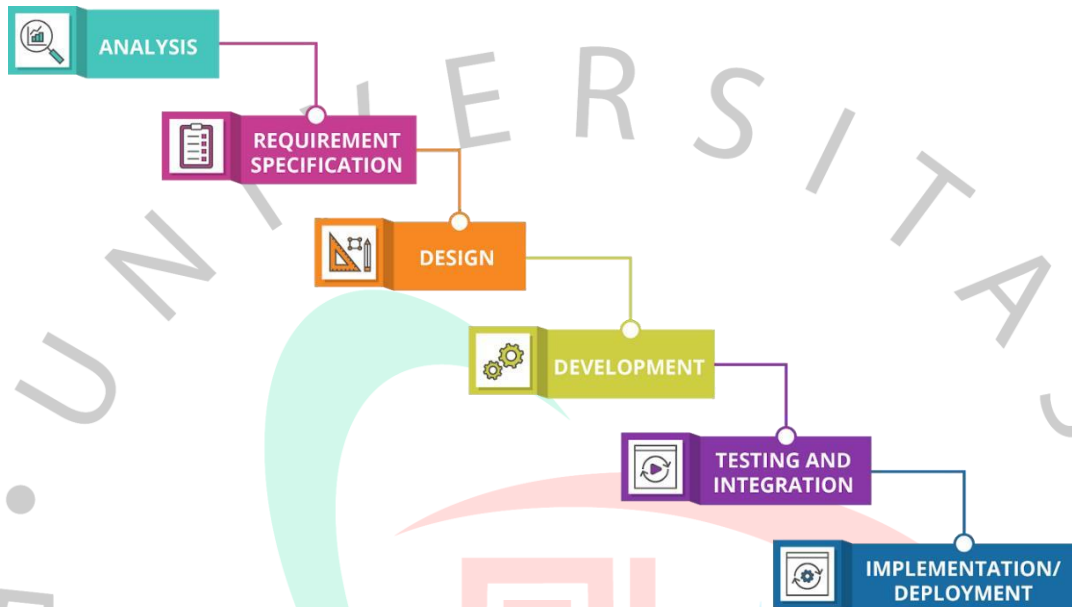
Modul ini merupakan modul utama dari aplikasi dan merupakan *UI Module*. Modul ini berisi beberapa modul yang sudah didependensikan ke dalamnya. Nantinya modul-modul yang sudah didependensikan, akan dipakai *logic* dan *functionnya*. Selain itu, modul ini juga mengonsumsi modul CorporateCalendar_CS untuk kebutuhan segala aktivitas yang berhubungan dengan data.

3.2.2. **Metode Pengembangan Aplikasi**

a. *Metode Pengembangan Waterfall*

Dalam pembuatan aplikasi *Corporate Calendar* sendiri, *Project Manager* selaku yang memimpin dan mengawasi pengerjaan proyek memilih menggunakan metode pengembangan *waterfall*. Metode pengembangan *waterfall* dipilih karena ingin meminimalisir kesalahan yang terjadi selama proses pengembangan serta menghasilkan aplikasi yang stabil. Berikut

ini gambaran dari tahapan pengembangan menggunakan metode pengembangan *waterfall* :



Gambar 3.5 Metode Pengembangan Waterfall

Sumber : <https://baharzahdhani.medium.com/betapa-epik-sdlc-software-development-life-cycle-3cbdccf000be>

Berikut ini penjelasan detail dari setiap fase pengembangan yang telah dijalani Praktikan selama proses pengembangan aplikasi :

1. **Analysis & Requirement Specification**

Pada fase ini, Praktikan bersama dengan para *stakeholder* dan seorang *technical leader* melakukan pengumpulan kebutuhan untuk aplikasi yang ingin dikembangkan. Pengumpulan kebutuhan dilakukan dengan cara melakukan diskusi secara daring. Selain proses pengumpulan kebutuhan, hal lain yang dibahas adalah hasil akhir yang ingin dicapai dari para *stakeholder* terhadap aplikasi yang sedang dikembangkan.

2. **Design**

Pada fase ini, Praktikan melakukan perancangan antarmuka dari setiap halaman yang ada di dalam aplikasi.

Selain merancang bagian antarmuka, Praktikan juga melakukan perancangan dan penjabaran tabel yang akan dibuat di dalam basis data. Hal ini bertujuan untuk menjabarkan kebutuhan data dari setiap logika bisnis yang ada di dalam aplikasi.

3. **Development**

Pada fase ini, Praktikan melakukan pengkodean atau biasa disebut dengan *coding*. Proses *coding* sendiri merupakan menerjemahkan apa yang sudah dilakukan pada fase desain ke dalam bahasa program. Selama menjalani fase ini, Praktikan juga sering berdiskusi dengan *technical leader* terkait *best practices* ketika membuat sebuah *logic* dan juga *function*.

4. **Testing & Integration**

Pada fase ini, Praktikan melakukan pengetesan terhadap aplikasi. Namun proses pengetesan di sini hanya sebatas validasi yang telah dibuat oleh Praktikan di dalam aplikasi. Prosedur pengetesan yang lebih lengkap akan diserahkan kepada tim *Quality Assurance*. Selain melakukan pengetesan terhadap aplikasi, Praktikan juga melakukan integrasi dari setiap modul yang telah dibuat.

5. **Implementation / Deployment**

Fase ini merupakan fase terakhir dari metode pengembangan *waterfall*. Pada fase ini, Praktikan melakukan *deployment* atau pemindahan aplikasi dari *development server* ke *production server*. Tentunya proses *deployment* sendiri sudah bisa dilakukan ketika fase pengetesan aplikasi sudah selesai dan tidak lagi ditemukan *bug* ataupun *error*. Aplikasi yang sudah dipindahkan ke *production server* sudah siap digunakan

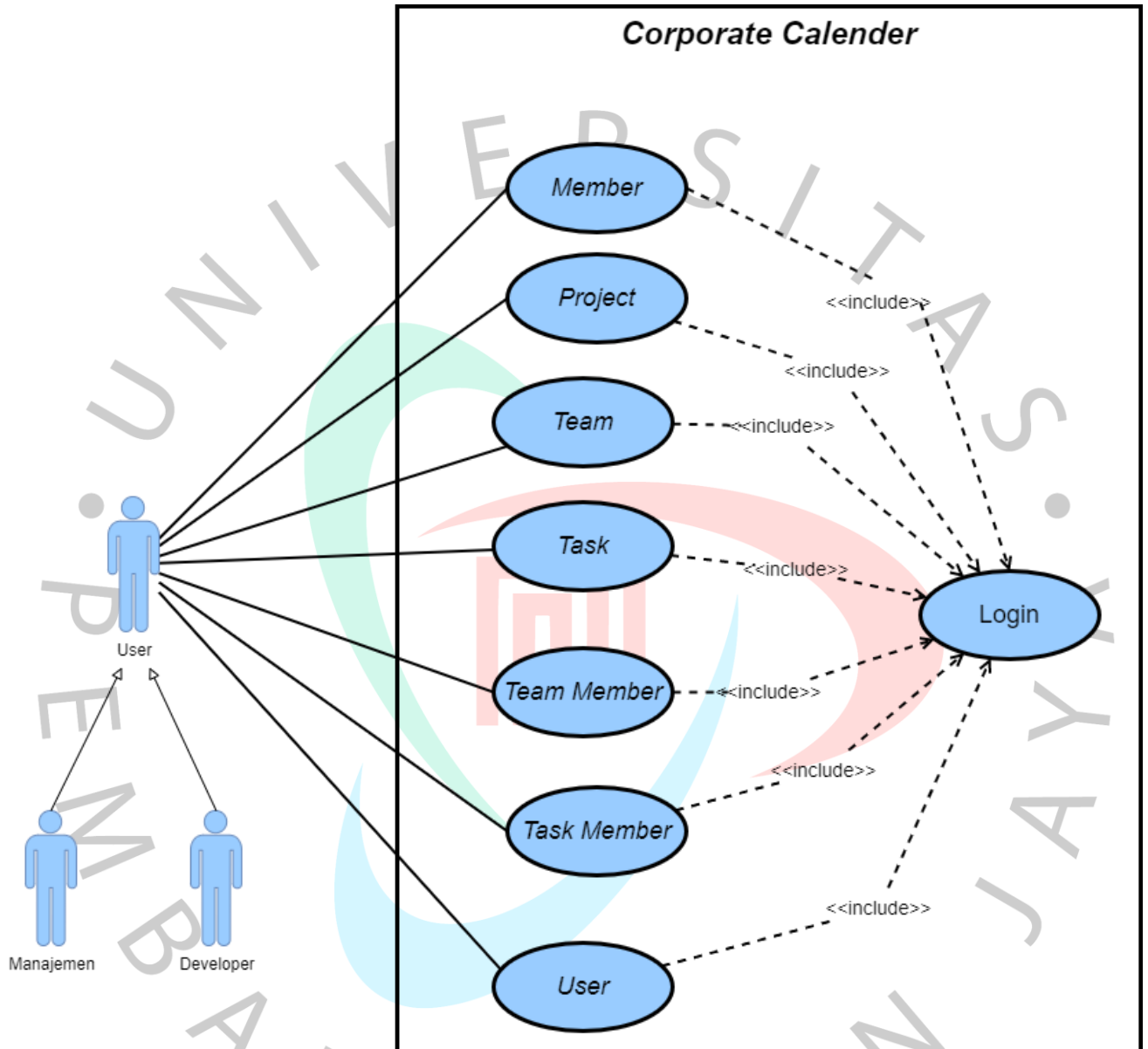
b. *Timeline* Pengerjaan Aplikasi

Tabel 3.1 *Timeline* Pengerjaan Aplikasi

NO	Kegiatan Pengembangan	JUNI				JULI				AGUSTUS	
		1	2	3	4	1	2	3	4	1	2
1	Analisa & Pengumpulan Kebutuhan Pengguna	■									
2	Perancangan Basis Data		■	■	■						
3	Pembuatan <i>Mock-up</i>				■						
4	Pengembangan Aplikasi					■	■	■	■		
5	Tes & Integrasi									■	
6	Implementasi										■

