

BAB 3

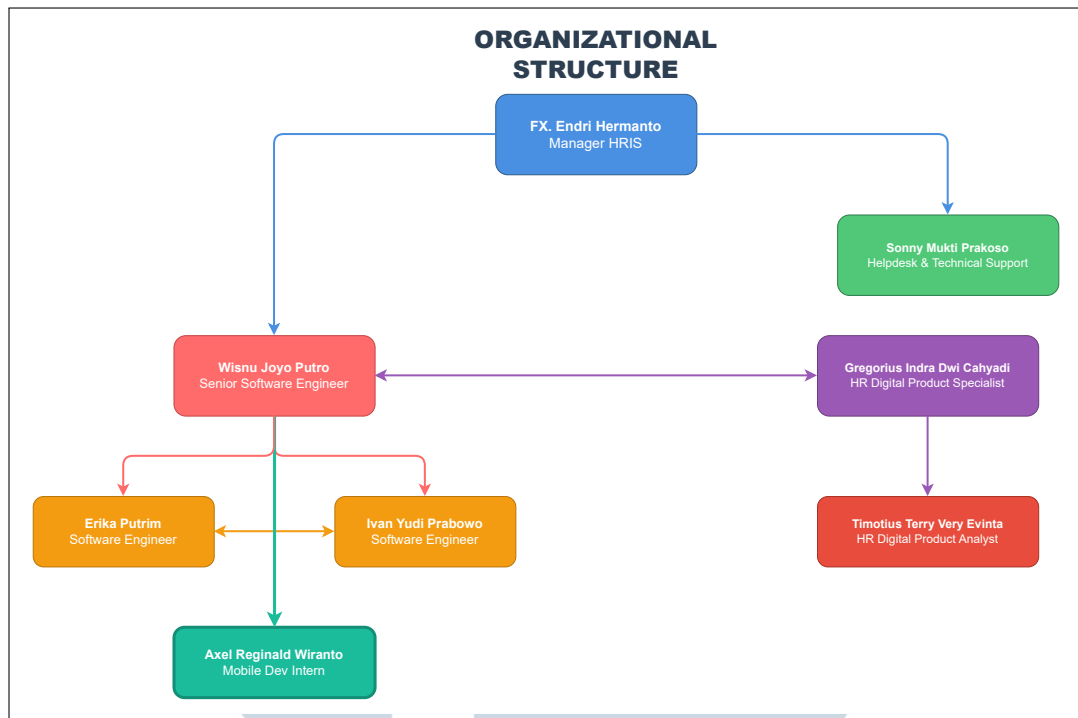
PELAKSANAAN KERJA MAGANG

Migrasi sistem *enterprise* dari *framework legacy* ke teknologi modern melampaui sekadar aktivitas teknis penulisan ulang kode—ia merupakan orkestrasi kompleks yang melibatkan pemahaman mendalam terhadap arsitektur sistem yang ada, translasi logika bisnis ke paradigma baru, serta penjaminan kontinuitas layanan bagi lebih dari 18.000 karyawan yang bergantung pada sistem tersebut setiap harinya. Dalam konteks inilah proyek pengembangan HR Mobile Portal dilaksanakan: sebuah inisiatif untuk mentransformasikan aplikasi berbasis PhoneGap yang telah berusia hampir satu dekade menjadi solusi React Native dengan Expo yang responsif, *maintainable*, dan siap menghadapi tuntutan era *mobile-first*.

3.1 Kedudukan dan Koordinasi

Kegiatan kerja magang dilaksanakan di *Functional Unit Corporate Human Resources* (CHR), tepatnya pada Departemen *Human Resource Information System* (HRIS) dengan posisi sebagai *Mobile Developer Intern*. Departemen HRIS berada di bawah kepemimpinan Bapak FX Endri Harmanto selaku manajer yang juga berperan sebagai pembimbing lapangan selama periode magang berlangsung. Secara struktural, departemen ini terbagi ke dalam tiga seksi utama, yaitu *Technical Support*, *Software Engineer*, dan *HR Digital Product*, yang seluruhnya berada di bawah koordinasi langsung beliau. Gambaran struktur organisasi dan kedudukan dalam Departemen HRIS ditunjukkan pada Gambar 3.1.

Divisi *HR Digital Product* memiliki tanggung jawab utama dalam menganalisis kebutuhan sistem serta merancang proses bisnis yang mendukung operasional sumber daya manusia. Divisi ini dipimpin oleh Bapak Gregorius Indra Dwi Cahyadi selaku *HR Digital Product Specialist*, dengan dukungan dari Bapak Thimotius Terry Very Erwinta yang menjabat sebagai *HR Digital Product Analyst*. Sementara itu, seksi *Software Engineer* bertugas dalam proses pengembangan, implementasi, serta pemeliharaan sistem. Seksi ini dikoordinasikan oleh Bapak Wisnu Joyo Putro sebagai *Senior Software Engineer* dan didukung oleh dua *Software Engineer*, yaitu Bapak Ivan Yudi Prabowo dan Ibu Erika Putrim. Untuk menjaga layanan teknis tetap optimal, Departemen HRIS juga didukung oleh Bapak



Gambar 3.1. Struktur Organisasi dan Kedudukan dalam Departemen HRIS

Sonny Mukti Prakoso yang bertugas sebagai *Helpdesk Technical Support*.

Sebagai *Mobile Developer Intern*, penulis ditempatkan pada seksi *Software Engineer* dan bekerja langsung di bawah koordinasi tim pengembang. Komunikasi harian dilakukan melalui *tools* kolaborasi seperti Microsoft Teams, sedangkan koordinasi teknis dilaksanakan melalui pertemuan *sprint* yang diadakan secara rutin. Mekanisme kerja mengikuti metodologi *Agile* dengan siklus *sprint* selama dua minggu, di mana setiap akhir *sprint* diadakan *review meeting* untuk membahas progres, hambatan, dan prioritas tugas selanjutnya.

3.2 Tugas yang Dilakukan

Selama menjalani program magang, penulis bertanggung jawab mengembangkan modul *Time Management* pada aplikasi HR Mobile Portal yang dibangun menggunakan *framework* React Native dengan Expo. Proyek ini merupakan bagian dari inisiatif migrasi aplikasi HR Portal dari *framework* lama berbasis PhoneGap ke teknologi yang lebih modern. Fokus pengembangan diarahkan pada modul-modul pengelolaan waktu kerja karyawan—fitur yang paling intensif diakses dalam keseharian operasional.

