

BAB III

PELAKSANAAN MBKM

3.1 Bidang Kerja

Dalam melaksanakan program magang MBKM, praktikan ditempatkan di PT Siaga Abdi Utama, dalam divisi *IT Development/Operation* dengan peran sebagai pengembang aplikasi *mobile*. Proyek yang dikembangkan oleh praktikan dalam program magang ini adalah pengembangan fitur pengajuan cuti/izin dan lembur untuk aplikasi MySAU untuk *platform* iOS dan Android.

MySAU merupakan sebuah aplikasi HRIS (Human Resource Information System), yang merupakan sebuah sistem komputer yang terintegrasi dirancang untuk menjalankan berbagai fungsi terkait dengan aliran informasi di dalam sebuah organisasi terkait dengan sumber daya manusia (Roach, 2017). Aplikasi ini sudah hadir dalam bentuk situs web dan aplikasi *mobile*. Fitur-fitur yang tersedia dalam aplikasi MySAU Mobile mencakup pencatatan presensi, rekam aktivitas, dan detail pekerja. Praktikan ditugaskan untuk mengembangkan fitur pengajuan cuti/izin dan lembur, yang sebelumnya hanya dapat diakses dalam versi situs web aplikasi.

Aplikasi MySAU dikembangkan dengan *framework* React Native dan bahasa pemrograman TypeScript, yang sudah dipelajari oleh praktikan. Dengan demikian, proyek tersebut relevan dengan bidang studi yang diampu praktikan sebagai mahasiswa Informatika. Praktikan juga mendapatkan pengetahuan baru tentang bagaimana cara penggunaan TypeScript dalam mengembangkan aplikasi *mobile*, mengembangkan perangkat lunak dengan metode Agile, dan menggunakan *framework* React Native untuk membuat tampilan antarmuka.

3.2 Pelaksanaan Kerja

Tahap-tahap yang dilakukan oleh praktikan dalam pelaksanaan kerja terdiri dari perancangan alur program, perancangan antarmuka, implementasi *front-end* menggunakan React Native, *consuming* API yang sebelumnya sudah dibuat untuk aplikasi MySAU berbasis web, dan perbaikan. Pada tahap perbaikan, praktikan

memperbaiki kesalahan atau *bug* yang ditemukan dalam pengujian. Setelah tahap perbaikan selesai, praktikan mengunggah aplikasi MySAU yang terbaru ke Apple App Store dan Google Play Store.

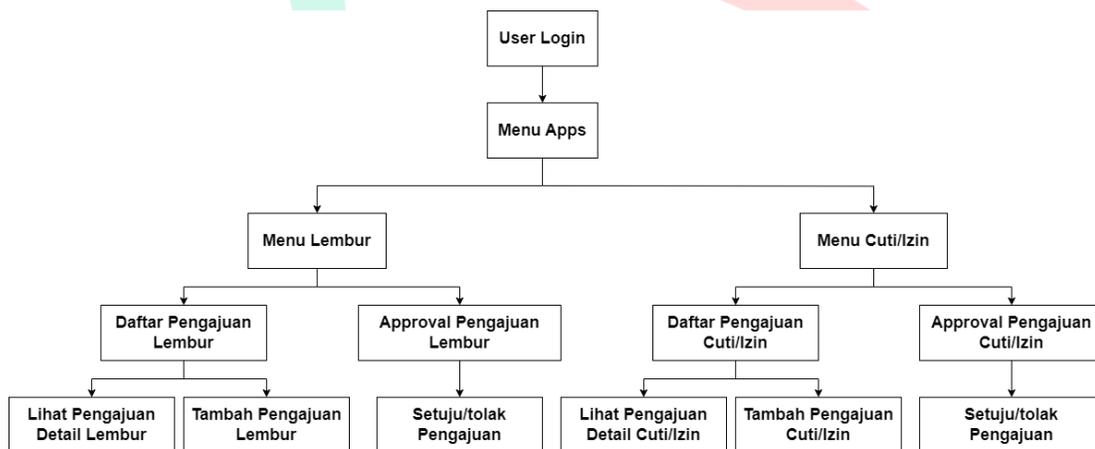
| Kegiatan | Juli | | Agustus | | | | September | | | |
|---|------|----|---------|----|----|----|-----------|----|----|----|
| | M2 | M3 | M1 | M2 | M3 | M4 | M1 | M2 | M3 | M4 |
| Perancangan Alur Program | ■ | ■ | | | | | | | | |
| Perancangan Antarmuka | | | | | | | | | | |
| Implementasi <i>Front-End</i> | | ■ | ■ | ■ | ■ | | | | | |
| Penggunaan atau <i>consuming</i> API Cuti/Izin dan Lembur | | | | | ■ | ■ | ■ | ■ | | |
| Pengujian | | | | | | | | ■ | | |
| Memperbaiki error atau bug yang ditemukan dalam pengujian | | | | | | | | | ■ | ■ |

Tabel 3,1 Linimasa Kegiatan

Berdasarkan Tabel 3.1, praktikan mengembangkan fitur cuti/izin dan lembur untuk aplikasi MySAU Mobile selama 3 bulan. Fitur tersebut dalam masa yang akan datang akan terus dikembangkan dari sisi tampilan antarmuka atau penambahan fitur pendukung. Dalam melaksanakan proyek ini, pembimbing eksternal bertanggung jawab memberikan arahan dan melakukan pemantauan terhadap pekerjaan praktikan. Praktikan juga memberikan informasi kemajuan proyek kepada pembimbing eksternal setiap minggu untuk mendapatkan saran, dengan tujuan mendapatkan hasil yang maksimal.

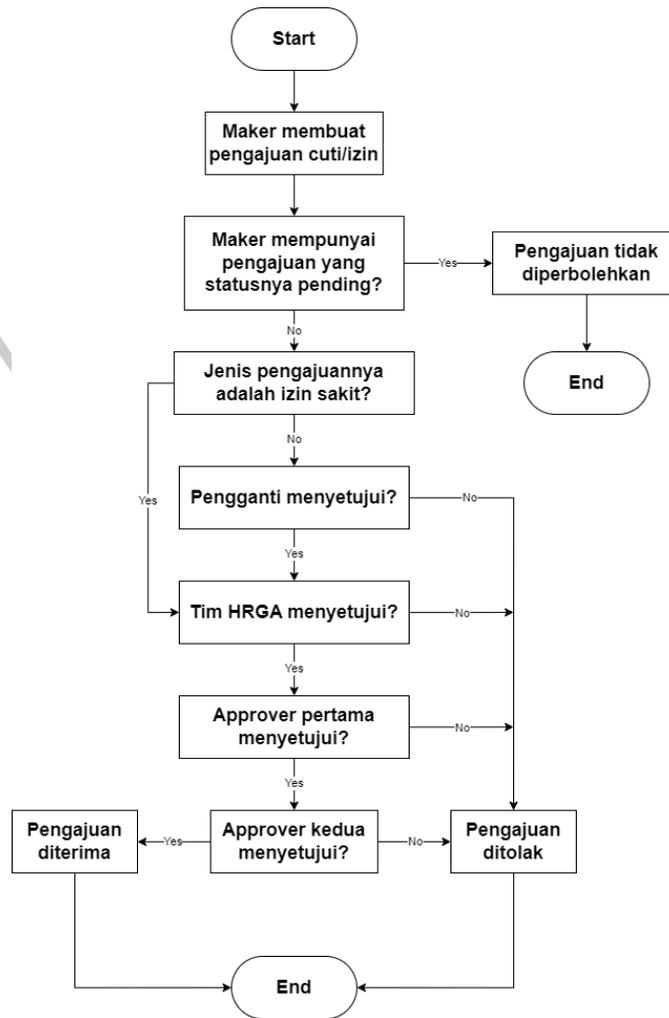
3.2.1 Perancangan

Perancangan perangkat lunak adalah proses dalam menciptakan dan memelihara perangkat lunak, yang meliputi pengembangan dan penyesuaian perangkat lunak dengan memperhitungkan aspek waktu. Perancangan sistem perangkat lunak dibuat dalam bentuk diagram alir (*flowchart*). Pada tahap perancangan, pembimbing eksternal memberikan arahan kepada praktikan untuk mengembangkan fitur pada aplikasi MySAU Mobile. Aplikasi MySAU berfungsi sebagai sistem informasi sumber daya manusia (HRIS) pekerja SAU. Fitur yang dikembangkan adalah pengajuan cuti/izin, persetujuan pengajuan cuti/izin, pengajuan lembur, dan persetujuan pengajuan lembur. Semua fitur tersebut sebelumnya sudah dikembangkan dan tersedia dalam aplikasi MySAU versi aplikasi website. Praktikan membuat alur fitur yang sudah dijelaskan oleh pembimbing eksternal yang disajikan dalam Gambar 3.1.