3.2.3. Unified Modelling Language (UML)

a. Use Case Diagram



Gambar 3.6 Use Case Aplikasi Corporate Calendar

Sumber : Hasil dokumentasi Praktikan

Pada aplikasi Corporate Calendar, terdapat dua aktor yaitu Admin dan *Member*. Berikut ini penjelasan dan interaksi yang terjadi antara aktor dan sistem (*interface*) :

a) Manajemen

Manajemen sebagai aktor di dalam aplikasi mempunyai peran dan otorisasi lebih banyak di dalam aplikasi. Peran aktor Manajemen di dalam aplikasi Corporate Calendar antara lain mengelola *use*, *member*, *team*, *team member*, *task* dan *task member*.

b) *Developer*

Berbeda dengan Manajemen, *Developer* sebagai aktor kedua di dalam aplikasi, memiliki peran dan otorisasi lebih sedikit di dalam aplikasi Corporate Calendar. Peran *Developer* di dalam aplikasi Corporate Calendar antara lain yaitu menambahkan *Task*, menambahkan *Task Comment*, melihat informasi *My Teams*, melihat informasi *Team Member* dan melihat informasi *Task Member*.

b. Spesifikasi *Use Case Diagram*

Spesifikasi *use case diagram* merupakan penjelasan dari gambar *use case* pada point sebelumnya:

a. Login

Tabel 3.2 Spesifikasi *Use Case Login*

UC-1	Login
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Mengakses aplikasi Corporate Calendar
Pre-conditions	Pengguna sudah teregistrasi di aplikasi Corporate Calendar
Post-conditions	Masuk ke halaman utama dari aplikasi Corporate Calendar
Main Success Scenario	1. Pengguna mengakses aplikasi melalui peramban. 2. Pengguna memasukkan username pada form login.

3. Pengguna memasukkan kata sandi pengguna pada form login.
4. Melakukan submit pada form login.
5. Aplikasi melakukan validasi terhadap pengguna yang masuk ke dalam aplikasi. Proses penyamaan kata sandi yang diinputkan pengguna dengan kata sandi yang ada di dalam database telah melalui proses enkripsi terlebih dahulu.
6. Jika akun pengguna lolos proses validasi, maka sistem akan mengarahkan pengguna ke halaman utama

Extension / Alternate Scenario

1. Saat pengguna salah memasukkan antara username atau kata sandi, atau saat akun pengguna tidak ditemukan, maka aplikasi akan menampilkan pesan kesalahan pada halaman login.
2. Pengguna tetap berada di halaman login.

Priority *High*

Special Requirements -

Open Questions -

b. Pendaftaran Pengguna

Tabel 3.3 Spesifikasi *Use Case* Pendaftaran Pengguna

UC-9	Pendaftaran <i>User</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Registrasi pengguna baru
Pre-conditions	Pengguna belum teregistrasi di aplikasi
Post-conditions	Pengguna baru telah terdaftar dan memiliki hak akses di dalam aplikasi
Main Success Scenario	<ol style="list-style-type: none"> 1. <include:login> 2. Membuka Menu <i>User</i> 3. Mendaftarkan pengguna 4. Aplikasi melakukan validasi terhadap data pengguna 5. Melakukan <i>refresh</i> halaman. 6. Pengguna dapat login ke dalam aplikasi
Extension / Alternate Scenario	<ol style="list-style-type: none"> 1. Pendaftaran pengguna gagal karena pengguna telah terdaftar di aplikasi 2. Pendaftaran pengguna gagal karena data yang diinputkan tidak valid
Priority	<i>High</i>
Special Requirements	<i>Simpan kata sandi pengguna dalam bentuk enkripsi</i>
Open Questions	-

c. Menambahkan Data *Project*

Tabel 3.4 Spesifikasi *Use Case* Menambahkan Data *Project*

UC-3	Menambahkan Data <i>Project</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Mendaftarkan proyek baru
Pre-conditions	-
Post-conditions	Proyek baru telah tersimpan ke <i>database</i>
Main Success Scenario	1. <include:login>

2. Membuka Menu *Project*
3. Meinputkan informasi proyek
4. Melakukan submit
5. Aplikasi menyimpan data
6. Jika proses simpan data berhasil, akan tampil pesan bahwa data telah tersimpan
7. Aplikasi melakukan *refresh* halaman.
8. Manajemen dapat melihat informasi proyek yang baru ditambahkan

Extension / Alternate Scenario	1. Akan tampil pesan kesalahan jika proses simpan data gagal
Priority	<i>Medium</i>
Special Requirements	-
Open Questions	-

d. Melihat Data *Project*

Tabel 3.5 Spesifikasi *Use Case* Melihat Data *Project*

UC-3	Melihat Data <i>Project</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Melihat data <i>project</i> yang sudah pernah ditambahkan
Pre-conditions	-
Post-conditions	-
Main Success Scenario	<ol style="list-style-type: none"> 1. <include:login> 2. Membuka Menu <i>Project</i> 3. Aplikasi informasi <i>project</i> yang ada di halaman Menu <i>Project</i>
Extension / Alternate Scenario	-
Priority	<i>Medium</i>
Special Requirements	-
Open Questions	-

e. Mengubah Data *Project*

Tabel 3.6 Spesifikasi *Use Case* Mengubah Data *Project*

UC-3	Mengubah Data <i>Project</i>
Actor	Pengguna

Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Mengubah data <i>project</i> yang sudah pernah ditambahkan
Pre-conditions	Data <i>project</i> sudah pernah ditambahkan ke dalam aplikasi
Post-conditions	Data <i>project</i> berhasil diubah

Main Success Scenario

1. <include:login>
2. Membuka Menu *Project*
3. Aplikasi informasi *project* yang ada di halaman Menu *Project*
4. Pengguna memilih data *project* yang ingin diubah
5. Pengguna mengklik ikon edit pada baris data yang dipilih
6. Aplikasi menampilkan form yang berisi data *project* yang ingin diubah
7. Pengguna melakukan perubahan pada data *project*
8. Jika sudah selesai melakukan perubahan, pengguna menekan tombol "Save"
9. Aplikasi menampilkan pesan sukses jika data *project* berhasil diubah

Extension / Alternate Scenario

1. Aplikasi akan menampilkan pesan kesalahan jika gagal menyimpan perubahan data *project* yang dilakukan pengguna
2. Pengguna batal melakukan proses ubah data *project* dengan menekan tombol "Cancel" pada form

Priority	<i>Medium</i>
Special Requirements	-
Open Questions	-

f. Menghapus Data *Project*

Tabel 3.7 Spesifikasi Use Case Menghapus Data *Project*

UC-3	Menghapus Data <i>Project</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Menghapus data <i>project</i> yang sudah pernah ditambahkan

Pre-conditions	Data <i>project</i> sudah pernah ditambahkan ke dalam aplikasi
Post-conditions	Data <i>project</i> berhasil dihapus
Main Success Scenario	<ol style="list-style-type: none"> 1. <include:login> 2. Membuka Menu <i>Project</i> 3. Aplikasi informasi <i>project</i> yang ada di halaman Menu <i>Project</i> 4. Pengguna memilih data <i>project</i> yang ingin dihapus 5. Pengguna mengklik ikon hapus pada baris data yang dipilih 6. Aplikasi menampilkan modal konfirmasi untuk melanjutkan proses hapus data 7. Pengguna menekan tombol "Yes" pada modal untuk menghapus data 8. Aplikasi menampilkan pesan sukses jika data <i>project</i> berhasil dihapus
Extension / Alternate Scenario	<ol style="list-style-type: none"> 1. Aplikasi akan menampilkan pesan kesalahan jika gagal menghapus data <i>project</i> yang dilakukan pengguna 2. Pengguna batal menghapus data <i>project</i> dengan menekan tombol "Cancel"
Priority	<i>Medium</i>
Special Requirements	-
Open Questions	-

g. Menambahkan Data *Team*

Tabel 3.8 Spesifikasi *Use Case* Menambahkan Data *Team*

UC-4	Menambahkan Data <i>Team</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Mendaftarkan tim baru
Pre-conditions	-
Post-conditions	Tim baru telah tersimpan ke dalam <i>database</i>
Main Success Scenario	<ol style="list-style-type: none"> 1. <include:login> 2. Membuka Menu <i>Team</i> 3. Menginputkan informasi tim 4. Melakukan submit 5. Aplikasi menyimpan data 6. Jika proses simpan data berhasil,

akan tampil pesan bahwa data telah tersimpan
 7. Aplikasi melakukan *refresh* halaman.
 8. Manajemen dapat melihat informasi tim yang baru ditambahkan