Modul *Time Management* yang dikembangkan mencakup delapan sub-

modul utama dengan tingkat kompleksitas yang bervariasi. Pertama, modul *Leave* (Cuti) yang merupakan fitur paling lengkap dengan total 35 *file*, mencakup fungsionalitas pengajuan cuti baru, persetujuan cuti untuk atasan, riwayat cuti, serta informasi saldo cuti. Kedua, modul *Absences* (Ketidakhadiran) dengan 29 *file* yang menangani pengajuan izin tidak masuk kerja, persetujuan, riwayat, serta riwayat pembatalan. Ketiga, modul *Attendance* (Kehadiran) dengan 23 *file* yang berfungsi untuk koreksi data kehadiran ketika terjadi ketidaksesuaian pencatatan.

Keempat, modul *Substitution* (Substitusi) dengan 23 *file* yang memungkinkan karyawan mengajukan penggantian jadwal kerja dan mendelegasikan tugas kepada rekan kerja. Kelima, modul SKKL (*Surat Keterangan Kerja Lembur*) dengan 6 *file* yang mengelola pembuatan dan rekapitulasi surat lembur. Keenam, modul *Overtime* (Lembur) dengan 3 *file* untuk menampilkan data lembur karyawan. Ketujuh, modul *Meal Allowance* (Tunjangan Makan) dengan 3 *file* untuk informasi tunjangan makan. Kedelapan, modul *Time Event* dengan 2 *file* untuk menampilkan catatan waktu kerja.

Arsitektur pengembangan mengikuti pola terstruktur dengan pemisahan antara lapisan *services* untuk integrasi API, *hooks* untuk manajemen *state* menggunakan TanStack Query, *components* untuk elemen antarmuka yang *reusable*, dan *types* untuk definisi tipe data menggunakan TypeScript. Tantangan unik muncul karena *backend* yang sudah ada menggunakan PHP dan mengembalikan respons dalam format HTML—bukan JSON yang lazim dijumpai dalam pengembangan *mobile* modern. Untuk mengatasi hal ini, dikembangkan *parser functions* menggunakan library *node-html-parser* guna mengekstrak data dari elemen HTML dan mengonversinya menjadi objek TypeScript yang terstruktur.

Selain pengembangan modul *Time Management*, tugas juga mencakup implementasi sistem autentikasi (*Login*, *Register*, *Forgot Password*), pengembangan halaman *Home* sebagai *dashboard* utama, serta halaman *Profile* yang menampilkan informasi pengguna dan pengaturan *Superior Settings*. Navigasi aplikasi menggunakan pola *Bottom Navigation* dengan tiga tab utama: *Home*, *Time Management*, dan *Profile*.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang diuraikan dalam beberapa tahap, yaitu proses pelaksanaan, perancangan sistem, dan tampilan antarmuka aplikasi.

3.3.1 Proses Pelaksanaan

Minggu-minggu awal difokuskan pada pengenalan lingkungan kerja, pemahaman alur kerja tim, serta teknologi yang digunakan dalam pengembangan proyek. Proses pembelajaran dimulai dengan mempelajari *framework* React Native dan Expo sebagai teknologi utama pengembangan aplikasi *mobile*. Pendalaman juga dilakukan terhadap NativeWind sebagai *library* untuk *styling* menggunakan Tailwind CSS pada React Native, serta TanStack Query untuk manajemen *state* dan *caching* data dari server.

Setelah memahami *tech stack* yang digunakan, pengembangan dimulai secara bertahap dari modul autentikasi hingga modul-modul *Time Management* yang lebih kompleks. Setiap minggu diadakan pertemuan *Progress Report Meeting* pada hari Jumat pukul 14.00 WIB untuk membahas capaian, kendala, dan rencana kerja minggu berikutnya. Mekanisme ini memastikan pengembangan tetap sejalan dengan kebutuhan bisnis dan standar kualitas yang ditetapkan oleh tim.

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke-	Pekerjaan yang Dilakukan
1, 2	Orientasi lingkungan kerja, pemahaman arsitektur sistem HR Portal yang sudah ada, serta pengenalan <i>tech stack</i> React Native, Expo, NativeWind, dan TanStack Query. Mempelajari pola integrasi API dengan <i>backend</i> PHP yang mengembalikan respons HTML. Pengembangan modul autentikasi meliputi halaman <i>Login</i> , <i>Register</i> , dan <i>Forgot Password</i> . Implementasi <i>AuthGuard</i> untuk proteksi <i>route</i> dan penyimpanan token menggunakan Expo SecureStore.
3, 4	Pengembangan halaman <i>Home</i> sebagai <i>dashboard</i> utama dengan integrasi <i>slideshow</i> dan data ringkasan karyawan. Pengembangan halaman utama modul <i>Time Management</i> dengan tampilan menu dan sub-menu untuk akses ke fitur-fitur pengelolaan waktu kerja. Implementasi <i>Bottom Navigation</i> dengan tiga tab utama.
Lanjut pada halaman berikutnya	

Tabel 3.1 – Pekerjaan yang dilakukan tiap minggu (lanjutan)