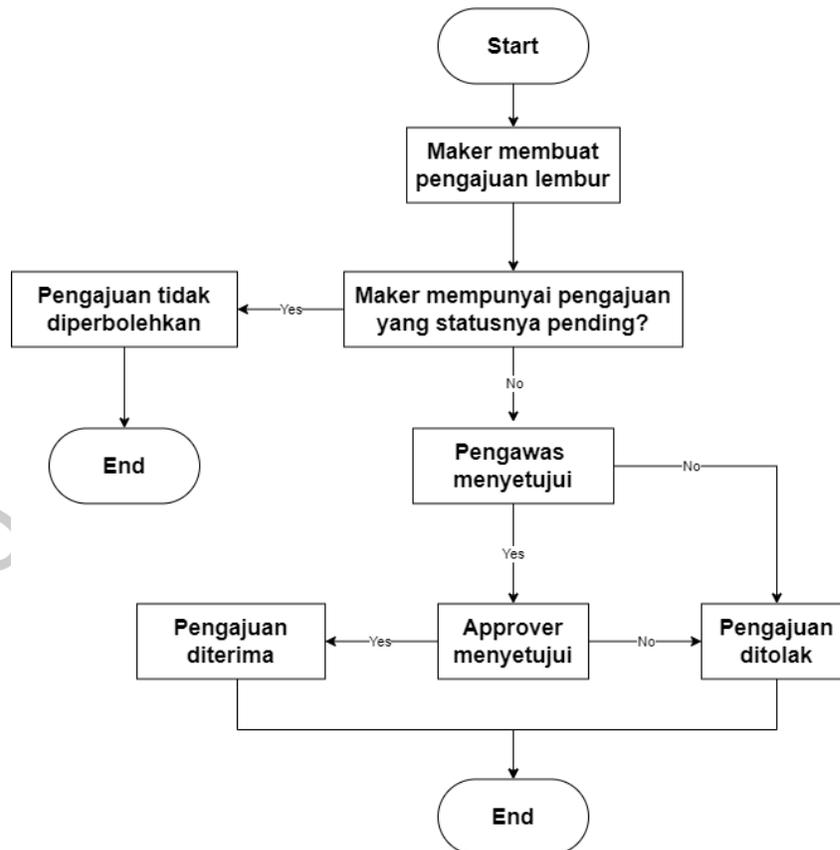
Gambar 3.1 Alur Fitur Cuti/Izin dan Lembur

Gambar 3.1 menjelaskan alur dari fitur aplikasi yang dibuat oleh praktikan. Pembimbing eksternal juga menjelaskan bagaimana alur pengajuan dan proses persetujuan cuti/izin dan lembur yang dijelaskan dalam Gambar 3.2. Kedua fitur tersebut memiliki alur yang serupa.



Gambar 3.2 Alur Pengajuan Cuti/Izin

Gambar 3.2 menggambarkan alur pengajuan dan persetujuan pengajuan permohonan cuti/izin. Permohonan cuti/izin dapat dilakukan jika tidak ada pengajuan yang masih berstatus pending. Dalam proses pengajuan ini, terdapat 3 atau 4 pihak yang terlibat dalam proses persetujuan, yaitu pengganti (jika jenis izin bukan izin sakit), Tim HRGA, pihak manajemen sebagai approver 1 dan 2. Apabila salah satu dari pihak-pihak menyetujui pengajuan, pengajuan akan diteruskan ke pihak berikutnya hingga mencapai approver 2 sebagai pihak terakhir. Sebaliknya, jika salah satu pihak menolak pengajuan, permohonan cuti/izin akan ditolak dan tidak akan diteruskan ke pihak berikutnya.

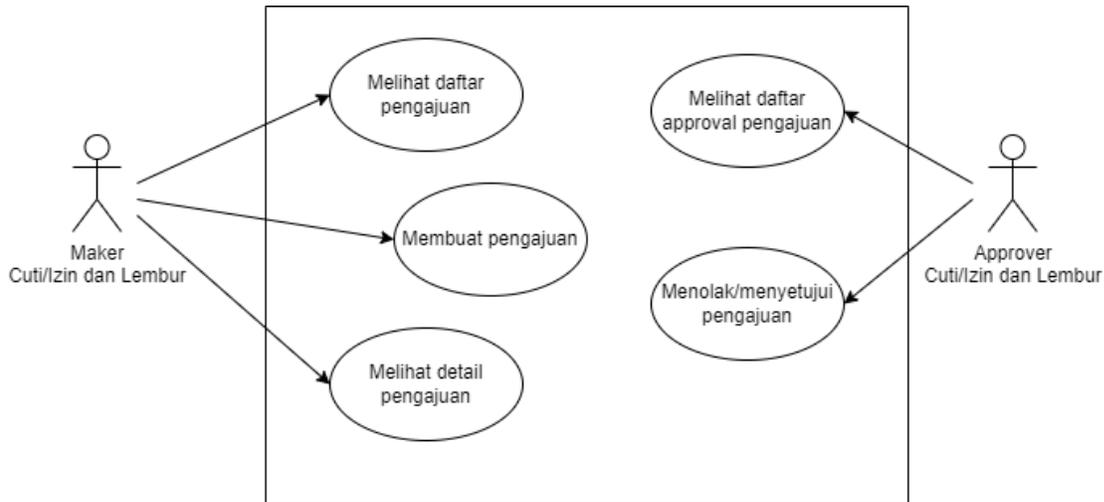


Gambar 3.3 Alur Pengajuan Lembur

Gambar 3.3 menggambarkan alur pengajuan dan persetujuan permohonan lembur. Seperti alur pengajuan cuti/izin, permohonan pengajuan lembur hanya dapat dilakukan jika pemohon tidak memiliki pengajuan yang statusnya masih berstatus pending. Berbeda dengan pengajuan cuti/izin, dalam pengajuan lembur hanya terdapat 2 pihak yang terlibat dalam proses persetujuan, yakni pengawas lembur dan approver yang berasal dari pihak manajemen. Serupa dengan alur pengajuan cuti/izin, jika terdapat salah satu pihak yang menolak pengajuan, permohonan akan ditolak, sementara jika salah satu pihak menyetujui permohonan maka akan diteruskan ke pihak berikutnya.

Kemudian, sebagai tahap selanjutnya praktikan membuat diagram *Unified Modelling Language* (UML). UML merupakan pendekatan yang umum digunakan untuk menggambarkan dan mencatat rancangan perangkat lunak dengan cara yang

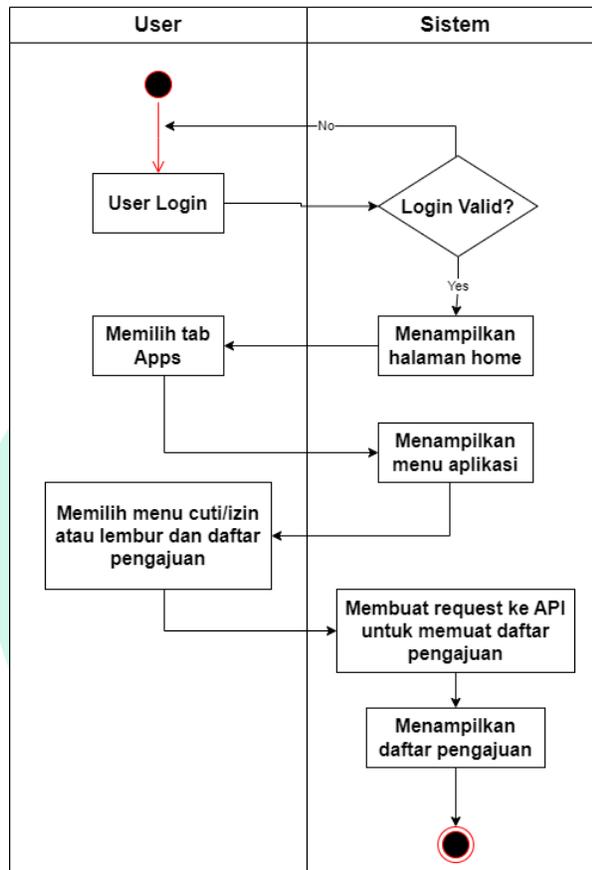
luas (Shelly, 2012). Diagram UML yang akan dirancang praktikan mencakup *Use Case Diagram* dan *Activity Diagram*.



Gambar 3.4 Use Case untuk Fitur Pengajuan Cuti/Izin dan Lembur

Gambar 3.4 menunjukkan use case diagram untuk fitur penambahan cuti/izin dan lembur. Use case diagram adalah jenis diagram dalam Unified Modeling Language (UML) yang menggambarkan interaksi antara aplikasi yang sedang dianalisis dan aktor yang berinteraksi dengan sistem. Aktor adalah entitas yang terlibat dalam interaksi dengan sistem tersebut. Diagram tersebut memiliki tiga *use case* untuk aktor pembuat pengajuan atau maker cuti/izin dan lembur dan dua *use case* untuk aktor approver. Lima *use case* tersebut akan dibuatkan *activity diagram*.