Extension / Alternate Scenario	1. Akan tampil pesan kesalahan jika proses simpan data gagal
Priority	<i>Medium</i>
Special Requirements	-
Open Questions	-

h. Melihat Data *Team*

Tabel 3.9 Spesifikasi *Use Case* Melihat Data *Team*

UC-4	Melihat Data <i>Team</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Melihat informasi data <i>team</i> yang sudah ditambahkan ke dalam aplikasi
Pre-conditions	-
Post-conditions	-
Main Success Scenario	<ol style="list-style-type: none"> 1. <include:login> 2. Membuka Menu <i>Teams</i> 3. Aplikasi informasi <i>teams</i> yang ada di halaman Menu <i>Teams</i>
Extension / Alternate Scenario	-
Priority	<i>Medium</i>
Special Requirements	-
Open Questions	-

i. Mengubah Data *Team*

Tabel 3.10 Spesifikasi *Use Case* Mengubah Data *Team*

UC-4	Mengubah Data <i>Team</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource

Trigger	Mengubah data <i>team</i> yang sudah ditambahkan ke dalam aplikasi
Pre-conditions	Data <i>team</i> sudah pernah ditambahkan ke dalam aplikasi
Post-conditions	Data <i>team</i> berhasil diubah
Main Success Scenario	<ol style="list-style-type: none"> 1. <include:login> 2. Membuka Menu <i>Teams</i> 3. Aplikasi informasi <i>teams</i> yang ada di halaman Menu <i>Teams</i> 4. Pengguna memilih baris data yang ingin diubah lalu menekan tombol edit 5. Aplikasi menampilkan form yang berisi data <i>team</i> 6. Pengguna melakukan perubahan pada <i>field</i> nama <i>team</i> 7. Pengguna menekan tombol "Save" 8. Aplikasi menyimpan perubahan data <i>team</i> 9. Aplikasi menampilkan pesan sukses jika perubahan data yang dilakukan pengguna berhasil disimpan
Extension / Alternate Scenario	<ol style="list-style-type: none"> 1. Aplikasi menampilkan pesan kesalahan jika perubahan data yang dilakukan pengguna gagal disimpan 2. Pengguna batal melakukan perubahan data <i>team</i> dengan menekan tombol "Cancel"
Priority	<i>Medium</i>
Special Requirements	-
Open Questions	-

j. Menghapus Data *Team*

Tabel 3.11 Spesifikasi *Use Case* Menghapus Data *Team*

UC-4	Menghapus Data <i>Team</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Menghapus data <i>team</i> yang sudah ditambahkan ke dalam aplikasi
Pre-conditions	Data <i>team</i> sudah pernah ditambahkan ke dalam aplikasi
Post-conditions	Data <i>team</i> berhasil dihapus dari aplikasi
Main Success Scenario	<ol style="list-style-type: none"> 1. <include:login> 2. Membuka Menu <i>Teams</i> 3. Aplikasi informasi <i>teams</i> yang ada di halaman Menu <i>Teams</i> 4. Pengguna memilih baris data yang

ingin dihapus lalu menekan tombol dengan ikon hapus
 5. Aplikasi menampilkan modal konfirmasi untuk proses hapus
 6. Pengguna menekan tombol "Yes" untuk mengkonfirmasi proses hapus
 7. Aplikasi menghapus data dari aplikasi
 8. Aplikasi menampilkan pesan sukses jika proses hapus data yang dilakukan pengguna berhasil

Extension / Alternate Scenario

1. Aplikasi menampilkan pesan kesalahan jika proses hapus data yang dilakukan pengguna gagal
 2. Pengguna batal melakukan perubahan data *team* dengan menekan tombol "Cancel" pada modal konfirmasi

Priority *Medium*

Special Requirements -

Open Questions -

k. Menambahkan Data *Team Member*

Tabel 3.12 Spesifikasi Use Case Menambahkan Data *Team Member*

UC-6	Menambahkan Data <i>Team Member</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Mendaftarkan tim member baru
Pre-conditions	Tim sudah ditambahkan di dalam aplikasi
Post-conditions	Tim member baru telah tersimpan ke dalam <i>database</i>
Main Success Scenario	<ol style="list-style-type: none"> 1. <include:login> 2. Membuka Menu <i>Team Member</i> 3. Memilih tim 4. Menginputkan informasi tim member 5. Melakukan submit 6. Aplikasi menyimpan data 7. Jika proses simpan data berhasil, akan tampil pesan bahwa data telah tersimpan 8. Aplikasi melakukan <i>refresh</i> halaman. 9. Manajemen dapat melihat informasi tim member yang baru ditambahkan
Extension / Alternate Scenario	1. Akan tampil pesan kesalahan jika proses simpan data gagal

Priority	<i>Medium</i>
Special Requirements	-
Open Questions	-

I. Melihat Data *Team Member*

Tabel 3.13 Spesifikasi *Use Case* Melihat Data *Team Member*

UC-6	Melihat Data <i>Team Member</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Melihat data <i>team member</i> yang sudah ditambahkan ke dalam aplikasi
Pre-conditions	-
Post-conditions	-
Main Success Scenario	<ol style="list-style-type: none"> 1. <include:login> 2. Pengguna membuka Menu <i>Team Member</i> 3. Aplikasi informasi <i>teams</i> yang ada di halaman <i>Team Member</i>
Extension / Alternate Scenario	-
Priority	<i>Medium</i>
Special Requirements	-
Open Questions	-

m. Mengubah Data *Team Member*

Tabel 3.14 Spesifikasi *Use Case* Mengubah Data *Team Member*

UC-6	Mengubah Data <i>Team Member</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Mengubah data <i>team member</i> yang sudah ditambahkan ke dalam aplikasi
Pre-conditions	Data <i>team member</i> sudah pernah ditambahkan ke dalam aplikasi
Post-conditions	Pengguna berhasil mengubah data <i>team member</i>
Main Success Scenario	<ol style="list-style-type: none"> 1. <include:login> 2. Pengguna membuka Menu <i>Task Member</i> 3. Aplikasi menampilkan informasi <i>task member</i> 4. Pengguna memilih baris data yang

ingin diubah dengan cara menekan tombol dengan ikon edit
 5. Aplikasi menampilkan form yang berisi data *Team Member*
 6. Pengguna melakukan perubahan data *team member* pada form
 7. Pengguna menekan tombol "Save" untuk menyimpan perubahan data
 8. Aplikasi menyimpan data *team member* yang sudah diubah
 9. Aplikasi menampilkan pesan sukses jika perubahan data yang dilakukan pengguna berhasil disimpan
 10. Aplikasi kembali mengarahkan ke halaman *Team Member*

Extension / Alternate Scenario

1. Aplikasi akan menampilkan pesan kesalahan jika perubahan data yang dilakukan pengguna gagal di simpan
 2. Pengguna batal melakukan perubahan data dengan menekan tombol "Cancel"

Priority *Medium*

Special Requirements -

Open Questions -

n. Menghapus Data *Team Member*

Tabel 3.15 Spesifikasi Use Case Menghapus Data *Team Member*

UC-6	Menghapus Data <i>Team Member</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Menghapus data <i>team member</i> yang sudah ditambahkan ke dalam aplikasi
Pre-conditions	Data <i>team member</i> sudah pernah ditambahkan ke dalam aplikasi
Post-conditions	Pengguna berhasil menghapus data <i>team member</i>
Main Success Scenario	<ol style="list-style-type: none"> 1. <include:login> 2. Pengguna membuka Menu <i>Team Member</i> 3. Aplikasi menampilkan informasi <i>team member</i> 4. Pengguna memilih baris data yang

ingin dihapus dengan cara menekan tombol dengan ikon hapus
 5. Aplikasi menampilkan modal konfirmasi untuk melanjutkan proses hapus data
 6. Pengguna menekan tombol "Yes" untuk melanjutkan hapus data
 8. Aplikasi menghapus data
 9. Aplikasi menampilkan pesan sukses jika proses hapus data yang dilakukan pengguna berhasil

Extension / Alternate Scenario

1. Aplikasi akan menampilkan pesan kesalahan jika proses hapus data yang dilakukan pengguna gagal
 2. Pengguna batal melakukan perubahan data dengan menekan tombol "Cancel"