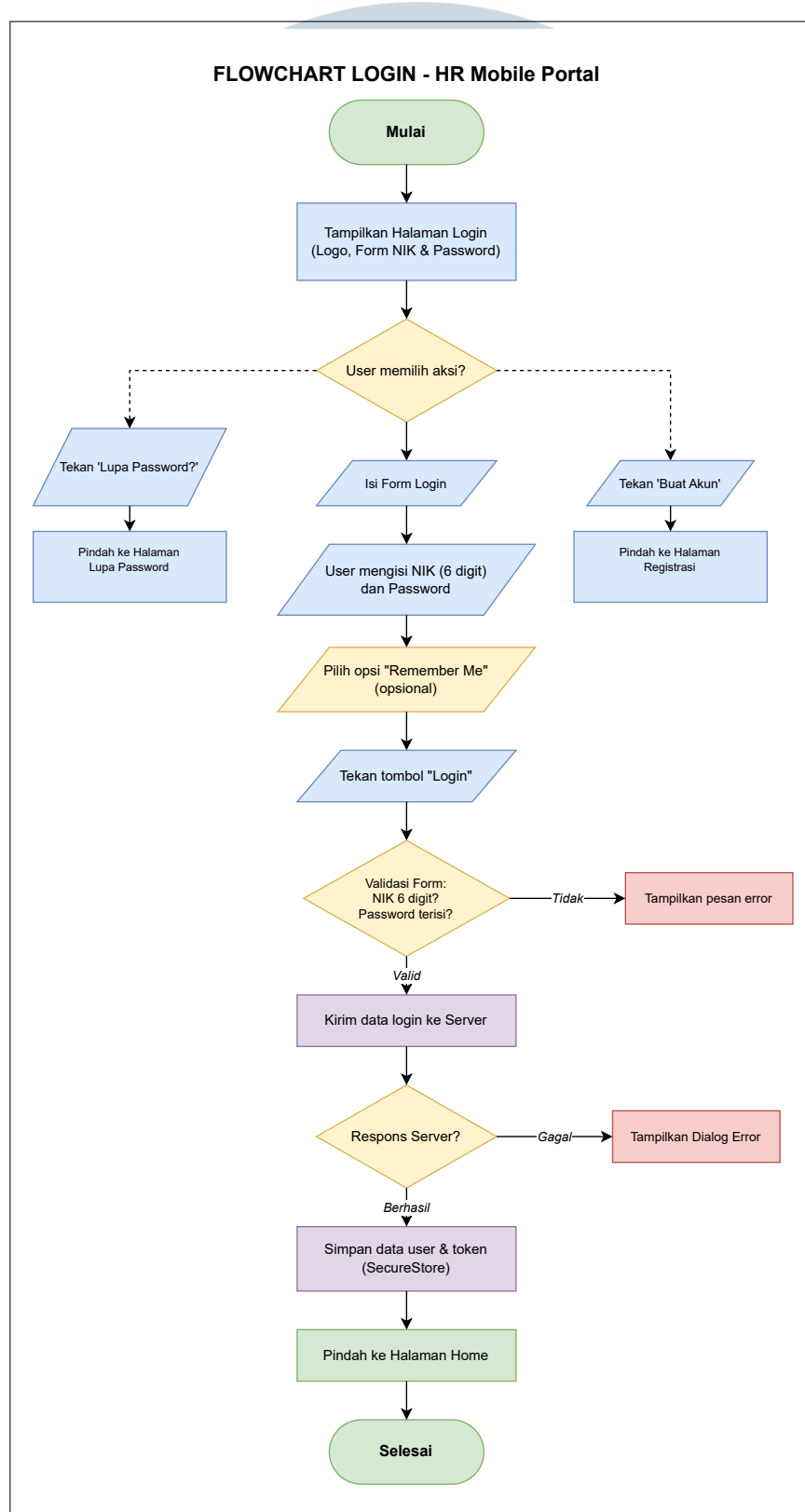
Minggu Ke-	Pekerjaan yang Dilakukan
5, 6	Pengembangan halaman <i>Profile</i> dengan informasi pengguna dan implementasi fitur <i>Superior Settings</i> untuk pengaturan atasan pengganti dan <i>bypass</i> . Pengembangan halaman <i>Personal Information</i> untuk menampilkan data personal karyawan secara detail.
7, 8	Pengembangan modul <i>Time Event</i> untuk menampilkan catatan waktu kerja seperti jam masuk dan pulang. Pengembangan modul <i>Meal Allowance</i> (Tunjangan Makan) untuk menampilkan informasi tunjangan makan dengan detail per hari. Implementasi <i>filter</i> berdasarkan periode.
9, 10	Pengembangan modul <i>Overtime</i> (Lembur) untuk menampilkan data lembur karyawan berdasarkan periode. Pengembangan modul <i>Substitution</i> (Substitusi) mencakup pengajuan penggantian jadwal, konfirmasi dari penerima substitusi, dan persetujuan atasan. Pembuatan komponen <i>reusable</i> untuk form substitusi.
11, 12	Pengembangan modul <i>Leave</i> (Cuti) mencakup fitur pengajuan cuti baru, tampilan saldo cuti, dan riwayat pengajuan. Pengembangan fitur persetujuan cuti untuk atasan dan riwayat pembatalan cuti. Implementasi <i>dialog</i> detail dan mekanisme <i>approval/rejection</i> dengan validasi form. Pembuatan <i>parser</i> untuk mengekstrak data dari respons HTML <i>backend</i> .
13	Pengembangan modul <i>Absences</i> (Ketidakhadiran) dengan fitur pengajuan, persetujuan, riwayat, dan riwayat pembatalan. Pengembangan modul <i>Attendance</i> (Kehadiran) untuk koreksi data kehadiran dengan tampilan kalender dan <i>filter</i> tanggal. Pengembangan modul SKKL (<i>Surat Keterangan Kerja Lembur</i>) dengan fitur pembuatan, rekapitulasi, status pengajuan, dan status penugasan. Pengujian integrasi seluruh modul, perbaikan <i>bug</i> , dan optimalisasi performa.

3.3.2 Perancangan Sistem

Perancangan sistem aplikasi HR Mobile Portal ditampilkan dalam bentuk *flowchart* yang menggambarkan alur navigasi dan interaksi pengguna dengan berbagai fitur dalam modul *Time Management*. *Flowchart* berfungsi untuk memvisualisasikan bagaimana pengguna mengakses dan berinteraksi dengan setiap fitur, mulai dari proses autentikasi hingga penggunaan fitur-fitur spesifik dalam modul *Time Management*.