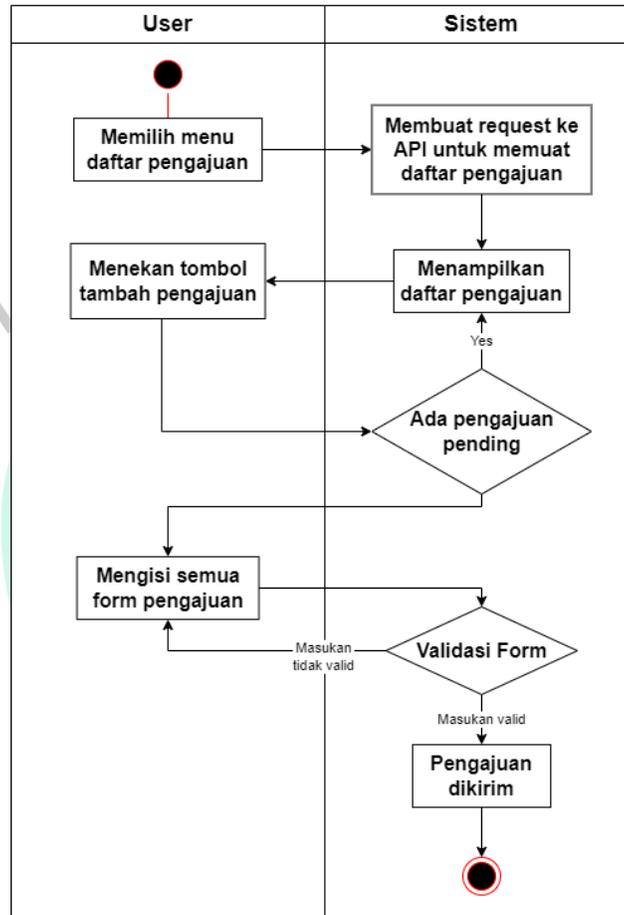
Activity Diagram Daftar Pengajuan



Gambar 3.5 Activity Diagram untuk use case Daftar Pengajuan Cuti/izin dan Lembur

Gambar 3.5 menyajikan *activity diagram* untuk pengguna yang ingin melihat daftar pengajuan cuti/izin maupun lembur yang pernah mereka buat. Pertama, halaman login ditampilkan sebelum mereka dapat mengakses menu Daftar Pengajuan. Selanjutnya, setelah pengguna berhasil melakukan proses login, mereka akan memilih tab “Apps” dan kemudian memilih menu “Daftar Pengajuan”.

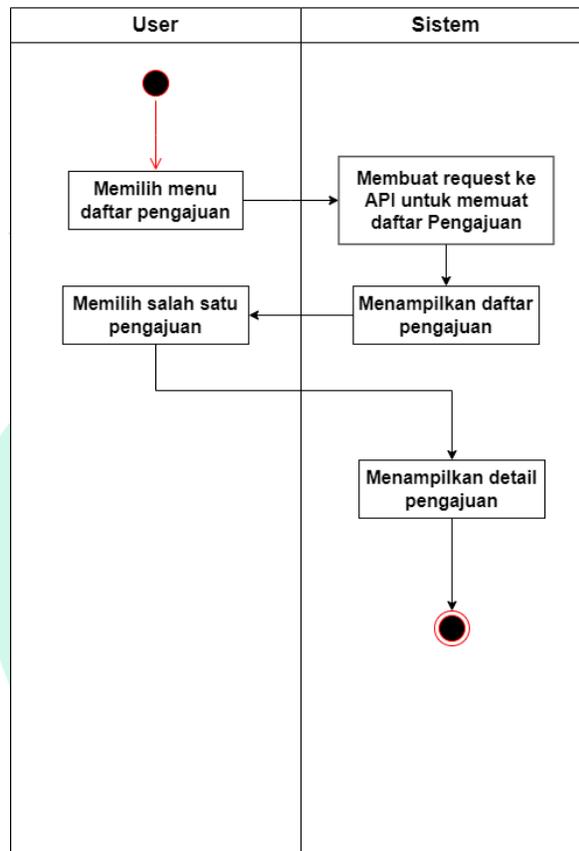
Activity Diagram Membuat Pengajuan



Gambar 3.6 Activity Diagram untuk use case Membuat Pengajuan Cuti/Izin dan Lembur

Gambar 3.6 menyajikan Activity Diagram untuk daftar pengajuan cuti/izin dan lembur. Activity Diagram ini dimulai dengan pengguna (user) memilih menu daftar pengajuan. Selanjutnya, sistem akan membuat permintaan ke API untuk memuat daftar pengajuan. Pengguna kemudian menekan tombol tambah pengajuan untuk memulai proses pengajuan baru. Pengajuan hanya bisa dibuat jika pengguna tidak memiliki pengajuan yang pending. Sistem atau antar muka akan menampilkan formular pengajuan yang harus diisi pengguna, dan setelah semua diisi, sistem akan melakukan validasi formular. Jika ada bagian formulir masih belum terisi, sistem akan menampilkan peringatan untuk mengisi bagian tersebut. Terakhir, sistem akan menyimpan formular jika tidak ada masalah dalam validasi formulir.

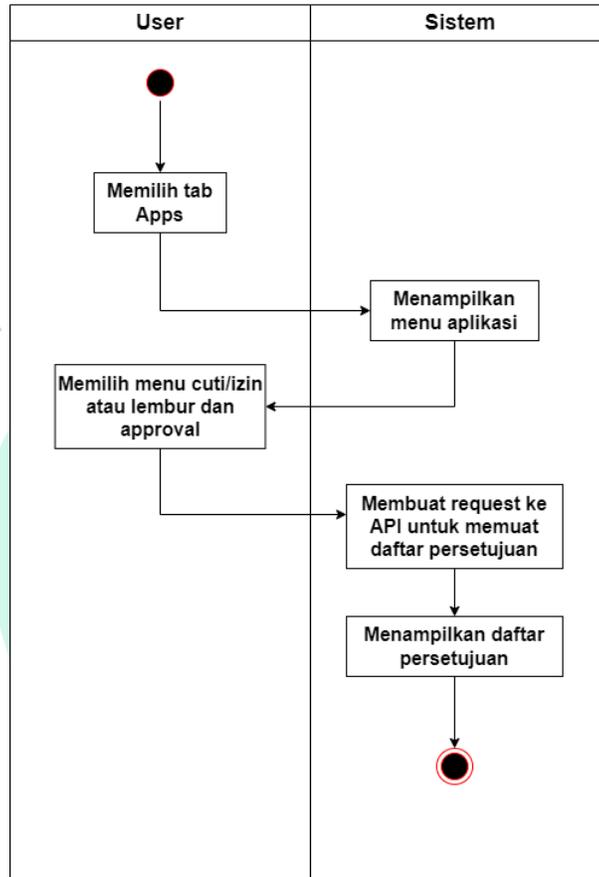
Activity Diagram Melihat Detail Pengajuan



Gambar 3.7 Activity Diagram untuk use case Melihat Detail Pengajuan Cuti/Izin dan Lembur

Activity diagram diawali dengan pengguna memilih menu daftar pengajuan cuti/izin dan lembur. Selanjutnya, sistem akan membuat permintaan ke API untuk melihat semua pengajuan yang pernah diajukan pengguna. Pengguna akan mengklik tombol yang tersedia dalam *card* pengajuan untuk melihat detail pengajuan,

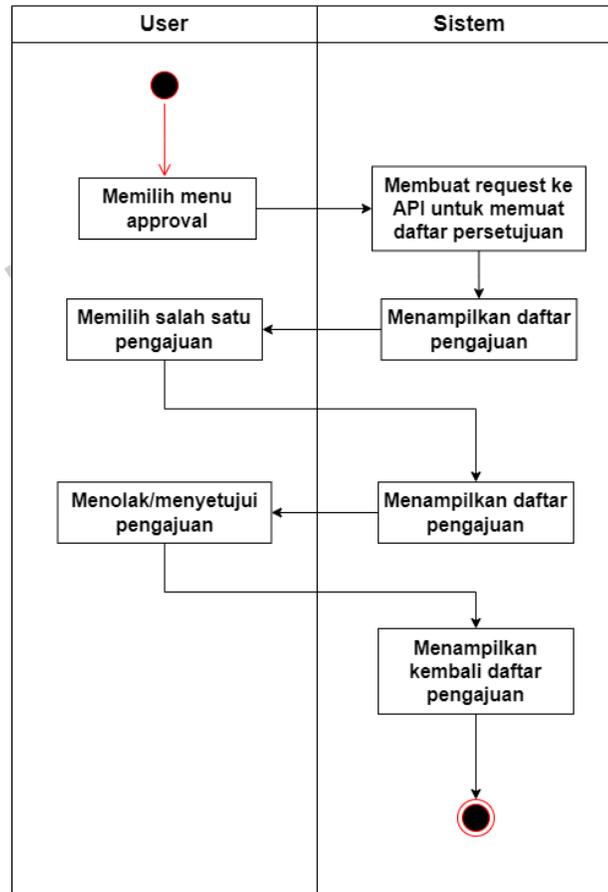
Activity Diagram Daftar Persetujuan



Gambar 3.8 Activity Diagram untuk use case Melihat Daftar Persetujuan Cuti/Izin dan Lembur

Gambar 3.8 menjelaskan Activity Diagram untuk melihat daftar pengajuan cuti/izin dan lembur yang menunggu persetujuan dari pengguna. Pertama, pengguna akan memilih tab “Apps” kemudian memilih menu “Cuti/izin” atau “Lembur”. Selanjutnya sistem akan menampilkan daftar persetujuan setelah berhasil melakukan permintaan kepada API Cuti/Izin atau Lembur.