Priority

Medium

Special Requirements

-

Open Questions

-

o. Menambahkan *Task*

Tabel 3.16 Spesifikasi *Use Case* Menambahkan Data *Task*

UC-5	Menambahkan Data <i>Task</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Menambahkan tugas baru di dalam proyek
Pre-conditions	Proyek sudah ditambahkan di dalam aplikasi
Post-conditions	Tugas baru telah tersimpan ke dalam <i>database</i>
Main Success Scenario	<ol style="list-style-type: none"> 1. <include:login> 2. Membuka Menu <i>Task</i> 3. Memilih proyek 4. Menginputkan informasi tugas 5. Melakukan submit 6. Aplikasi menyimpan data 7. Jika proses simpan data berhasil, akan tampil pesan bahwa data telah tersimpan 8. Aplikasi melakukan <i>refresh</i> halaman. 9. Manajemen dan <i>developer</i> dapat melihat informasi tugas yang baru ditambahkan

Extension / Alternate Scenario	1. Akan tampil pesan kesalahan jika proses simpan data gagal
Priority	<i>Medium</i>
Special Requirements	-
Open Questions	-

p. Melihat Data *Task*

Tabel 3.17 Spesifikasi *Use Case* Melihat Data *Task*

UC-5	Melihat Data <i>Task</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Melihat data <i>task</i> yang sudah ditambahkan ke dalam aplikasi
Pre-conditions	-
Post-conditions	-

Main Success Scenario

1. <include:login>
2. Pengguna membuka Menu *Task*
3. Aplikasi informasi *task* yang ada di halaman *Task*

Extension / Alternate Scenario	-
Priority	<i>Medium</i>
Special Requirements	-
Open Questions	-

q. Mengubah Data *Task*

Tabel 3.18 Spesifikasi *Use Case* Mengubah Data *Task*

UC-5	Mengubah Data <i>Task</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Mengubah data <i>task</i> yang sudah ditambahkan ke dalam aplikasi
Pre-conditions	Data <i>task</i> sudah pernah ditambahkan ke dalam aplikasi
Post-conditions	Pengguna berhasil mengubah data <i>task</i>
Main Success Scenario	1. <include:login>

2. Pengguna membuka Menu *Task*
3. Aplikasi menampilkan informasi *task*
4. Pengguna memilih baris data yang ingin diubah dengan cara menekan tombol dengan ikon edit
5. Aplikasi menampilkan form yang berisi data *Task*
6. Pengguna melakukan perubahan data *team member* pada form
7. Pengguna menekan tombol "Save" untuk menyimpan perubahan data
8. Aplikasi menyimpan data *task* yang sudah diubah
9. Aplikasi menampilkan pesan sukses jika perubahan data yang dilakukan pengguna berhasil disimpan
10. Aplikasi kembali mengarahkan ke halaman *Task*

● **Extension / Alternate Scenario**

1. Aplikasi akan menampilkan pesan kesalahan jika perubahan data yang dilakukan pengguna gagal di simpan
2. Pengguna batal melakukan perubahan data dengan menekan tombol "Cancel"

Priority

Medium

Special Requirements

-

Open Questions

-

r. Menghapus Data *Task*

Tabel 3.19 Spesifikasi *Use Case* Menghapus Data *Task*

UC-5	Menghapus Data <i>Task</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Menghapus data <i>task</i> yang sudah ditambahkan ke dalam aplikasi
Pre-conditions	Data <i>task</i> sudah pernah ditambahkan ke dalam aplikasi
Post-conditions	Pengguna berhasil menghapus data <i>task</i>
Main Success Scenario	<ol style="list-style-type: none"> 1. <include:login> 2. Pengguna membuka Menu <i>Task</i> 3. Aplikasi menampilkan informasi <i>task</i> 4. Pengguna memilih baris data yang ingin hapus dengan cara menekan

	tombol dengan ikon hapus 5. Aplikasi menampilkan modal konfirmasi untuk melanjutkan proses hapus data 6. Pengguna menekan tombol "Yes" untuk melanjutkan hapus data 8. Aplikasi menghapus data 9. Aplikasi menampilkan pesan sukses jika proses hapus data yang dilakukan pengguna berhasil
Extension / Alternate Scenario	1. Aplikasi akan menampilkan pesan kesalahan jika proses hapus data yang dilakukan pengguna gagal 2. Pengguna batal melakukan perubahan data dengan menekan tombol "Cancel"
Priority	Medium
Special Requirements	-
Open Questions	-

s. Menambahkan *Task Member*

Tabel 3.20 Spesifikasi *Use Case* Menambahkan Data *Task Member*

UC-8	Menambahkan Data <i>Task Member</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Menambahkan member baru di dalam sebuah tugas/ <i>task</i>
Pre-conditions	<i>Task</i> sudah pernah ditambahkan di dalam aplikasi
Post-conditions	<i>Task member</i> baru telah tersimpan ke dalam <i>database</i>
Main Success Scenario	<ol style="list-style-type: none"> 1. <include:login> 2. Membuka Menu <i>Task Member</i> 3. Memilih <i>task</i> 4. Memilih member 5. Melakukan submit 6. Aplikasi menyimpan data 7. Jika proses simpan data berhasil, akan tampil pesan bahwa data telah tersimpan 8. Aplikasi melakukan <i>refresh</i> halaman. 9. Manajemen dapat melihat informasi <i>task member</i> yang baru ditambahkan
Extension / Alternate Scenario	1. Akan tampil pesan kesalahan jika proses simpan data gagal

Priority	<i>Medium</i>
Special Requirements	-
Open Questions	-

t. Melihat Data *Task Member*

Tabel 3.21 Spesifikasi *Use Case* Melihat Data *Task Member*

UC-7	Melihat Data <i>Task Member</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Melihat informasi <i>task member</i> yang sudah ditambahkan ke dalam aplikasi
Pre-conditions	-
Post-conditions	-
Main Success Scenario	<ol style="list-style-type: none"> 1. <include:login> 2. Pengguna membuka Menu <i>Task Member</i> 3. Aplikasi menampilkan informasi <i>task member</i>
Extension / Alternate Scenario	-
Priority	<i>Medium</i>
Special Requirements	-
Open Questions	-

u. Mengubah Data *Task Member*

Tabel 3.22 Spesifikasi *Use Case* Mengubah Data *Task Member*

UC-7	Mengubah Data <i>Task Member</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Mengubah data <i>task member</i> yang sudah ditambahkan ke dalam aplikasi
Pre-conditions	Data <i>task member</i> sudah pernah ditambahkan ke dalam aplikasi
Post-conditions	Pengguna berhasil mengubah data <i>task member</i>
Main Success Scenario	<ol style="list-style-type: none"> 1. <include:login> 2. Pengguna membuka Menu <i>Task Member</i> 3. Aplikasi menampilkan informasi <i>task member</i>

4. Pengguna memilih baris data yang ingin diubah dengan cara menekan tombol dengan ikon edit
5. Aplikasi menampilkan form yang berisi data *Task Member*
6. Pengguna melakukan perubahan data *team member* pada form
7. Pengguna menekan tombol "Save" untuk menyimpan perubahan data
8. Aplikasi menyimpan data *task member* yang sudah diubah
9. Aplikasi menampilkan pesan sukses jika perubahan data yang dilakukan pengguna berhasil disimpan
10. Aplikasi kembali mengarahkan ke halaman *Task Member*

Extension / Alternate Scenario

1. Aplikasi akan menampilkan pesan kesalahan jika perubahan data yang dilakukan pengguna gagal di simpan
2. Pengguna batal melakukan perubahan data dengan menekan tombol "Cancel"

Priority

Medium

Special Requirements

-

Open Questions

-

v. Menghapus Data *Task Member*

Tabel 3.23 Spesifikasi Use Case Menghapus Data *Task Member*

UC-7	Menghapus Data <i>Task Member</i>
Actor	Pengguna
Stakeholders and Interest	Direksi, Project Manager, Human Resource
Trigger	Menghapus data <i>task member</i> yang sudah ditambahkan ke dalam aplikasi
Pre-conditions	Data <i>task member</i> sudah pernah ditambahkan ke dalam aplikasi
Post-conditions	Pengguna berhasil menghapus data <i>task member</i>
Main Success Scenario	<ol style="list-style-type: none"> 1. <include:login> 2. Pengguna membuka Menu <i>Task Member</i> 3. Aplikasi menampilkan informasi <i>task member</i> 4. Pengguna memilih baris data yang

ingin hapus dengan cara menekan tombol dengan ikon hapus
 5. Aplikasi menampilkan modal konfirmasi untuk melanjutkan proses hapus data
 6. Pengguna menekan tombol "Yes" untuk melanjutkan hapus data
 8. Aplikasi menghapus data
 9. Aplikasi menampilkan pesan sukses jika proses hapus data yang dilakukan pengguna berhasil

Extension / Alternate Scenario

1. Aplikasi akan menampilkan pesan kesalahan jika proses hapus data yang dilakukan pengguna gagal
 2. Pengguna batal melakukan perubahan data dengan menekan tombol "Cancel"