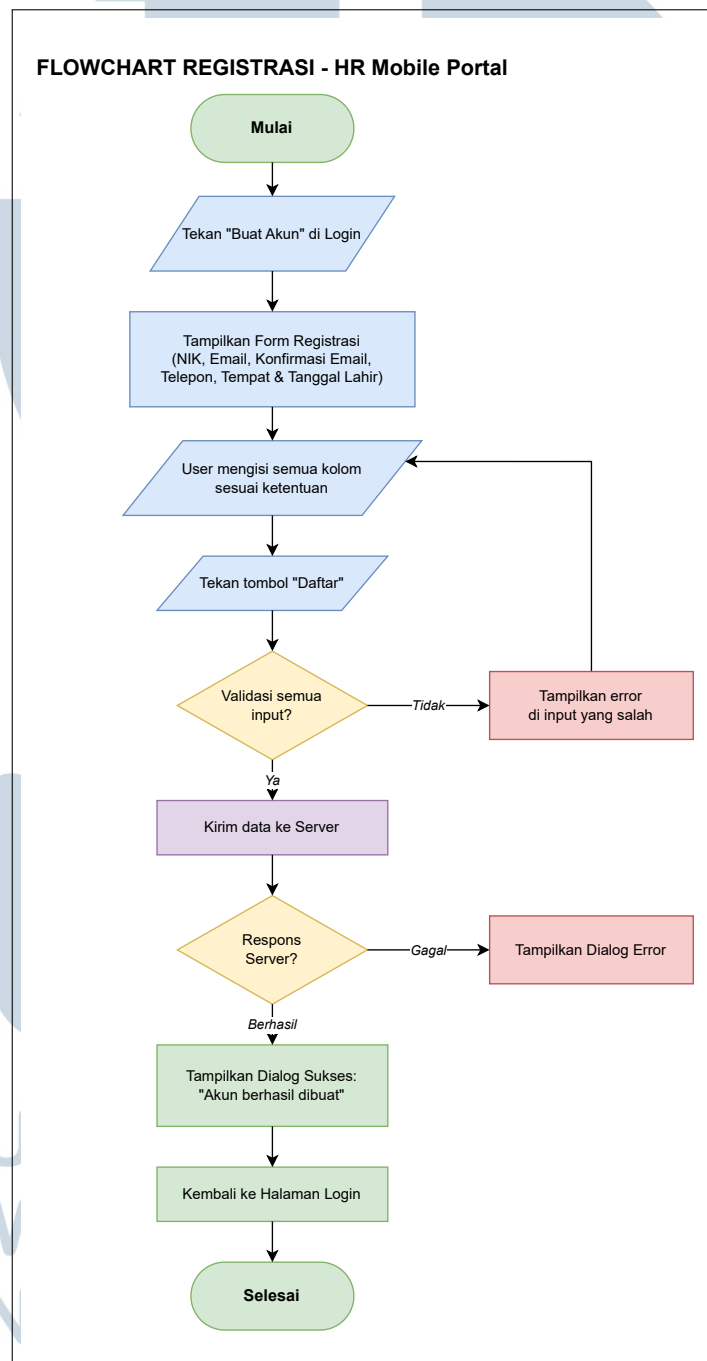


A Flowchart Modul Autentikasi



Gambar 3.2. Flowchart Login

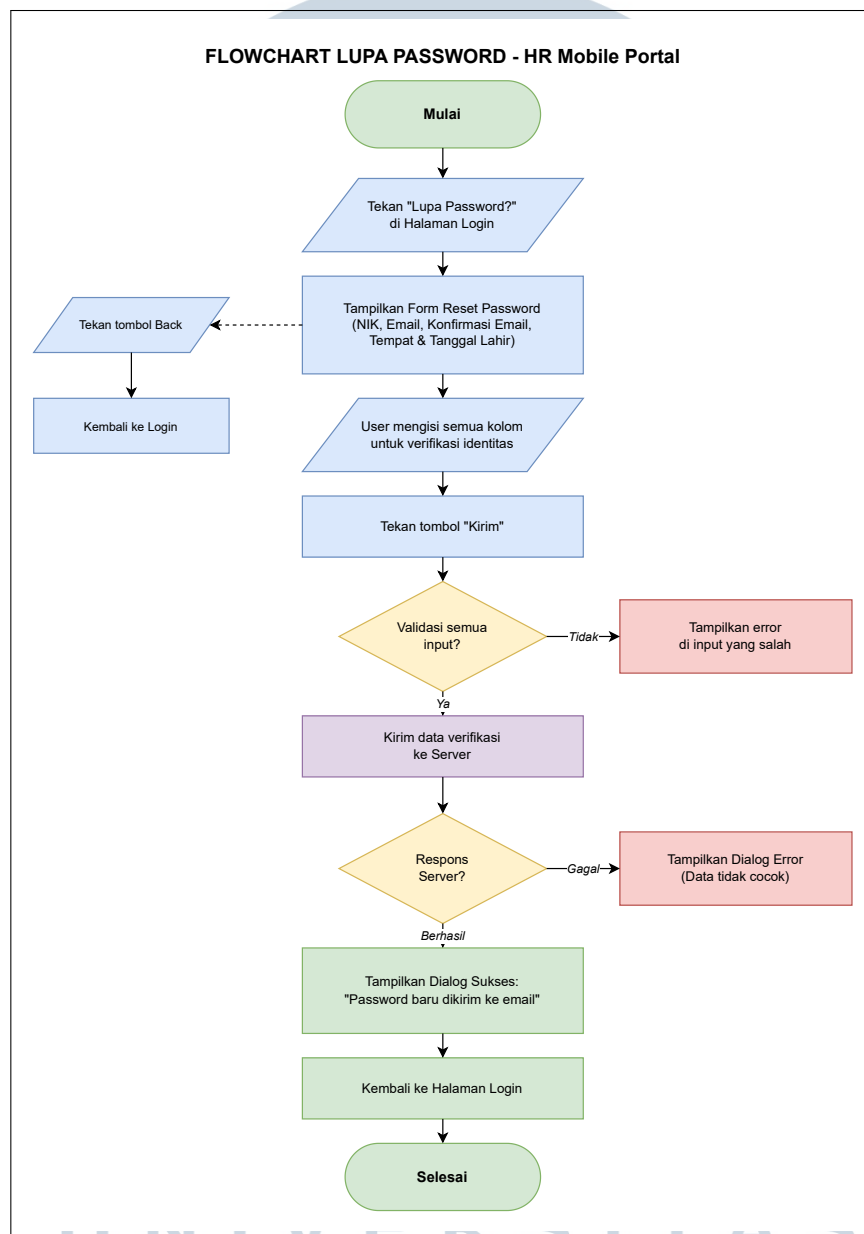
Gambar 3.2 menampilkan alur *Login* yang merupakan proses awal pengguna memasukkan NIK dan *password*. Sistem melakukan validasi kredensial, dan jika berhasil, pengguna diarahkan ke halaman *Home*. Jika gagal, sistem menampilkan pesan kesalahan dan pengguna dapat mencoba kembali.