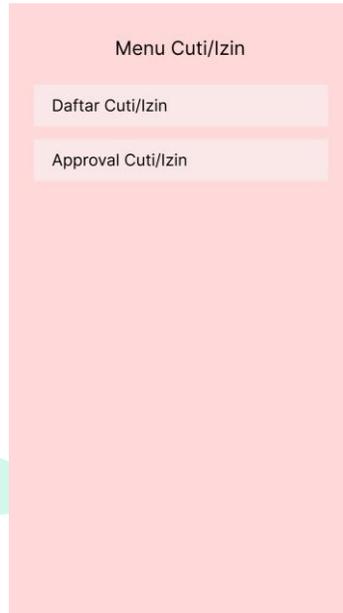
Activity Diagram Menolak/Menyetujui Persetujuan



Gambar 3.9 Activity Diagram untuk use case Menolak/Menyetujui Persetujuan

Dalam *Activity Diagram* yang disajikan pada Gambar 3.9, pengguna pertama akan memilih menu “Approval” dan kemudian memilih salah satu pengajuan dengan menekan tombol “Lihat Detail Pengajuan”. Pengguna selanjutnya melihat detail pengajuan, terdapat opsi untuk menolak dan menyetujui pengajuan. Setelah melakukan proses persetujuan, pengguna akan di bawa kembali ke halaman daftar pengajuan.

Setelah praktikan merancang diagram UML, praktikan membuat *mockup front-end* sebelum mengimplementasikan aplikasi ke dalam program. Tujuan dari pembuatan *mockup* aplikasi adalah untuk menunjukkan secara jelas bagaimana sebuah aplikasi akan dikembangkan.



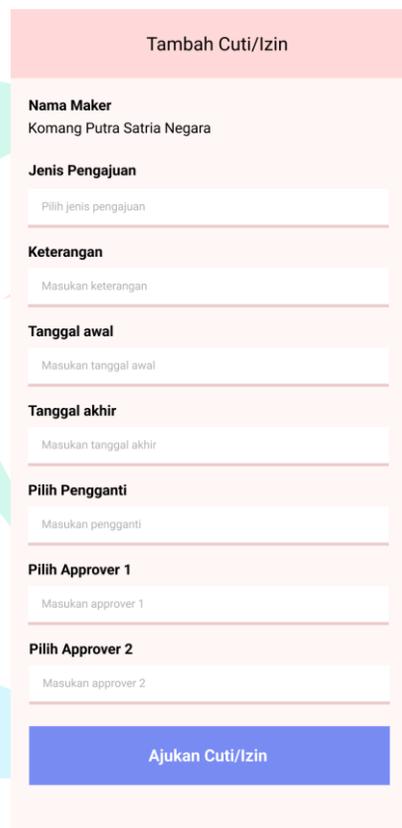
Gambar 3.10 Mockup Menu Cuti/Izin

Gambar 3.10 menunjukkan *mockup* halaman menu cuti/izin. Halaman ini berisi dua tombol yang berguna untuk mengakses daftar cuti/izin dan persetujuan (*approval*) cuti/izin.



Gambar 3.11 Mockup Daftar Cuti/Izin

Gambar 3.11 menampilkan *mockup* halaman Daftar Cuti/Izin. Halaman ini menampilkan daftar atau *list* pengajuan yang pernah di buat. Setiap pengajuan dalam daftar ini ditampilkan informasi ringkas, seperti jenis cuti, periode pengajuan, dan status pengajuan. Pengguna bisa membuat pengajuan baru dengan menekan *floating action button* yang berlogo tanda tambah (+). *Mockup* ini juga digunakan untuk halaman Daftar Approval Cuti/Izin dengan sedikit perbedaan, yakni tombol tanda tambah (+) dihilangkan.



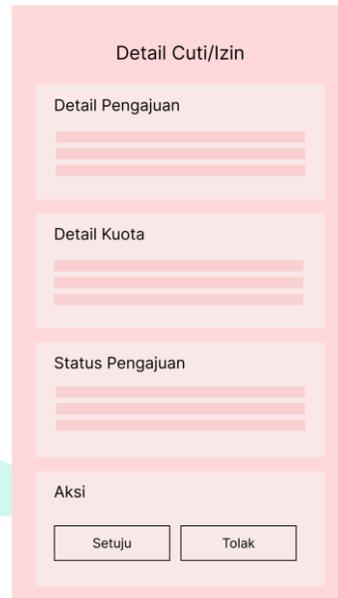
The image shows a vertical form titled "Tambah Cuti/Izin" (Add Leave/Permission). The form is divided into several sections, each with a label and a corresponding input field:

- Nama Maker**: Komang Putra Satria Negara
- Jenis Pengajuan**: Pilih jenis pengajuan
- Keterangan**: Masukan keterangan
- Tanggal awal**: Masukan tanggal awal
- Tanggal akhir**: Masukan tanggal akhir
- Pilih Pengganti**: Masukan pengganti
- Pilih Approver 1**: Masukan approver 1
- Pilih Approver 2**: Masukan approver 2

At the bottom of the form is a blue button labeled "Ajukan Cuti/Izin" (Submit Leave/Permission).

Gambar 3.12 Mockup Penambahan Cuti/Izin

Gambar 3.12 menampilkan *mockup* halaman penambahan cuti/izin. Pada layar ini, pengguna akan dapat memasukkan informasi terkait pengajuan yang mereka buat, seperti jenis pengajuan, keterangan, tanggal awal dan akhir izin, memilih pengganti serta approver pertama dan kedua.



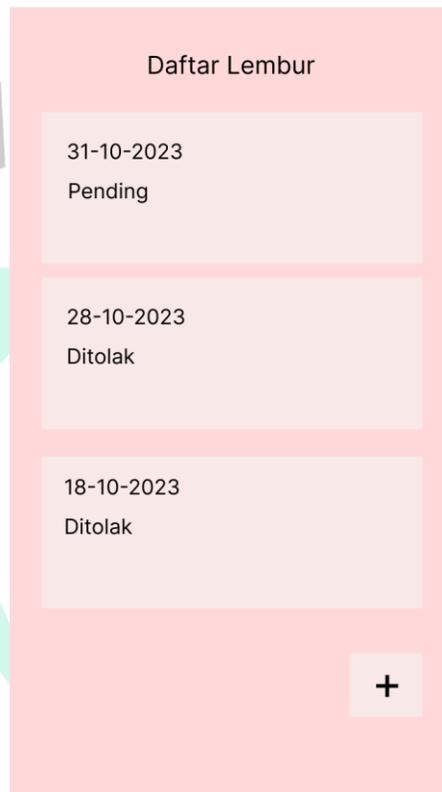
Gambar 3.13 Mockup Detail dan Approval Cuti/Izin

Gambar 3.13 menyajikan *mockup* halaman Detail sekaligus Approval Cuti/Izin. Dalam *mockup*, ditampilkan informasi yang berkaitan dengan permohonan cuti/izin seperti jumlah kuota yang digunakan, status dan posisi approver, dan tombol untuk menyetujui sekaligus menolak permohonan jika pengguna mempunyai posisi sebagai *approver*.



Gambar 3.14 Mockup Menu Lembur

Gambar 3.14 menunjukkan *mockup* halaman menu lembur yang memuat dua tombol, yaitu tombol akses untuk daftar lembur dan tombol persetujuan cuti/izin.



Gambar 3.15 Mockup Menu Daftar Lembur

Gambar 3.15 menampilkan *mockup* halaman Daftar Lembur. Halaman ini menampilkan daftar atau *list* pengajuan yang pernah di buat, dengan setiap pengajuan menampilkan informasi ringkas tentang periode lembur dan status pengajuan. Pengguna dapat membuat pengajuan baru dengan menekan tombol yang berlogo tanda tambah (+). *Mockup* ini juga digunakan untuk halaman Daftar Approval Cuti/Izin dengan satu perbedaan, yakni tombol tanda tambah (+) dihilangkan.

Gambar 3.16 Mockup Menu Tambah Lembur

Gambar 3.16 menampilkan *mockup* halaman penambahan lembur. Pada layar ini, pengguna dapat memasukkan informasi terkait pengajuan yang mereka buat, seperti detail pekerjaan yang dilakukan, pekerja yang terlibat dalam kegiatan lembur, waktu pelaksanaan lembur, serta pemilihan pengawas dan approver lembur.

3.2.2 Implementasi

Pada tahap implementasi, praktikan menggunakan perangkat lunak dengan spesifikasi seperti berikut ini.

| No. | Nama Perangkat Lunak | Spesifikasi Perangkat Lunak |
|-----|----------------------|-----------------------------|
| 1. | Operating System | Windows 10 |
| 2. | Frontend Framework | React Native 0.67.3 |
| 3. | IDE | Visual Studio Code 1.83.1 |

Tabel 3.2 Spesifikasi Perangkat Lunak

Adapun spesifikasi perangkat keras yang digunakan praktikan dalam membangun proyek aplikasi seperti berikut ini.

| No. | Nama Perangkat Keras | Spesifikasi Perangkat Keras |
|-----|----------------------|------------------------------|
| 1. | Processor | Intel Core i5-7300HQ |
| 2. | GPU | Nvidia GeForce GTX 1050 4 GB |
| 3. | RAM | 16 GB |

Tabel 3.3 Spesifikasi Perangkat Lunak

Praktikan juga menggunakan perangkat *smartphone* untuk tahap pengujian dengan spesifikasi sebagai berikut.

| No. | Nama Komponen | Spesifikasi Perangkat Lunak |
|-----|------------------|-----------------------------|
| 1. | Operating System | Android 12 |
| 2. | CPU | Exynos 9820 |
| 3. | RAM | 8 GB |

Pseudocode dari potongan kode program di atas dijelaskan pada bagian berikut.