Priority

Medium

Special Requirements

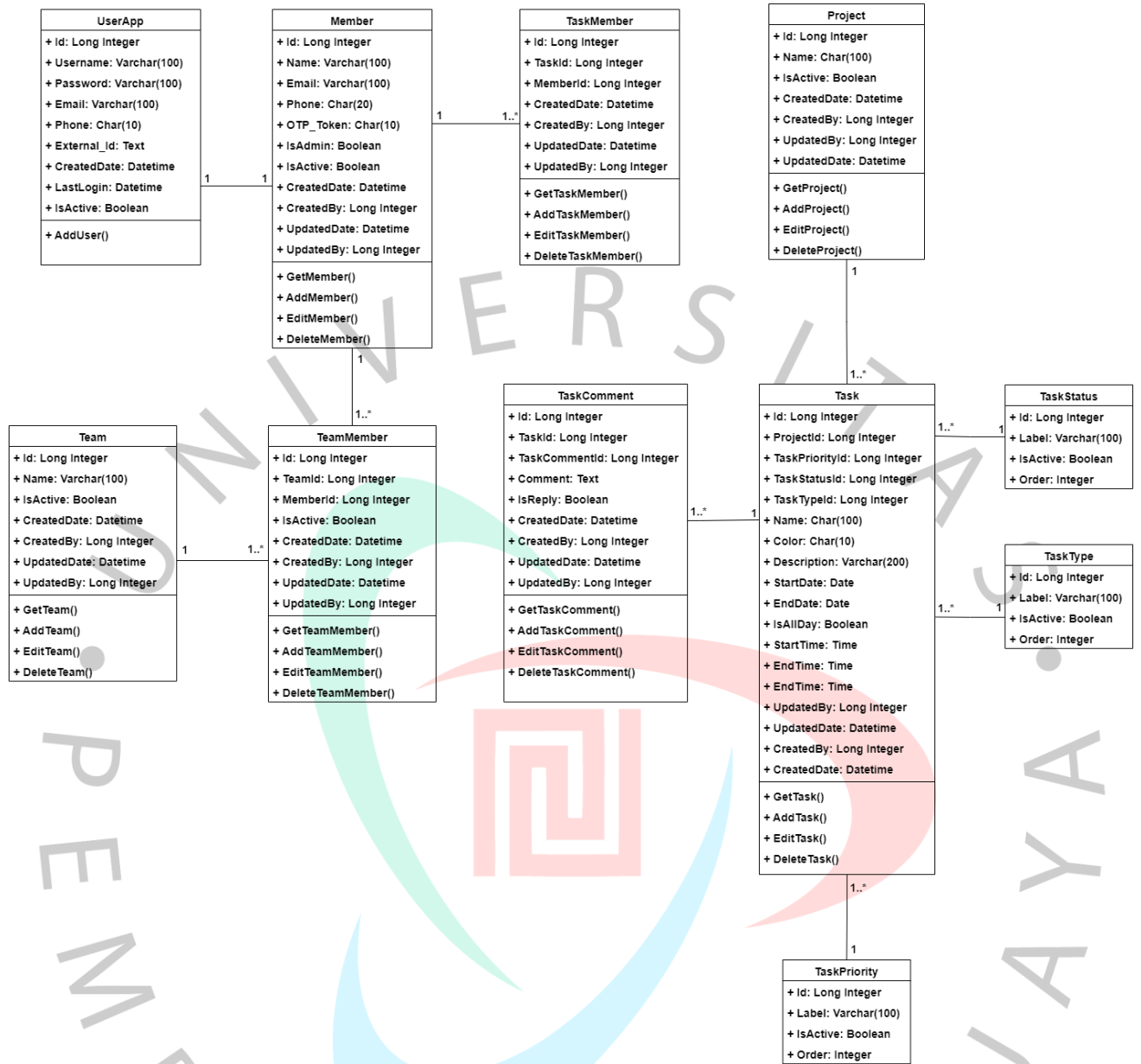
-

Open Questions

-

c. *Class Diagram*



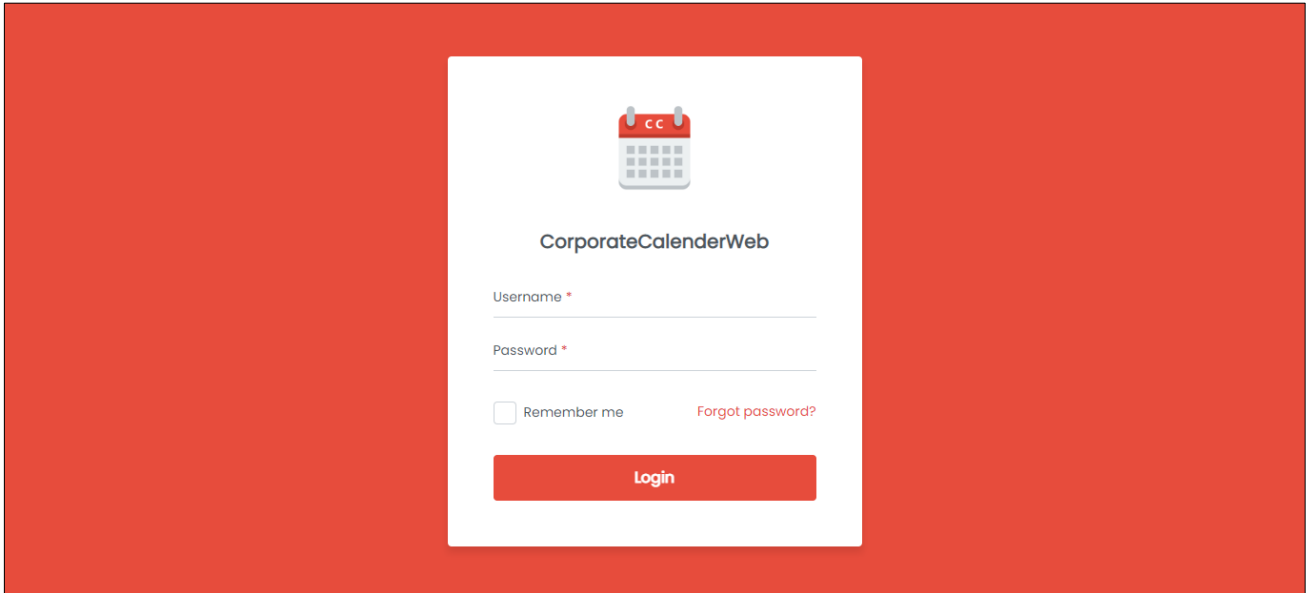


Gambar 3.7 Class Diagram Aplikasi Corporate Calendar
 Sumber : Hasil dokumentasi Praktikan

3.2.4. Tampilan Antarmuka Aplikasi (*User Interface*)

Berikut ini beberapa tampilan dari setiap halaman yang ada di dalam aplikasi Corporate Calender :

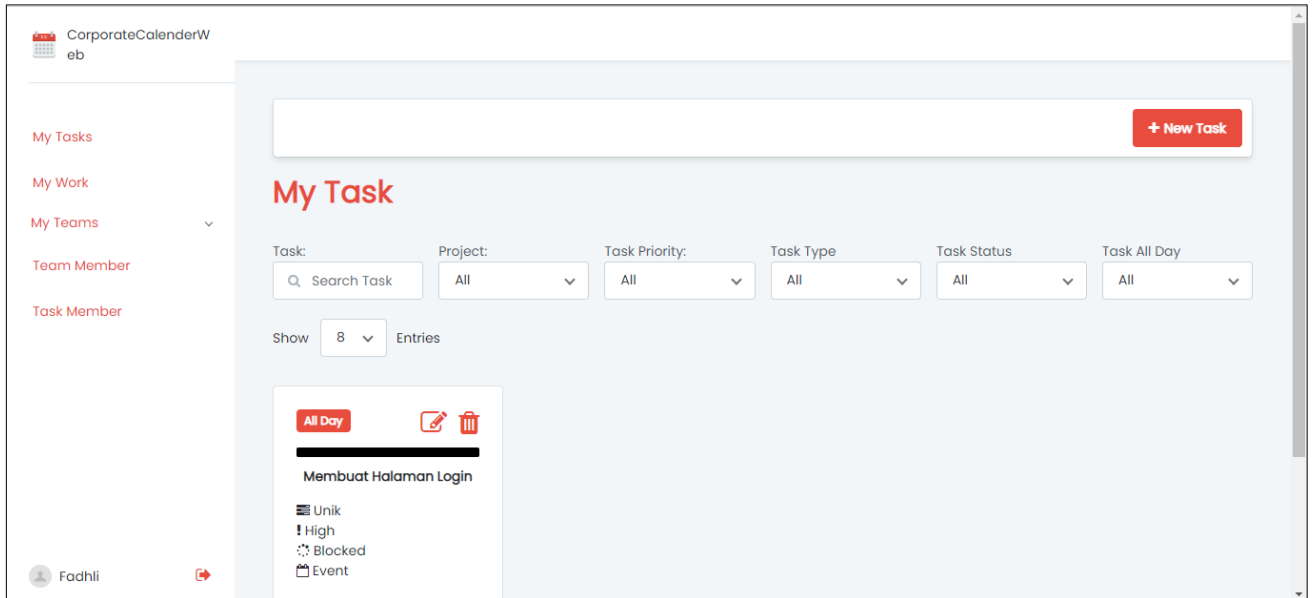
a) Halaman Login



Gambar 3.8 Halaman Login
Sumber : Hasil dokumentasi Praktikan

Halaman login merupakan halaman pertama yang dilihat oleh pengguna ketika pengguna mengakses aplikasi Corporate Calendar namun belum melakukan login ke dalam aplikasi.