Gambar 3.3. Flowchart Register

Gambar 3.3 menampilkan alur *Register* untuk pendaftaran pengguna

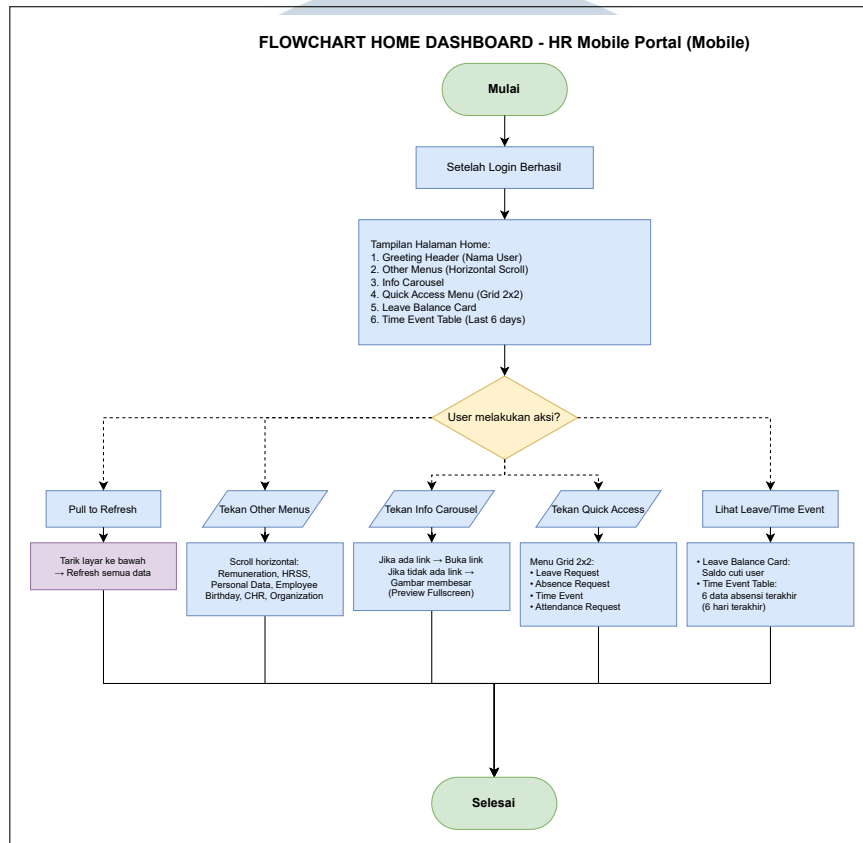
baru. Pengguna mengisi form dengan data NIK, email, dan *password*. Sistem memvalidasi data dan mengirimkan email verifikasi jika pendaftaran berhasil.



Gambar 3.4. Flowchart Forgot Password

Gambar 3.4 menampilkan alur *Forgot Password* untuk pemulihan *password*. Pengguna memasukkan email terdaftar, dan sistem mengirimkan tautan *reset password* ke email tersebut.