1. Definisi komponen fungsional React: `AddLeaveApplication`.
2. Definisi fungsi untuk mendapatkan hari kerja di antara dua tanggal.
3. Definisi fungsi untuk mendapatkan data jenis pengajuan, daftar pengganti, dan daftar approver dari API.
4. Definisi fungsi untuk mengirim pengajuan cuti
5. Di dalam komponen:
 - a. Gunakan hook `useFocusEffect` untuk mendapatkan data dari API
 - b. Gunakan hook `useEffect` untuk melakukan validasi form
5. Menangani kasus saat `approverList`, `penggantiList`, `jenisPengajuanList` masih `undefined`, tampilkan indikator `loading`
6. Melakukan rendering tampilan utama dalam komponen `Container`

Hasil dari realisasi *pseudocode* dan kode program *front-end* penambahan cuti/izin disajikan pada Gambar 3.18. Pada halaman ini, terdapat pilihan untuk memilih jenis pengajuan cuti/izin, keterangan untuk menjelaskan alasan mengambil cuti, tanggal awal dan akhir cuti, pengganti, dan *approver 1* dan *approver 2*. Halaman ini juga menampilkan jumlah hak cuti yang dimiliki pengguna. Untuk pengguna yang memilih jenis “Izin Sakit dengan Surat Dokter”, mereka harus memasukan bukti pendukung seperti surat dari dokter.

← Tambah Cuti/Izin

Nama (NIP)

Jenis Pengajuan
Pilih jenis pengajuan

Keterangan
Masukan Keterangan

Tanggal Awal
Pilih tanggal awal pengajuan

Tanggal Akhir
Pilih tanggal akhir pengajuan

Hapus Tanggal

Jumlah Hak Cuti
10

Jumlah yang digunakan
0

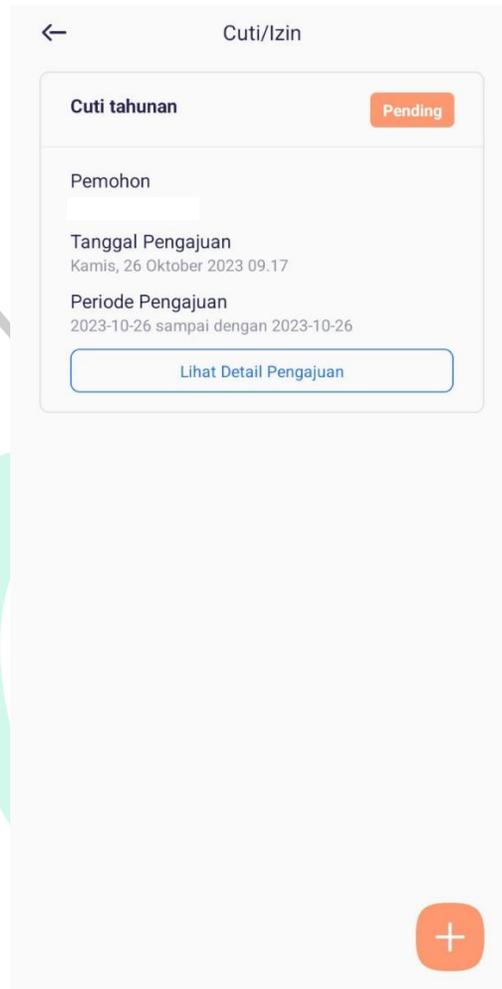
Jumlah sisa kuota
10

Pengganti
Pilih pengganti

Approver 1

Gambar 3.18 Tampilan Halaman Penambahan Cuti/Izin

Gambar 3.19 menyajikan kode program untuk menampilkan daftar cuti/izin. Halaman ini menggunakan komponen FlatList untuk menampilkan daftar item yang bisa di-*scroll*. Halaman ini memiliki tombol *floating action* untuk menambahkan cuti jika pengguna tidak memiliki pengajuan yang statusnya dalam tahap pending.



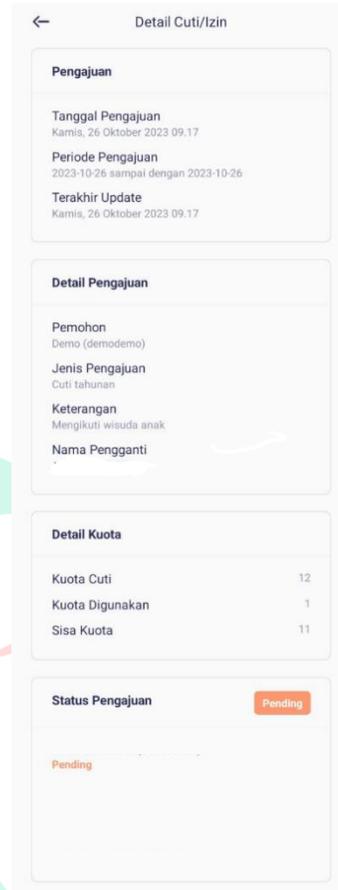
Gambar 3.20 Tampilan halaman Daftar Cuti/Izin

Gambar 3.20 menyajikan tampilan halaman daftar cuti/izin. Halaman ini menampilkan hingga lima pengajuan cuti/izin yang paling baru dari pengguna. Setiap pengajuan ditampilkan informasi ringkas, seperti pemohon, tanggal pengajuan, periode pengajuan, dan status pengajuan. Untuk melihat informasi yang lebih detail, pengguna bisa menekan tombol "Lihat Detail Pengajuan". Untuk membuat pengajuan baru, pengguna bisa menekan tombol tambah ("+") yang terletak di sebelah kanan bawah layar.

Pseudocode dari potongan kode program di atas dijelaskan pada bagian berikut.

1. Definisi komponen fungsional React:
OvertimeApplicationDetails.
2. Di dalam komponen:
 - a. Gunakan hook `useRoute` dari `'@react-navigation/native'` untuk mendapatkan navigation route.
 - b. Gunakan hook `useDetailLembur` untuk mendapatkan data terkait detail lembur pekerja
3. Definisi fungsi:
 - a. `downloadFilePengajuan`: membuka URL untuk mendownload file pengajuan
 - b. `downloadFilePendukung`: membuka URL untuk mendownload file pendukung
4. Membuat fungsi `hasDownloadableFile` untuk memeriksa apakah ada file yang bisa di download Berdasarkan status pengajuan.
5. Menangani kasus saat `detailLemburData` masih `undefined`, tampilkan indicator loading
6. Melakukan rendering tampilan utama dalam komponen `Container`

Hasil dari realisasi *pseudocode* dan kode program disajikan pada Gambar 3.22. Halaman ini terdiri dari 4 bagian, yakni informasi umum terkait pengajuan, detail pengajuan, detail kuota dan status pengajuan. Jika pengajuan berhasil disetujui, maka akan muncul tombol untuk mendownload file pengajuan yang berisi surat resmi dari perusahaan bahwa pengajuan berhasil disetujui.



Gambar 3.22 Tampilan Halaman Detail Cuti/izin

Gambar 3.23 menampilkan kode program untuk tampilan halaman daftar persetujuan (*approval*). Halaman ini menggunakan komponen `FlatList` untuk menampilkan daftar item yang dapat di-*scroll*. Komponen tersebut berguna saat memiliki banyak item yang perlu ditampilkan. `FlatList` juga memungkinkan pengguna untuk memuat ulang konten dengan fitur *pull down to refresh*.

```
import { useState } from 'react';
import { useNavigation } from '@react-navigation/native';
import { useFocusEffect } from '@react-navigation/native';

const LeaveApproval = () => {
  const [status, setStatus] = useState('');
  const [message, setMessage] = useState('');
  const navigation = useNavigation();

  useFocusEffect(
    React.useCallback(() => {
      // Logic for handling the leave approval process
    }, [])
  );

  // ... (rest of the code is blurred)
};

export default LeaveApproval;
```

Gambar 3.23 Kode Program Front End Daftar Persetujuan Cuti/izin (Disamarkan)

Pseudocode untuk potongan kode program pada Gambar 3.23 dijelaskan pada bagian berikut.

1. Definisi komponen fungsional React: LeaveApproval.
2. Di dalam komponen:
 - a. Gunakan hook useNavigation, useState, dan useFocusEffect.