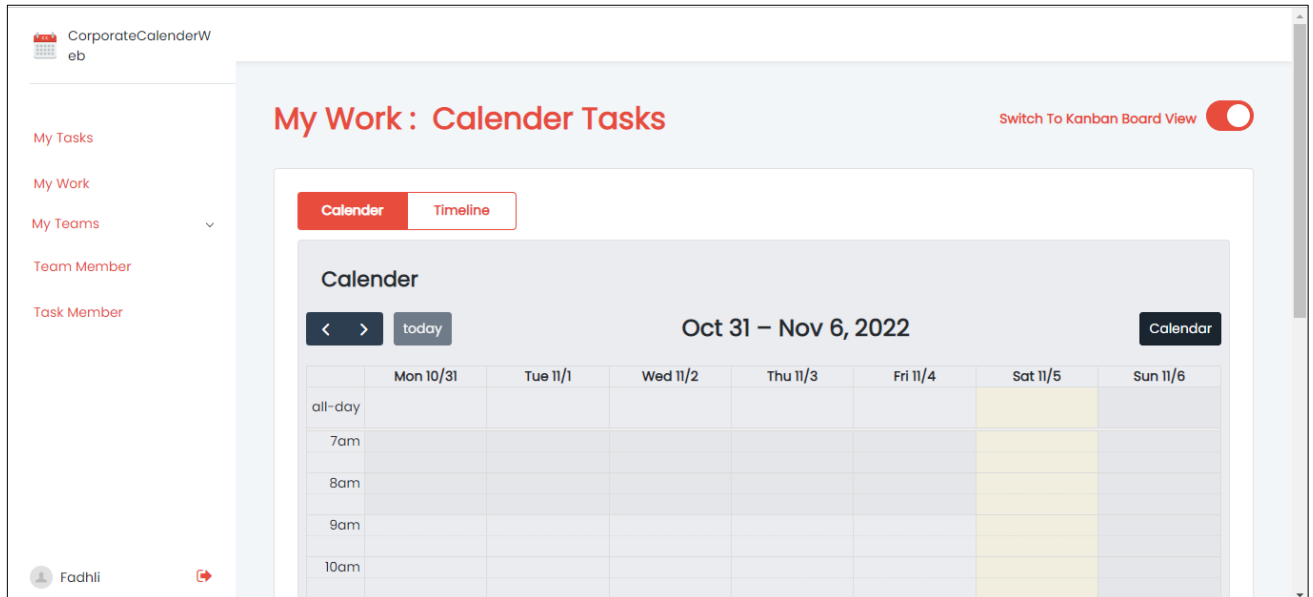
b) Halaman *My Task*



Gambar 3.9 Halaman *My Task*
Sumber : Hasil dokumentasi Praktikan

Halaman *My Task* berisi informasi mengenai *task* apa saja yang sudah dibuat. Selain itu, di halaman ini pengguna bisa mencari *task*, menambahkan *task* baru serta mengubah dan menghapus *task* yang sudah ada.

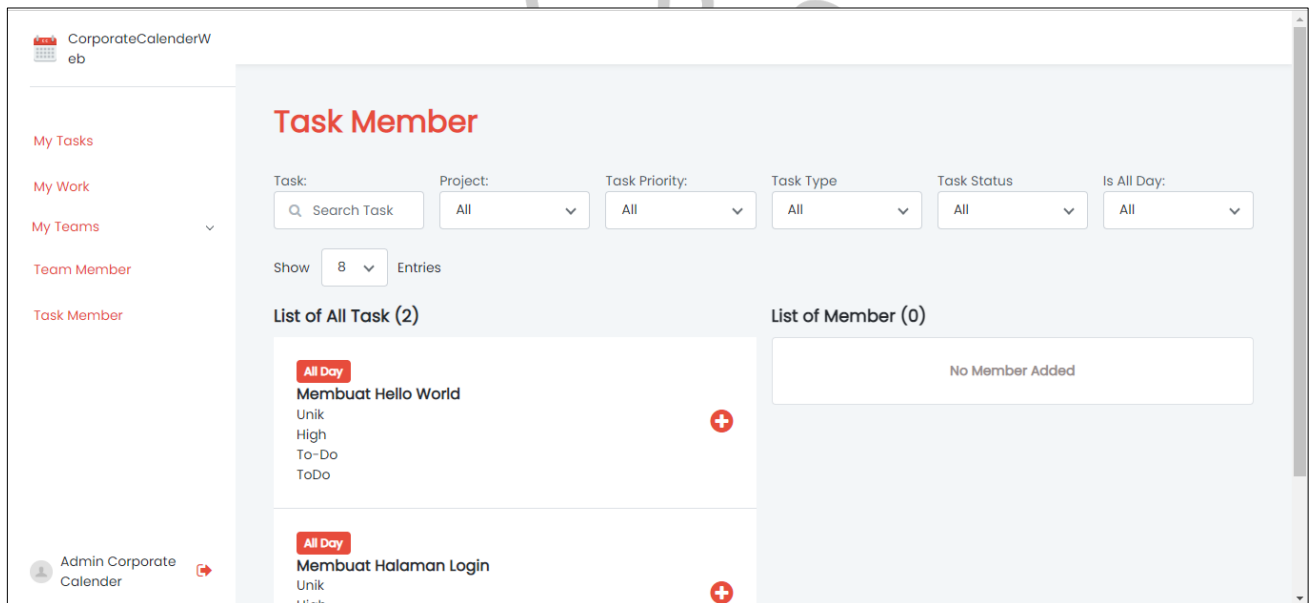
c) Halaman *My Work*



Gambar 3.10 Halaman *My Work*
Sumber : Hasil dokumentasi Praktikan

Halaman ini berisi informasi *task* namun menggunakan *custom widget* agar dapat menampilkan informasi *task* dalam tampilan *calendar* dan juga *timeline*. Tidak hanya itu, di halaman ini pengguna juga bisa merubah tampilan ke *kanban board view* untuk menampilkan informasi *task* dalam bentuk papan kanban.

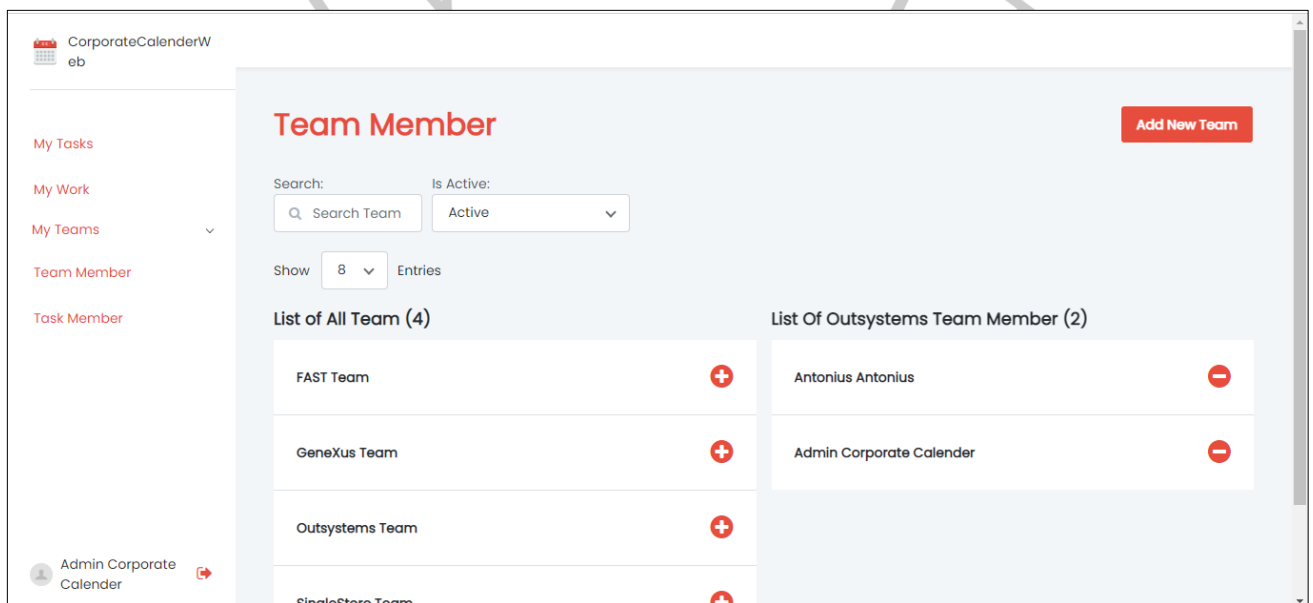
d) Halaman Task Member



Gambar 3.11 Halaman *Task Member*
Sumber : Hasil dokumentasi Praktikan

Halaman ini berisi informasi *task* dan member yang tergabung di dalam sebuah *task*. Pada halaman ini, pengguna bisa mencari *task* tertentu dan melihat member yang tergabung di dalam *task* tersebut. Jika pengguna login sebagai Admin, pengguna bisa menambahkan member ke dalam sebuah *task* dan menghapus member yang tergabung ke dalam sebuah *task*. Namun jika pengguna login sebagai member, pengguna hanya bisa melihat informasi yang ada pada halaman *Task Member*.