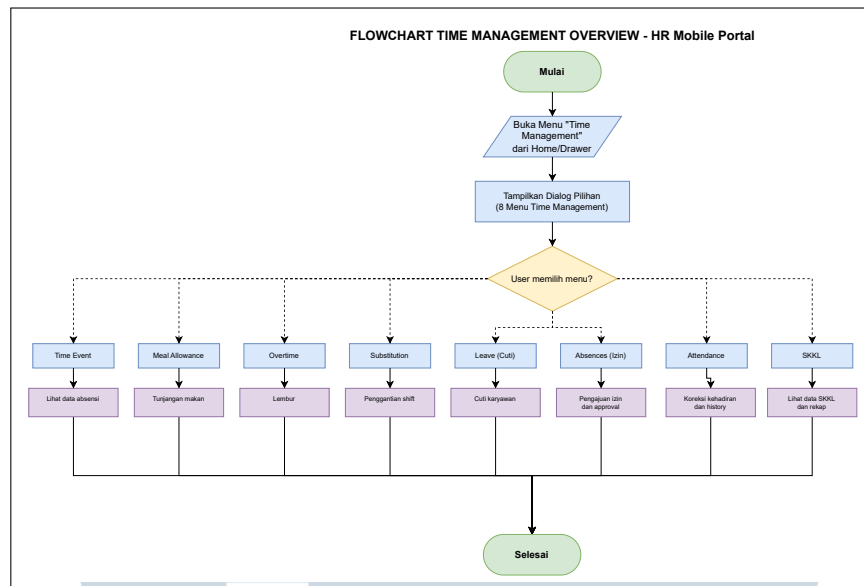
B Flowchart Navigasi Utama dan Profile



Gambar 3.5. Flowchart Home

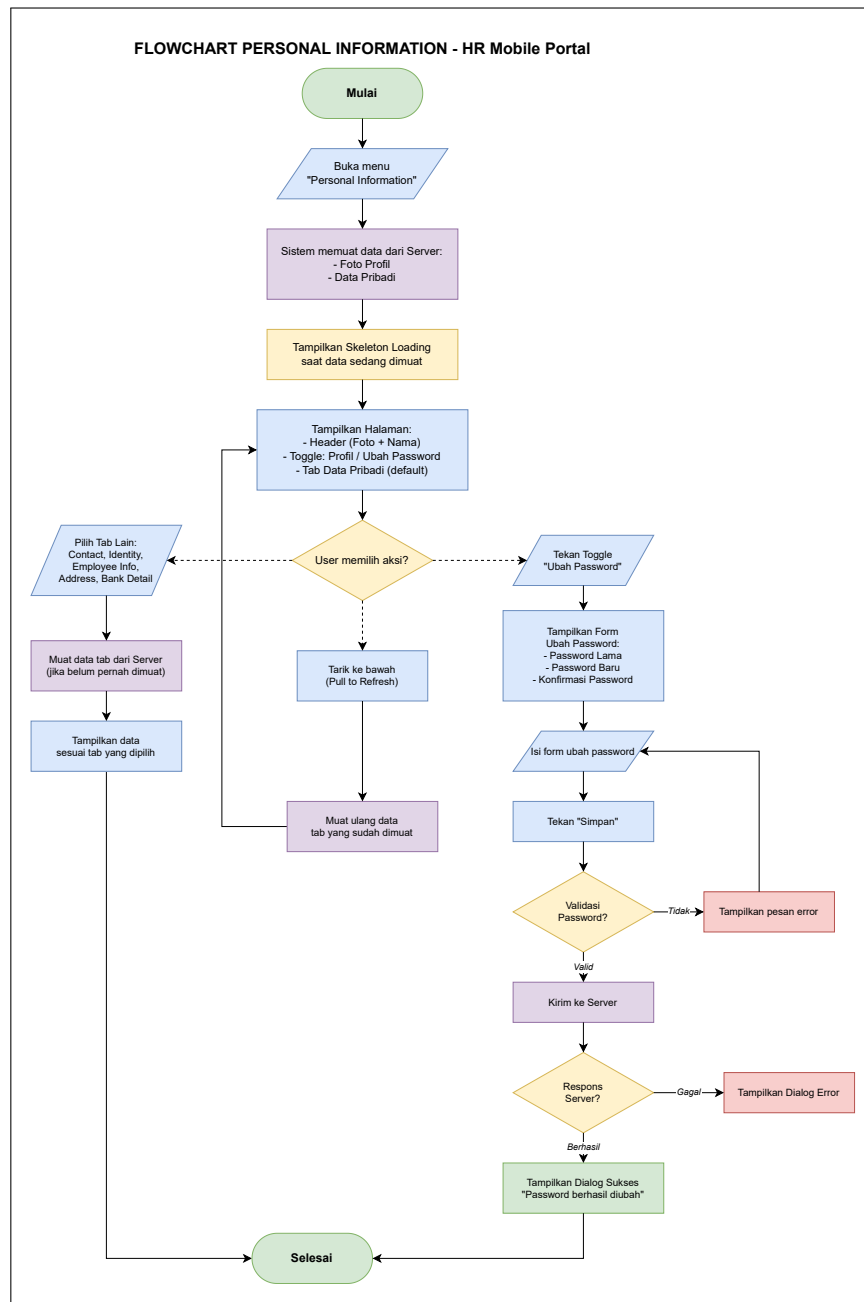
Gambar 3.5 menampilkan alur halaman *Home* yang berfungsi sebagai *dashboard* utama aplikasi. Halaman ini menampilkan *slideshow* informasi, ringkasan data karyawan, dan akses cepat ke fitur-fitur utama melalui *quick access menu*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.6. Flowchart Time Management

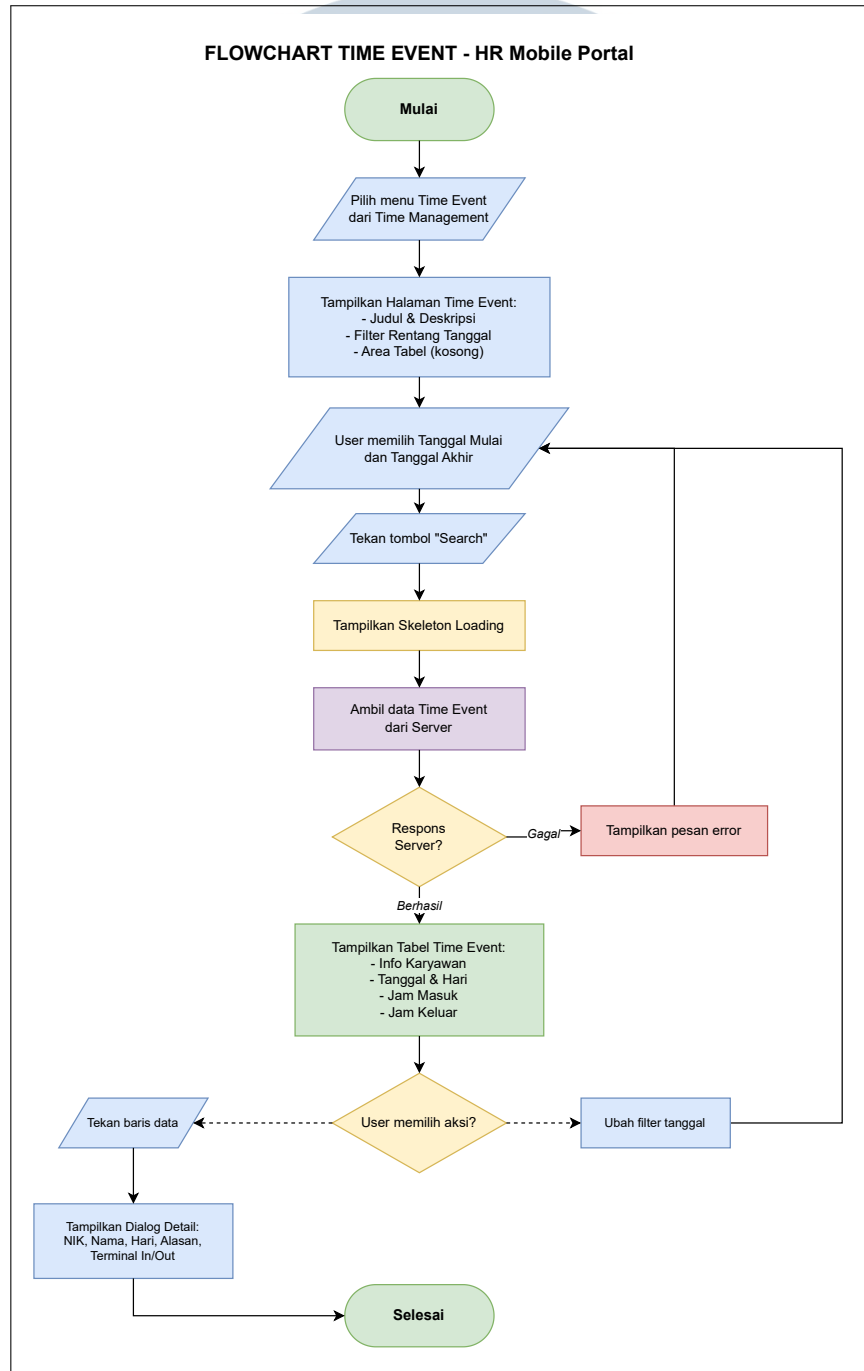
Gambar 3.6 menampilkan alur halaman utama modul *Time Management*. Halaman ini menampilkan daftar menu sub-modul seperti *Leave*, *Absences*, *Attendance*, *Substitution*, *SKKL*, *Overtime*, *Meal Allowance*, dan *Time Event*.