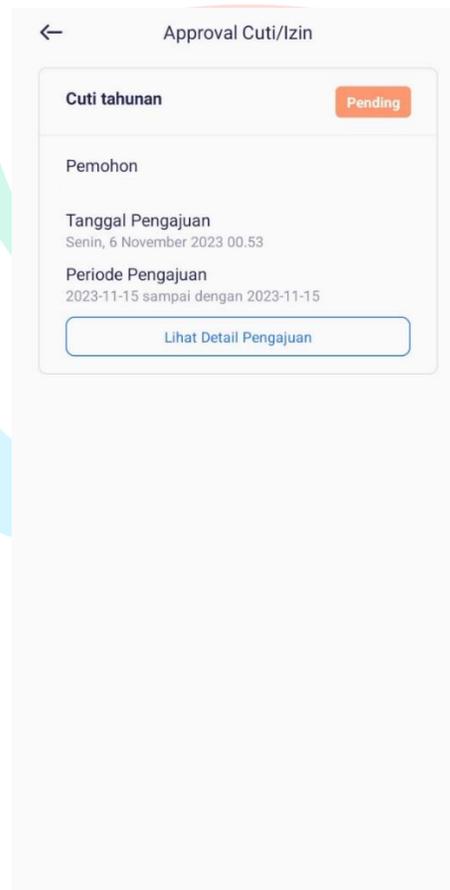
3. Definisi fungsi `fetchData` untuk mendapatkan data terkait list persetujuan cuti/izin:

- a. Gunakan `AsyncStorage` untuk mendapatkan ID pekerja
- b. Membuat request asynchronous menggunakan `axios` untuk mendapatkan list persetujuan cuti/izin dan status pengajuan