e) Halaman Team Member

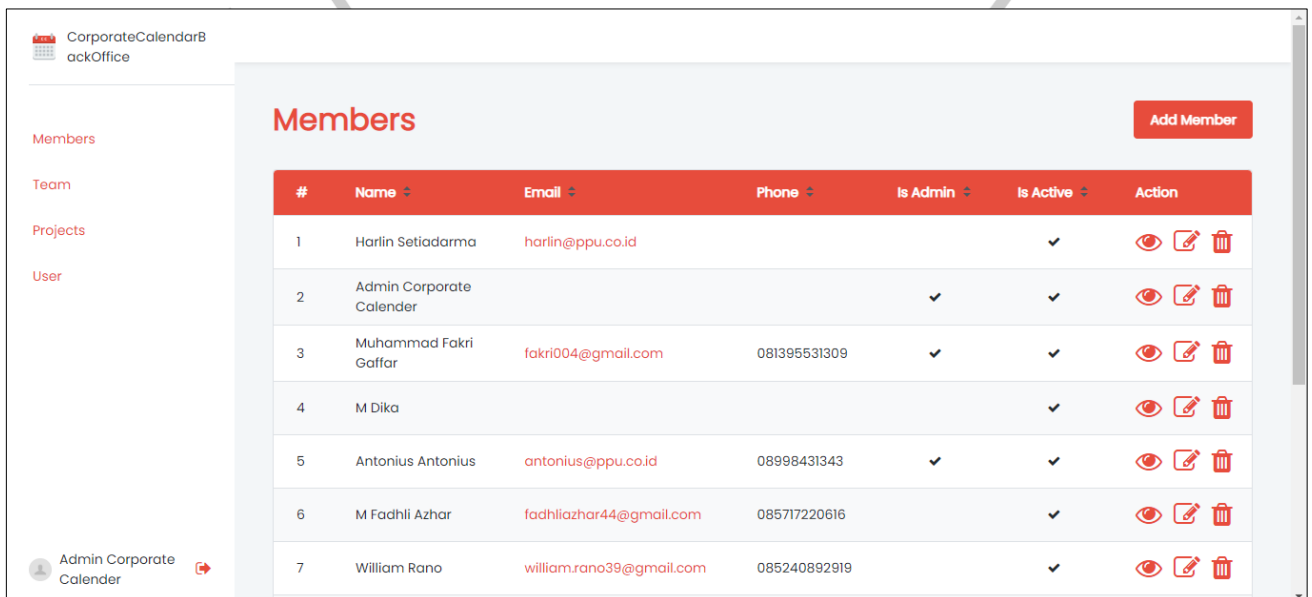


Gambar 3.12 Halaman *Team Member*

Sumber : Hasil dokumentasi Praktikan

Halaman ini berisi informasi mengenai *team* dan member yang tergabung dalam sebuah *team*. Pada halaman ini, pengguna bisa mencari *team* tertentu dan melihat member yang tergabung di dalam *team* tersebut. Jika pengguna login sebagai Admin, pengguna bisa menambahkan member ke dalam sebuah *team* dan menghapus member tertentu di dalam sebuah *team*. Namun jika pengguna login sebagai member, pengguna hanya bisa melihat informasi yang ada pada halaman *Team Member*.

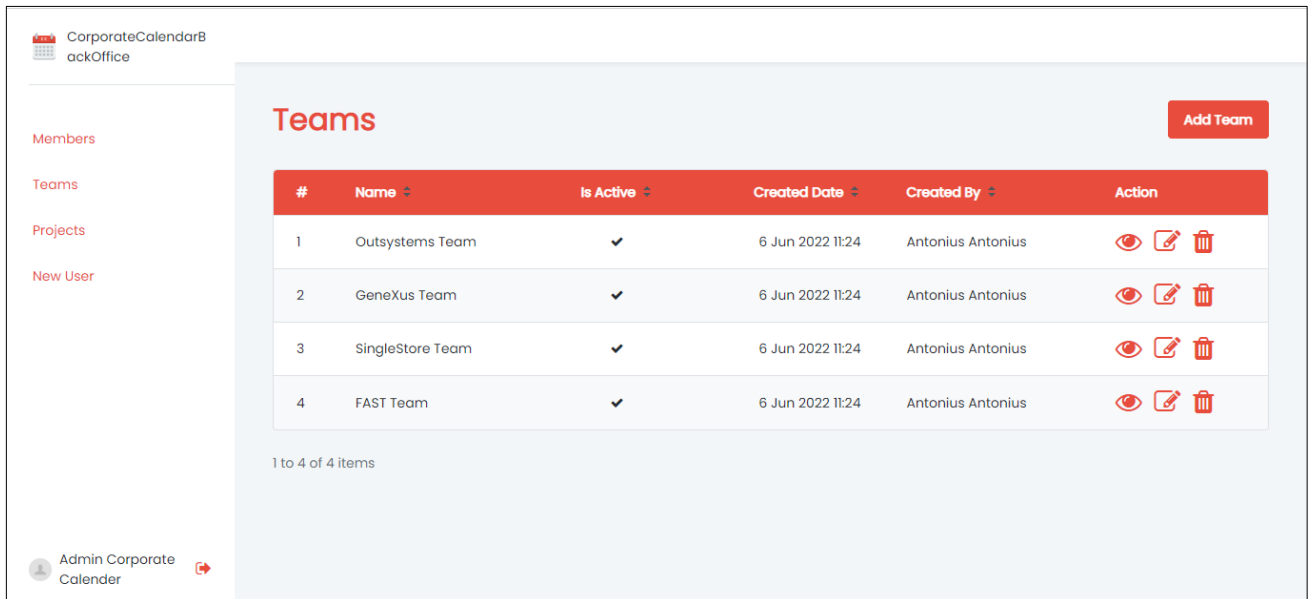
f) Halaman *Members*



Gambar 3.13 Halaman *Members*
 Sumber : Hasil dokumentasi Praktikan

Halaman ini diakses melalui modul CorporateCalendarBackOffice. Pengguna yang dapat mengakses modul ini hanya pengguna dengan *Admin Role*. Pada halaman ini, terdapat informasi member yang sudah ditambahkan ke dalam aplikasi. Tidak hanya itu, di halaman ini admin juga bisa menambahkan member baru serta merubah dan menghapus member yang sudah ada. Namun sebelum menambahkan member baru, admin harus menambahkan user terlebih dahulu di dalam aplikasi Corporate Calendar.

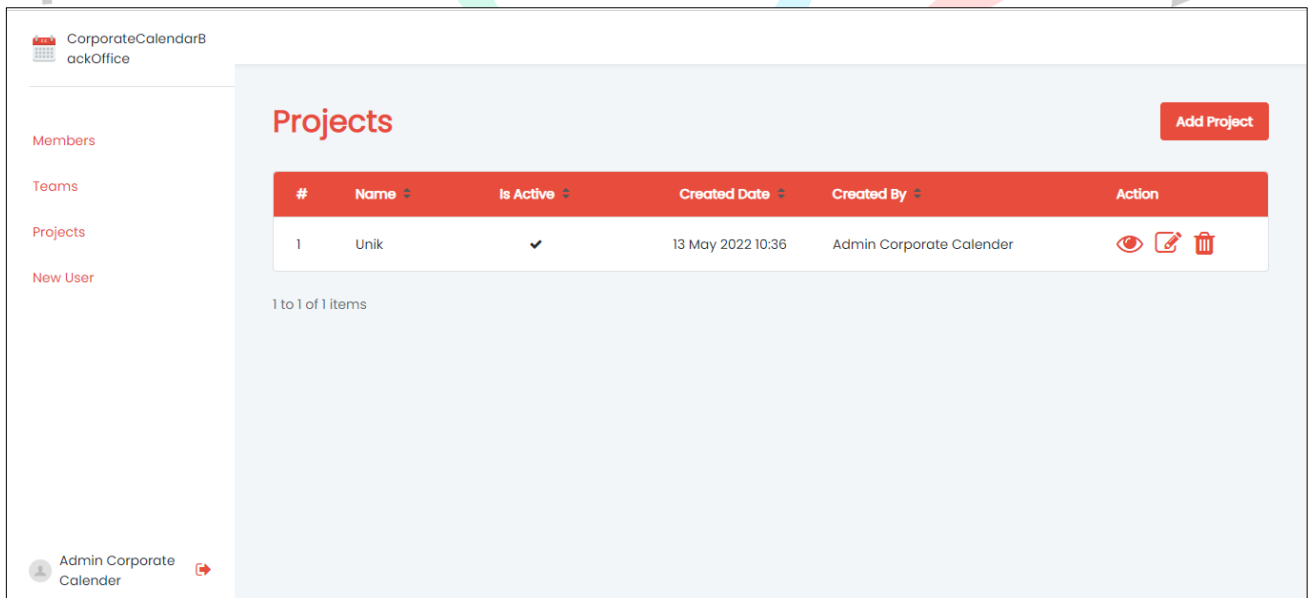
g) Halaman *Teams*



Gambar 3.14 Halaman Teams
Sumber : Hasil dokumentasi Praktikan

Halaman ini berisi informasi mengenai *team* yang sudah ditambahkan di dalam aplikasi Corporate Calendar. Pada halaman ini, admin juga bisa menambahkan *team* baru serta merubah dan menghapus *team* yang sudah ada. Sama seperti halaman *Members*, halaman ini hanya bisa diakses oleh admin.