Gambar 3.7. Flowchart Profile

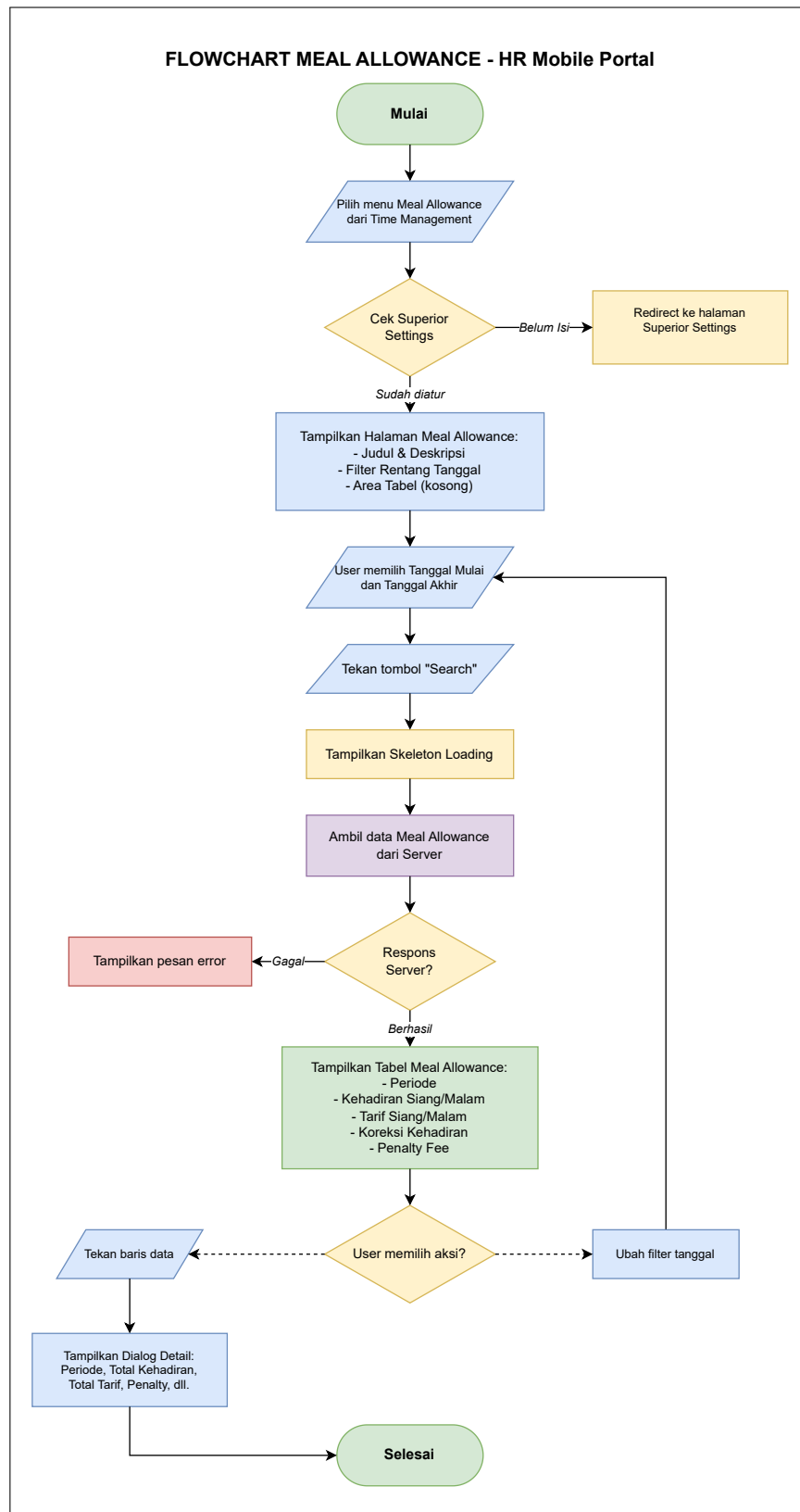
Gambar 3.7 menampilkan alur halaman *Profile* yang menunjukkan informasi profil pengguna. Pengguna dapat mengakses *Superior Settings* untuk pengaturan atasan, mengubah *password*, dan melakukan *logout* dari aplikasi.

C Flowchart Modul Time Event, Meal Allowance, dan Overtime



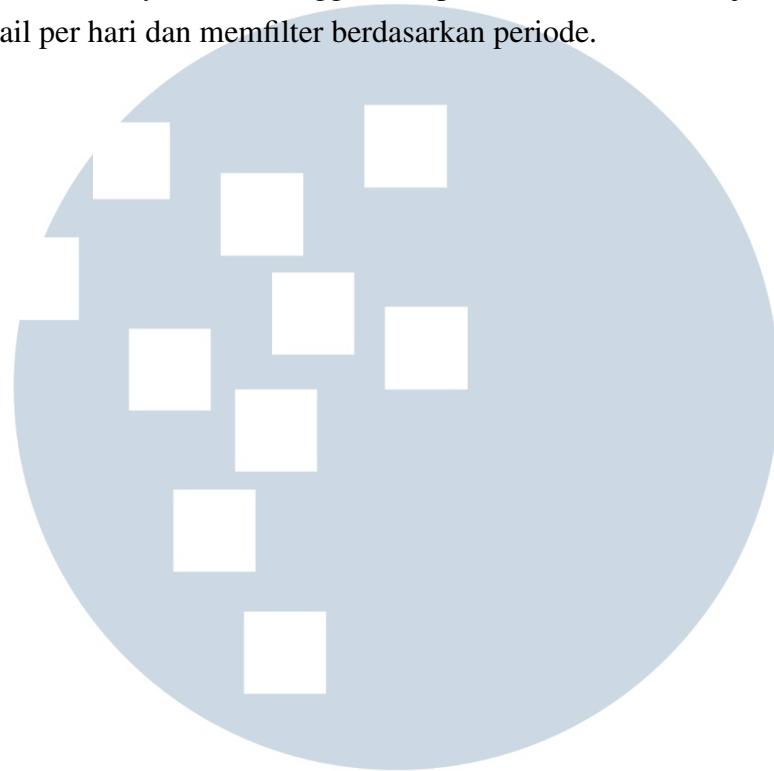
Gambar 3.8. Flowchart Time Event

Gambar 3.8 menampilkan alur modul *Time Event* yang berfungsi menampilkan catatan waktu kerja karyawan. Pengguna dapat melihat jam masuk dan pulang berdasarkan periode yang dipilih dengan fitur *filter* tanggal.