4. Gunakan hook `useFocusEffect` untuk memanggil fungsi `fetchData`

5. Tampilkan indicator loading saat variable `cutiIzinData` atau `jenisIzin` cuti masih `undefined`

6. Melakukan rendering tampilan utama dalam komponen `Container`



Gambar 3.24 Front End Daftar Persetujuan Cuti/izin

Gambar 3.24 menampilkan halaman daftar persetujuan cuti/izin. Halaman ini menampilkan hingga lima pengajuan cuti/izin terbaru dari pengguna. Setiap pengajuan menampilkan informasi ringkas, seperti pemohon, tanggal pengajuan, periode pengajuan, dan status pengajuan. Pengguna dapat melihat informasi lebih detail dengan menekan tombol “Lihat Detail Pengajuan”.



```

<div class="row">
  <div class="col">
    <div class="card">
      <div class="card-body">
        <div class="d-flex justify-content-between">
          <div>
            <div class="text">
              <span>{{user.username}}</span>
            </div>
            <div class="text">
              <span>{{request.created_at}}</span>
            </div>
            <div class="text">
              <span>{{request.start_date}}</span> - <span>{{request.end_date}}</span>
            </div>
            <div class="text">
              <span>{{request.status}}</span>
            </div>
          </div>
          <div>
            <button class="btn btn-primary">Lihat Detail Pengajuan</button>
          </div>
        </div>
      </div>
    </div>
  </div>
  <div class="col">
    <div class="card">
      <div class="card-body">
        <div class="d-flex justify-content-between">
          <div>
            <div class="text">
              <span>{{user.username}}</span>
            </div>
            <div class="text">
              <span>{{request.created_at}}</span>
            </div>
            <div class="text">
              <span>{{request.start_date}}</span> - <span>{{request.end_date}}</span>
            </div>
            <div class="text">
              <span>{{request.status}}</span>
            </div>
          </div>
          <div>
            <button class="btn btn-primary">Lihat Detail Pengajuan</button>
          </div>
        </div>
      </div>
    </div>
  </div>
  <div class="col">
    <div class="card">
      <div class="card-body">
        <div class="d-flex justify-content-between">
          <div>
            <div class="text">
              <span>{{user.username}}</span>
            </div>
            <div class="text">
              <span>{{request.created_at}}</span>
            </div>
            <div class="text">
              <span>{{request.start_date}}</span> - <span>{{request.end_date}}</span>
            </div>
            <div class="text">
              <span>{{request.status}}</span>
            </div>
          </div>
          <div>
            <button class="btn btn-primary">Lihat Detail Pengajuan</button>
          </div>
        </div>
      </div>
    </div>
  </div>
  <div class="col">
    <div class="card">
      <div class="card-body">
        <div class="d-flex justify-content-between">
          <div>
            <div class="text">
              <span>{{user.username}}</span>
            </div>
            <div class="text">
              <span>{{request.created_at}}</span>
            </div>
            <div class="text">
              <span>{{request.start_date}}</span> - <span>{{request.end_date}}</span>
            </div>
            <div class="text">
              <span>{{request.status}}</span>
            </div>
          </div>
          <div>
            <button class="btn btn-primary">Lihat Detail Pengajuan</button>
          </div>
        </div>
      </div>
    </div>
  </div>
  <div class="col">
    <div class="card">
      <div class="card-body">
        <div class="d-flex justify-content-between">
          <div>
            <div class="text">
              <span>{{user.username}}</span>
            </div>
            <div class="text">
              <span>{{request.created_at}}</span>
            </div>
            <div class="text">
              <span>{{request.start_date}}</span> - <span>{{request.end_date}}</span>
            </div>
            <div class="text">
              <span>{{request.status}}</span>
            </div>
          </div>
          <div>
            <button class="btn btn-primary">Lihat Detail Pengajuan</button>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

Gambar 3.25 Kode Program Front End Detail Persetujuan Cuti/izin (Disamarkan)

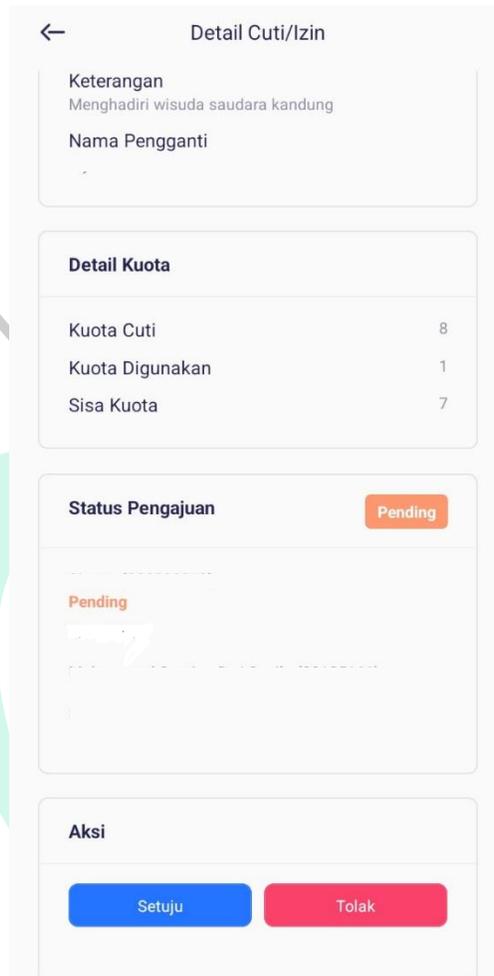
Gambar 3.25 menampilkan kode program untuk halaman detail persetujuan cuti/izin. Halaman ini memiliki struktur kode yang mirip dengan halaman Detail Pengajuan Cuti/Izin dengan perbedaan pada bagian paling bawah. Pada halaman

ini, terdapat dua tombol aksi, yaitu tombol “Setuju” dan “Tolak” untuk menangani persetujuan pengajuan tersebut.

Pseudocode untuk potongan kode program di atas dijelaskan pada bagian berikut.

1. Definisi komponen fungsional React: `LeaveApprovalDetails`.
2. Di dalam komponen:
 - a. Gunakan hook `useFocusEffect` untuk memanggil fungsi `fetchData`
3. Definisi fungsi `fetchData` untuk mendapatkan data terkait list persetujuan cuti/izin:
 - a. Gunakan `AsyncStorage` untuk mendapatkan ID pekerja
 - b. Membuat request asynchronous menggunakan `axios` untuk mendapatkan detail persetujuan cuti/izin dan status pengajuan
4. Definisi fungsi untuk menolak dan menyetujui persetujuan
5. Tampilkan indicator loading saat variable `cutiIzinData` cuti masih undefined
6. Melakukan rendering tampilan utama dalam komponen `Container`

Hasil dari realisasi *pseudocode* dan kode program halaman Detail Persetujuan Cuti/izin disajikan pada Gambar 3.26. Halaman ini mengikuti format dari halaman “Detail Pengajuan Cuti/Izin” yang terdiri dari 4 bagian, yakni informasi umum terkait pengajuan, detail pengajuan, detail kuota dan status pengajuan. Pada bagian paling bawah terdapat tombol aksi untuk menyetujui dan menolak pengajuan tersebut.



Gambar 3.26 Front End Detail Persetujuan Cuti/izin

Gambar 3.26 menyajikan kode program untuk menampilkan daftar lembur. Kode program untuk halaman ini memiliki struktur yang sama dengan halaman “Daftar Cuti/Izin” sehingga menggunakan komponen FlatList untuk menampilkan daftar yang dapat di-*scroll*. Halaman ini juga menyertakan tombol tambah (“+”) untuk membuat pengajuan lembur baru.

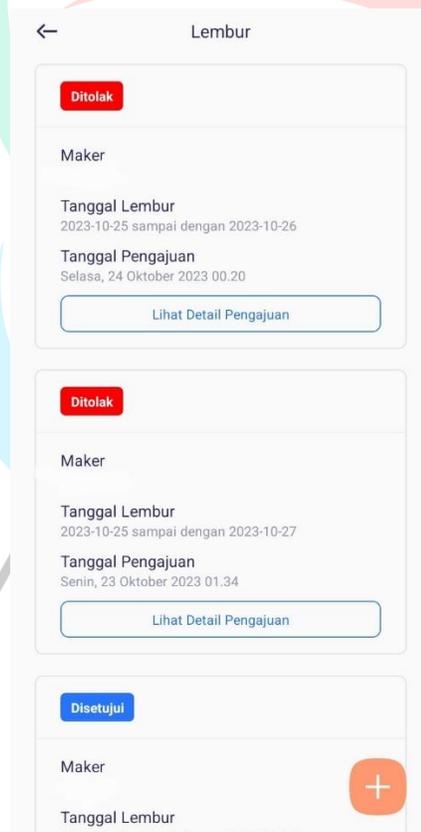


Gambar 3.27 Kode Program Front End Daftar Pengajuan Lembur (Disamarkan)

Gambar 3.27 menyajikan kode program untuk menampilkan daftar lembur. Kode program untuk halaman ini memiliki struktur yang sama dengan halaman “Daftar Cuti/Izin” sehingga menggunakan komponen FlatList untuk menampilkan daftar yang dapat di-*scroll*. Halaman ini juga menyertakan tombol tambah (“+”) untuk membuat pengajuan lembur baru.

Pseudocode untuk potongan kode program di atas dijelaskan pada bagian berikut.

1. Definisi komponen fungsional React: `OvertimeApplication`
2. Definisi fungsi `fetchData` untuk mendapatkan daftar lembur
3. Menangani kasus saat `lemburData` masih `undefined`, tampilkan `loading indicator`
4. Menampilkan rendering komponen utama dengan komponen `FlatList`
5. Jika masih terdapat pengajuan lembur yang `pending`, tampilkan `alert` saat pengguna mengklik tombol `tambah pengajuan`



Gambar 3.28 Halaman Daftar Pengajuan Lembur

Gambar 3.28 memperlihatkan tampilan daftar lembur. Halaman ini menampilkan hingga lima pengajuan lembur yang terbaru dari pengguna. Setiap pengajuan ditampilkan informasi singkat, seperti pembuat pengajuan lembur, tanggal pengajuan, periode pengajuan, dan status pengajuan. Untuk melihat detail lebih lanjut, pengguna bisa menekan tombol "Lihat Detail Pengajuan". Untuk membuat pengajuan baru, pengguna bisa menekan tombol tambah ("+") yang terletak di sudut kanan bawah layar.



Gambar 3.29 Kode Program Front End Penambahan Lembur (Disamarkan)

Gambar 3.29 menyajikan potongan kode program untuk halaman penambahan lembur. Halaman ini menggunakan komponen CustomDropDown untuk memilih satu opsi dari beberapa pilihan dan Input untuk memasukan deskripsi lembur

pekerjaan. Selain itu, untuk menambahkan pekerja yang terlibat dalam kegiatan lembur, terdapat *dropdown* dinamis yang memungkinkan penambahan maupun penghapusan pekerja.

Pseudocode dari potongan kode program di atas dijelaskan pada bagian berikut

1. Definisi komponen fungsional React: `AddOvertimeApplication`.
2. Definisi fungsi untuk mendapatkan data pekerja dan daftar approver dari API.
4. Definisi fungsi untuk mengirim pengajuan lembur
5. Di dalam komponen:
 - a. Gunakan hook `useFocusEffect` untuk mendapatkan data dari API
 - b. Gunakan hook `useEffect` untuk melakukan validasi form
5. Menangani kasus saat `pekerjaList` dan `approverList` masih `undefined`, tampilkan indicator loading
6. Melakukan rendering tampilan utama dalam komponen `Container`

Hasil dari realisasi *pseudocode* dan kode program halaman penambahan lembur disajikan pada Gambar 3.29. Dalam halaman ini, pengguna akan mengisi beberapa kolom formulir untuk pengajuan lembur, yakni detail pekerjaan lembur, pekerja yang terlibat dalam kegiatan lembur, tanggal awal dan akhir kegiatan lembur, pemilihan pengawas dan *approver*.

Pengguna dapat menambahkan maksimal 5 pekerja lembur. Tombol “Ajukan Lembur” digunakan untuk mengirim pengajuan lembur dan hanya bisa ditekan jika semua kolom sudah diisi.

← Tambah Lembur

Nama Maker (NIP)

Detail Pekerjaan