h) Halaman Projects

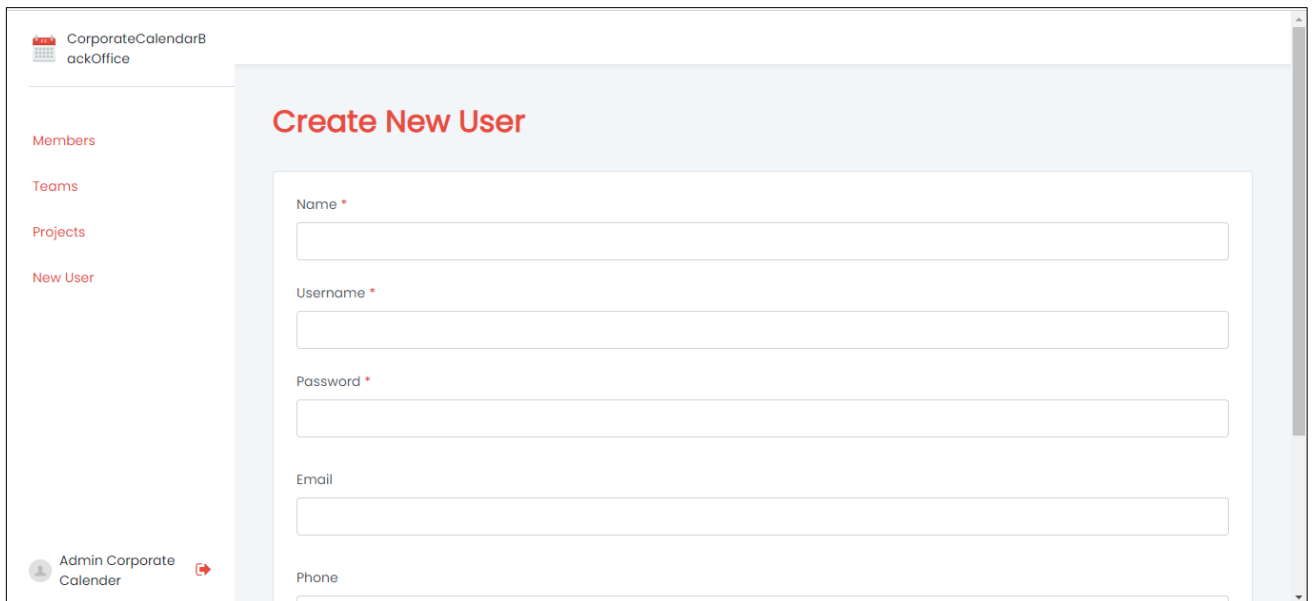


Gambar 3.15 Halaman Projects
Sumber : Hasil dokumentasi Praktikan

Halaman ini berisi informasi mengenai *project* yang sudah ditambahkan di dalam aplikasi. Pengguna yang bisa mengakses halaman ini hanya pengguna yang memiliki *Admin Role*. Selain melihat informasi

project, admin juga bisa menambahkan *project* baru serta merubah dan menghapus *project* yang sudah ada.

i) Halaman *New User*



Gambar 3.16 Halaman *New User*

Sumber : Hasil dokumentasi Praktikan

Pada halaman ini, terdapat form yang dapat digunakan admin untuk mendaftarkan pengguna baru. Halaman ini hanya bisa diakses pengguna yang memiliki *Admin Role*.

3.3 Kendala Yang Dihadapi

Selama Praktikan menjalani KP di PT. Perkasa Pilar Utama, ada beberapa kesulitan yang dihadapi ketika melakukan pengembangan aplikasi, antara lain yaitu :

1. Praktikan kesulitan mendesain tampilan dari aplikasi. Mulai dari penentuan tema dari aplikasi hingga tampilan aplikasi per halaman. Ini dikarenakan tidak adanya *mock-up* atau dokumen yang menjelaskan mengenai *UI* dari aplikasi.
2. Sulitnya mengimplementasikan beberapa fitur dan komponen yang ada di dalam aplikasi. Beberapa fitur dan komponen tersebut seperti komponen *drag and drop* dan *timeline*. Hal itu dikarenakan beberapa fitur dan komponen tersebut belum pernah digunakan oleh Praktikan sendiri. Selain itu fitur dan komponen yang disebutkan juga ingin dikustomisasi agar sesuai dengan kebutuhan aplikasi.

3.4 Cara Mengatasi Kendala

Berikut ini beberapa cara yang dilakukan Praktikan ketika menghadapi kendala sewaktu mengembangkan aplikasi, antara lain :

1. Melakukan eksplorasi tampilan untuk aplikasi. Eksplorasi yang dilakukan yaitu dengan cara mencari referensi- referensi yang dianggap sesuai dengan profil PT. Perkasa Pilar Utama dan serasi dengan fitur dan komponen yang ada di dalam aplikasi. Tentunya hasil eksplorasi akan didiskusikan terlebih dahulu dengan tim sebelum dilakukan implementasi.
2. Melakukan eksplorasi terhadap fitur dan komponen yang dianggap sulit dan belum pernah digunakan oleh Praktikan. Eksplorasi tersebut dilakukan Praktikan dengan mencari dan mempelajari dokumentasinya. Hal itu membantu Praktikan untuk melakukan konfigurasi dan mengkustomisasi komponen-komponen tersebut sesuai dengan kebutuhan.

3.5 Pembelajaran Yang Diperoleh dari Kerja Profesi

Kerja profesi sendiri memiliki maksud dan tujuan yang baik bagi Praktikan untuk persiapan di dunia kerja. Banyak manfaat dan pengalaman yang bisa menjadi bekal bagi Praktikan untuk persiapan di dunia kerja yang sebenarnya. Selain itu, menurut pandangan Praktikan pribadi, KP merupakan wadah bagi Praktikan untuk belajar bekerja sama dengan rekan tim yang ada. Karena kemampuan bekerja sama seseorang sangat berpengaruh terhadap kecepatan penyelesaian dan hasil akhir dari sebuah proyek yang sedang di kerjakan.

Selama Praktikan melakukan KP di PT. Perkasa Pilar Utama, Praktikan banyak mendapatkan pembelajaran dari *soft skills* maupun *hard skills*. Secara *soft skills*, Praktikan dituntut pihak manajemen untuk mampu berkomunikasi dengan baik sesama rekan tim. Selain itu, Praktikan juga diajarkan cara melakukan manajemen waktu sebaik mungkin agar tugas yang diberikan kepada Praktikan dapat selesai tepat waktu. Secara *hard*

skills, Praktikan banyak belajar hal-hal teknis terkait pengembangan perangkat lunak. Banyak hal-hal teknis yang Praktikan pelajari dari para pengembang senior di PT. Perkasa Pilar Utama. Praktikan diajarkan untuk selalu menerapkan *best practices* ketika mengembangkan sebuah aplikasi. Mulai dari desain arsitektur aplikasi yang harus rapi dan sesuai, membuat validasi yang baik dan benar di sisi *client* dan juga *server*, membuat struktur tabel yang kompleks, dan lain-lain. Selain itu, Praktikan yang bertugas sebagai pengembang dituntut untuk selalu teliti terhadap pembuatan sebuah *flow*, *logic* dan juga *function*. Hal itu selalu ditekankan pihak manajemen agar pengembang mampu menghasilkan sebuah aplikasi dengan kualitas yang baik.

Secara garis besar, pelaksanaan KP yang dilakukan oleh Praktikan sehari-harinya selalu berkaitan dengan *software development*. Hal-hal yang menyangkut dengan *software development* mulai dari analisa dan pengumpulan kebutuhan pengguna, melakukan implementasi fitur, melakukan *deployment* aplikasi ke *production server* hingga melakukan *maintenance* sudah dirasakan Praktikan selama melakukan KP di PT. Perkasa Pilar Utama.