Masukan Detail Pekerjaan

Pekerja 1

Pilih Pekerja 1

+ Tambah Pekerja

Tanggal awal lembur

Pilih tanggal awal pengajuan

Tanggal akhir lembur

Pilih tanggal akhir pengajuan

Hapus Tanggal

Total waktu lembur: 0 hari

Pengawas

Pilih pengawas

Approver

Pilih approver

Ajukan Lembur

Gambar 3.30 Tampilan Halaman Penambahan Lembur

Gambar 3.30 menampilkan potongan kode program untuk tampilan detail pengajuan lembur. Halaman ini menggunakan struktur kode yang serupa dengan halaman detail pengajuan cuti/izin sehingga terdapat penggunaan komponen seperti *ScrollView* agar konten dapat di-*scroll*.



Gambar 3.31 Kode Program Front End Detail Pengajuan Lembur (Disamarkan)

Pseudocode dari potongan kode program di atas dijelaskan pada bagian berikut

1. Definisi komponen fungsional React: `OvertimeApplicationDetails`.
2. Di dalam komponen:
 - a. Gunakan hook `useRoute` dari `'@react-navigation/native'` untuk mendapatkan navigation route.
 - b. Gunakan hook `useDetailLembur` untuk mendapatkan data terkait detail lembur pekerja

3. Definisi fungsi:

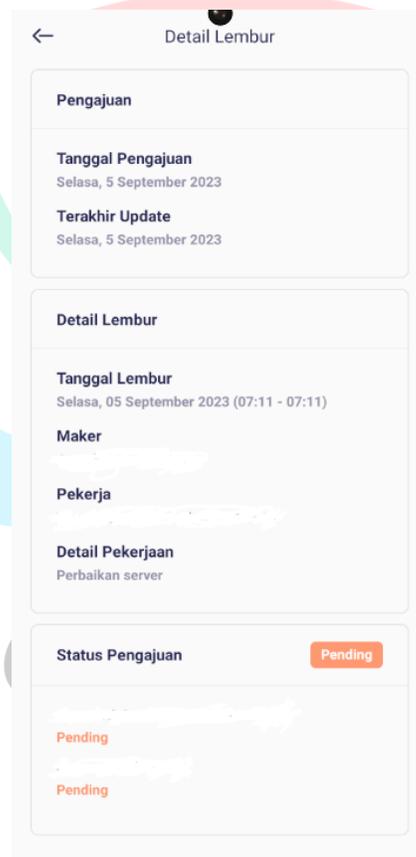
a. `downloadFilePengajuan`: membuka URL untuk mendownload file pengajuan

b. `downloadFilePendukung`: membuka URL untuk mendownload file pendukung

4. Membuat fungsi `hasDownloadableFile` untuk memeriksa apakah ada file yang bisa di download Berdasarkan status pengajuan.

5. Menangani kasus saat `detailLemburData` masih undefined, tampilkan indicator loading

6. Melakukan rendering tampilan utama dalam komponen Container



Gambar 3.32 Tampilan Halaman Detail Lembur

Gambar 3.31 menyajikan tampilan halaman detail lembur. Halaman ini terdiri dari 3 bagian, yakni informasi umum terkait pengajuan lembur, detail pengajuan lembur, dan status pengajuan. Jika pengajuan berhasil disetujui, maka akan muncul tombol untuk mendownload file pengajuan yang berisi surat resmi dari perusahaan bahwa pengajuan lembur berhasil disetujui.



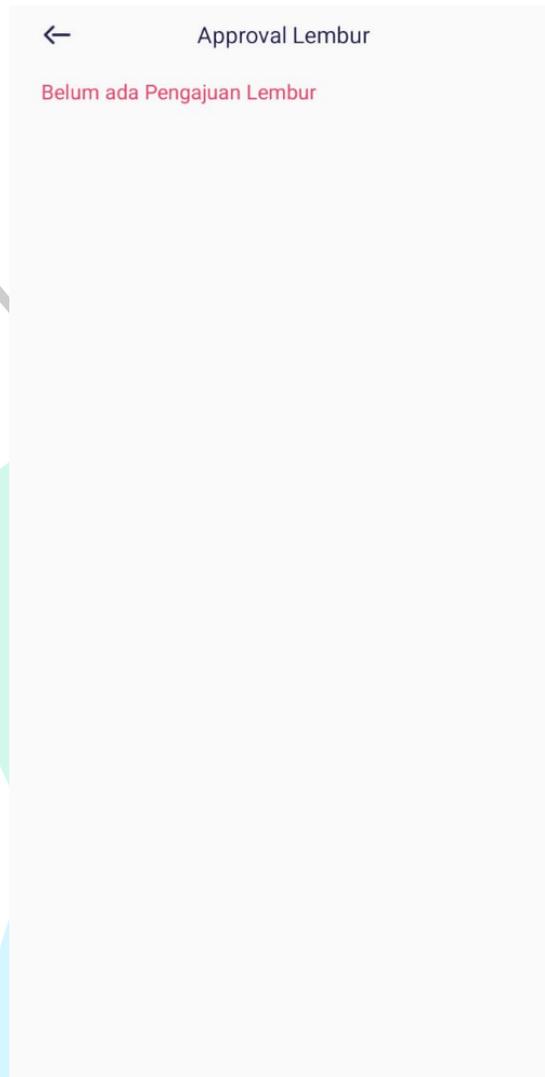
Gambar 3.33 Kode Program Front End Daftar Persetujuan Lembur (Disamarkan)

Gambar 3.23 menampilkan kode program untuk tampilan halaman daftar persetujuan (*approval*) lembur. Halaman ini menggunakan komponen FlatList untuk menampilkan daftar item yang dapat di-*scroll*. Komponen tersebut berguna saat memiliki banyak item yang perlu ditampilkan. Apabila tidak ada pengajuan lembur yang menunggu persetujuan, tampilan akan menyajikan teks yang memberikan informasi tentang keadaan tersebut

Pseudocode dari potongan kode program di atas dijelaskan pada bagian berikut.

1. Definisi komponen fungsional React: `OvertimeApproval`.
2. Di dalam komponen:
 - a. Gunakan hook `useNavigation`, `useState`, dan `useFocusEffect`.
3. Definisi fungsi `fetchData` untuk mendapatkan data terkait list persetujuan lembur:
 - a. Gunakan `AsyncStorage` untuk mendapatkan ID pekerja
 - b. Membuat request asynchronous menggunakan `axios` untuk mendapatkan list persetujuan lembur dan status pengajuan
4. Gunakan hook `useFocusEffect` untuk memanggil fungsi `fetchData`
5. Tampilkan indicator loading saat variable `lemburData` undefined
6. Melakukan rendering tampilan utama dalam komponen `Container` dan `FlatList`

Hasil dari realisasi *pseudocode* dan potongan kode program pada Gambar 3.32. Halaman ini menampilkan hingga lima pengajuan lembur terbaru dari pengguna. Setiap pengajuan menampilkan informasi ringkas, seperti maker, tanggal pengajuan, periode pengajuan, dan status pengajuan. Pengguna dapat melihat informasi lebih detail dengan menekan tombol "Lihat Detail Pengajuan".



Gambar 3.34 Tampilan Daftar Persetujuan Lembur

Gambar 3.34 menampilkan halaman daftar persetujuan lembur. Halaman ini menampilkan hingga lima pengajuan lembur terbaru dari pengguna. Setiap pengajuan menampilkan informasi ringkas, seperti maker, tanggal pengajuan, periode pengajuan, dan status pengajuan. Pengguna dapat melihat informasi lebih detail dengan menekan tombol "Lihat Detail Pengajuan".



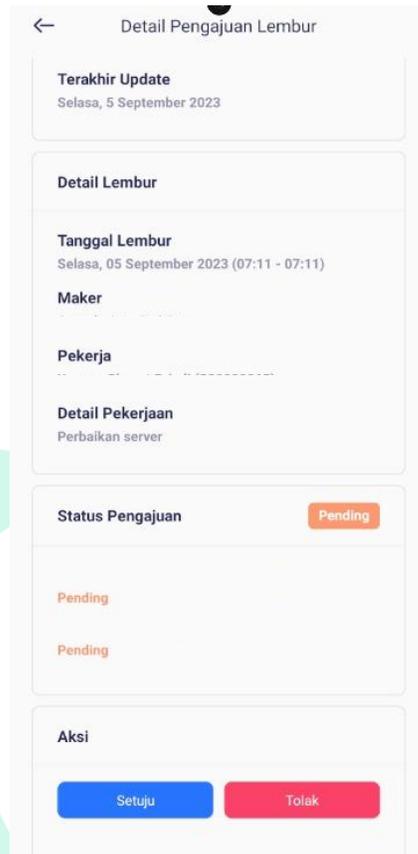
Gambar 3.35 Kode Program Tampilan Detail Persetujuan Lembur (Disamarkan)

Gambar 3.35 menampilkan kode program untuk halaman detail persetujuan lembur. Halaman ini memiliki struktur kode yang mirip dengan halaman Detail Pengajuan Lembur dengan perbedaan pada bagian paling bawah. Pada halaman ini, terdapat dua tombol aksi, yaitu tombol “Setuju” dan “Tolak” untuk menangani persetujuan pengajuan tersebut.

Pseudocode untuk potongan kode program di atas dijelaskan pada bagian berikut.

1. Definisi komponen fungsional React:
OvertimeApplicationDetails.
2. Di dalam komponen:
 - a. Gunakan hook `useRoute` dari `'@react-navigation/native'` untuk mendapatkan `navigation route`.
 - b. Gunakan hook `useDetailLembur` untuk mendapatkan data terkait detail lembur pekerja
3. Definisi fungsi:
 - a. `downloadFilePengajuan`: membuka URL untuk mendownload file pengajuan
 - b. `downloadFilePendukung`: membuka URL untuk mendownload file pendukung
4. Membuat fungsi `hasDownloadableFile` untuk memeriksa apakah ada file yang bisa di download Berdasarkan status pengajuan.
5. Menangani kasus saat `detailLemburData` masih `undefined`, tampilkan `indicator loading`
6. Definisi fungsi untuk menolak dan menyetujui persetujuan
7. Melakukan rendering tampilan utama dalam komponen `Container`

Gambar 3.35 menyajikan tampilan halaman detail lembur. Halaman ini mengikuti format dari halaman “Detail Pengajuan Lembur” yang terdiri dari 4 bagian, yakni informasi umum terkait pengajuan, detail pengajuan, dan status pengajuan. Persetujuan dan penolakan pengajuan dilakukan dengan menekan tombol aksi yang terletak pada bagian paling bawah halaman.



Gambar 3.36 Tampilan Detail Persetujuan Lembur

3.2.3 Pengujian

Praktikan menggunakan metode fungsional atau *black box* dalam tahap pengujian. Pengujian fungsional merujuk pada pengujian yang melibatkan hanya observasi keluaran untuk nilai input tertentu, dan tidak ada upaya untuk menganalisis kode yang menghasilkan keluaran (B. B. Agarwal et al, 2010).

| No. | Skenario Pengujian | Hasil yang Diharapkan | Kesimpulan |
|-----|---------------------------------|--|------------|
| 1. | Menu daftar pengajuan cuti/izin | Aplikasi dapat menampilkan semua pengajuan cuti/izin yang pernah dibuat pengguna | Berhasil |

| | | | |
|----|---|--|----------|
| 2. | Menu detail pengajuan cuti/izin | Aplikasi dapat menampilkan detail pengajuan cuti/izin | Berhasil |
| 3. | Menu tambah cuti/izin | Aplikasi dapat menambah pengajuan cuti/izin yang baru | Berhasil |
| 4. | Menu daftar persetujuan (<i>approval</i>) cuti/izin | Aplikasi dapat menampilkan daftar pengajuan cuti/izin yang menunggu persetujuan | Berhasil |
| 5. | Menu detail persetujuan cuti/izin | Aplikasi dapat melihat detail pengajuan cuti/izin dan menolak maupun menyetujui permohonan cuti/izin | Berhasil |
| 6. | Menu daftar pengajuan lembur | Aplikasi dapat menampilkan semua pengajuan lembur yang pernah dibuat pengguna | Berhasil |
| 7. | Menu detail pengajuan lembur | Aplikasi dapat menampilkan detail pengajuan lembur | Berhasil |

| | | | |
|------|--|--|----------|
| 8.. | Menu tambah lembur | Aplikasi dapat menambah pengajuan lembur yang baru | Berhasil |
| 9. | Menu daftar persetujuan (<i>approval</i>) lembur | Aplikasi dapat menampilkan daftar pengajuan lembur yang menunggu persetujuan | Berhasil |
| 10.. | Menu detail persetujuan lembur | Aplikasi dapat melihat detail pengajuan lembur dan menolak maupun menyetujui permohonan lembur | Berhasil |

Tabel 3.4 Pengujian Black Box

Tabel 3.4 merupakan tabel skenario pengujian *Black Box* pada aplikasi MySAU. Pengujian dilakukan oleh Tim *Quality Assurance* SAU.

3.3 Kendala yang Dihadapi

Saat praktikan melaksanakan kegiatan magang MBKM di PT Siaga Abdi Utama, ditemukan beberapa kendala dalam melakukan pekerjaan sebagaimana disebutkan pada poin poin berikut ini.

1. Praktikan belum mempunyai dasar-dasar JavaScript maupun TypeScript, praktikan membutuhkan waktu tambahan untuk mempelajari cara membuat tampilan aplikasi menggunakan React Native.
2. Spesifikasi perangkat keras (laptop) yang dimiliki praktikan belum maksimal untuk dipakai pemrograman aplikasi berbasis mobile.
3. Pembimbing eksternal tidak mempunyai waktu yang cukup untuk melakukan sesi konsultasi.

3.4 Cara Mengatasi Kendala

Selama melakukan kegiatan magang MBKM di PT Siaga Abdi Utama, praktikan mendapatkan beberapa cara untuk mengatasi kendala tersebut, diantaranya:

1. mempelajari JavaScript maupun TypeScript secara *project-based* untuk memahami bagaimana penggunaan teknologi tersebut di dunia nyata,
2. menggunakan fasilitas komputer yang sudah disediakan di kantor, dan
3. berdiskusi dengan pembimbing eksternal untuk mengatur jadwal sesi bimbingan.

3.5 Pembelajaran yang diperoleh dari MBKM

● Kegiatan magang MBKM yang dilaksanakan di PT Siaga Abdi Utama memiliki manfaat yang dirasakan praktikan. Manfaat yang diperoleh adalah sebagai berikut ini:

1. memahami bagaimana kondisi dunia industri dalam bidang IT,
2. mendapatkan kemampuan berbicara dan komunikasi dengan rekan kerja maupun pembimbing eksternal,
3. mendapatkan kemampuan bekerja sama bersama rekan kerja dalam mengembangkan aplikasi dan memecahkan suatu permasalahan,
4. melatih kemampuan pemecahan masalah (*problem solving*) dan berpikir kritis saat melakukan pemrograman, dan
5. mengetahui *best practices* dalam menulis kode program agar terhindar dari bug dan mempermudah *refactoring* di kemudian hari.