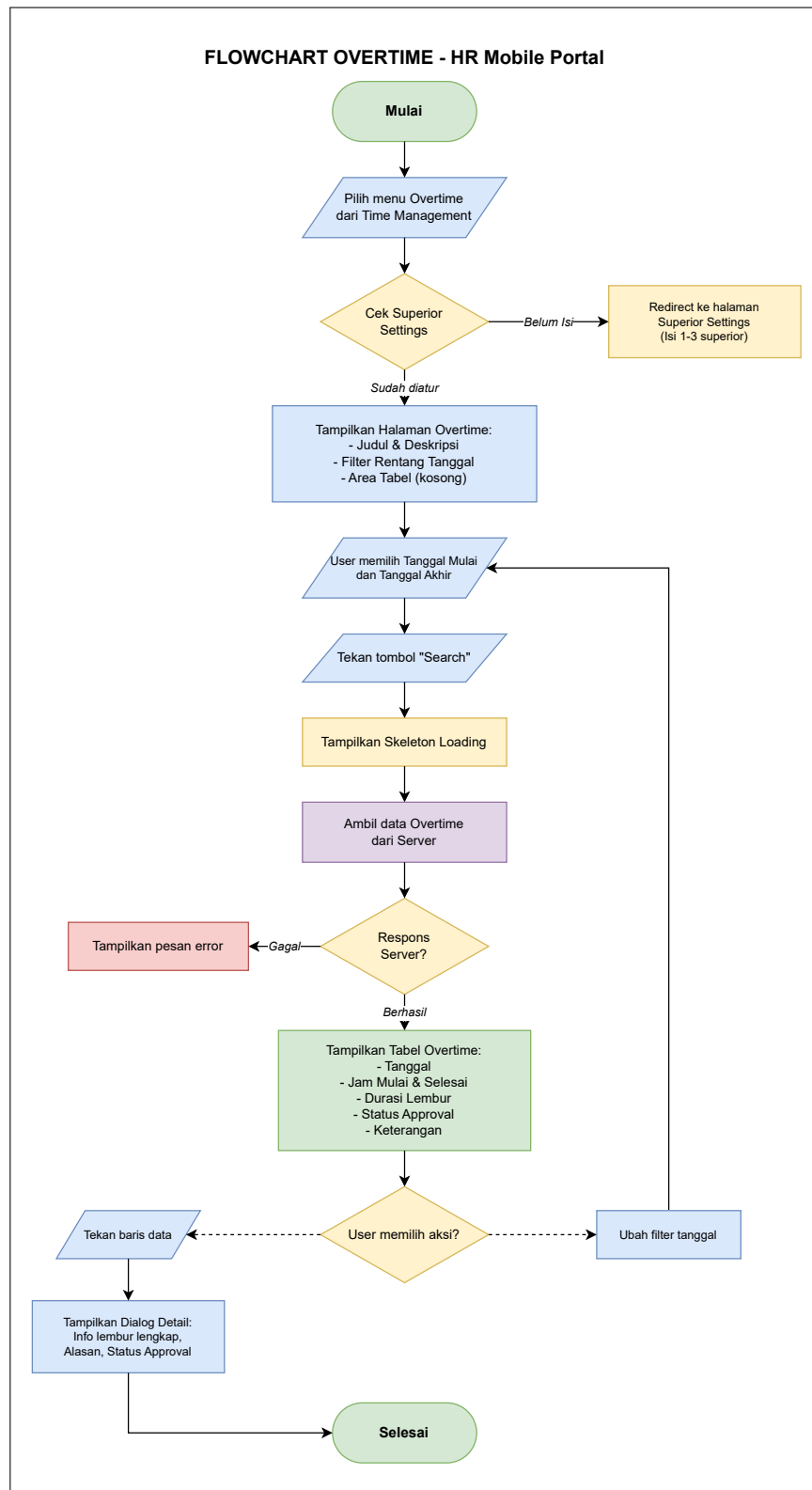


Gambar 3.9. Flowchart Meal Allowance

Gambar 3.9 menampilkan alur modul *Meal Allowance* untuk informasi tunjangan makan karyawan. Pengguna dapat melihat daftar tunjangan makan dengan detail per hari dan memfilter berdasarkan periode.



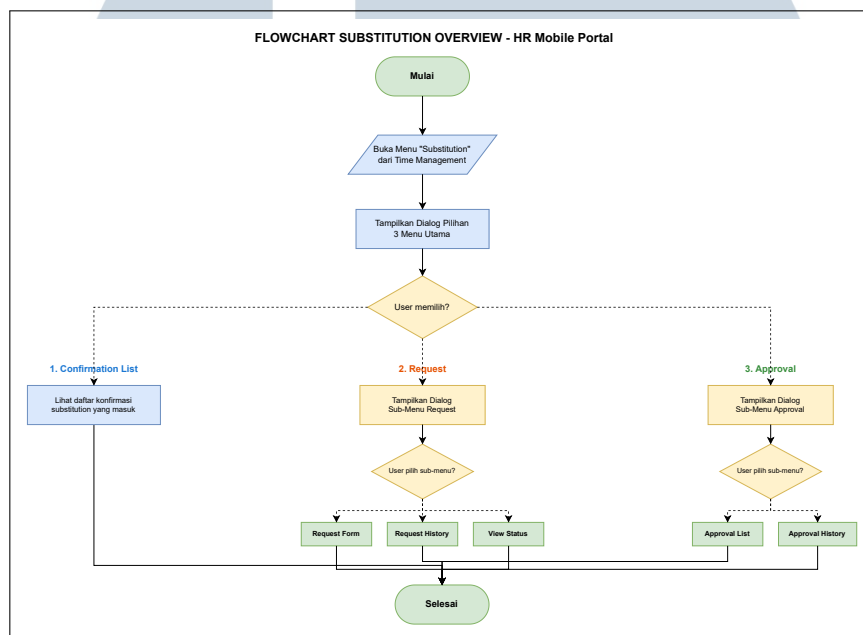
UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.10. Flowchart Overtime

Gambar 3.10 menampilkan alur modul *Overtime* yang menampilkan data lembur karyawan. Pengguna dapat melihat riwayat lembur berdasarkan periode dengan informasi tanggal, durasi, dan status.

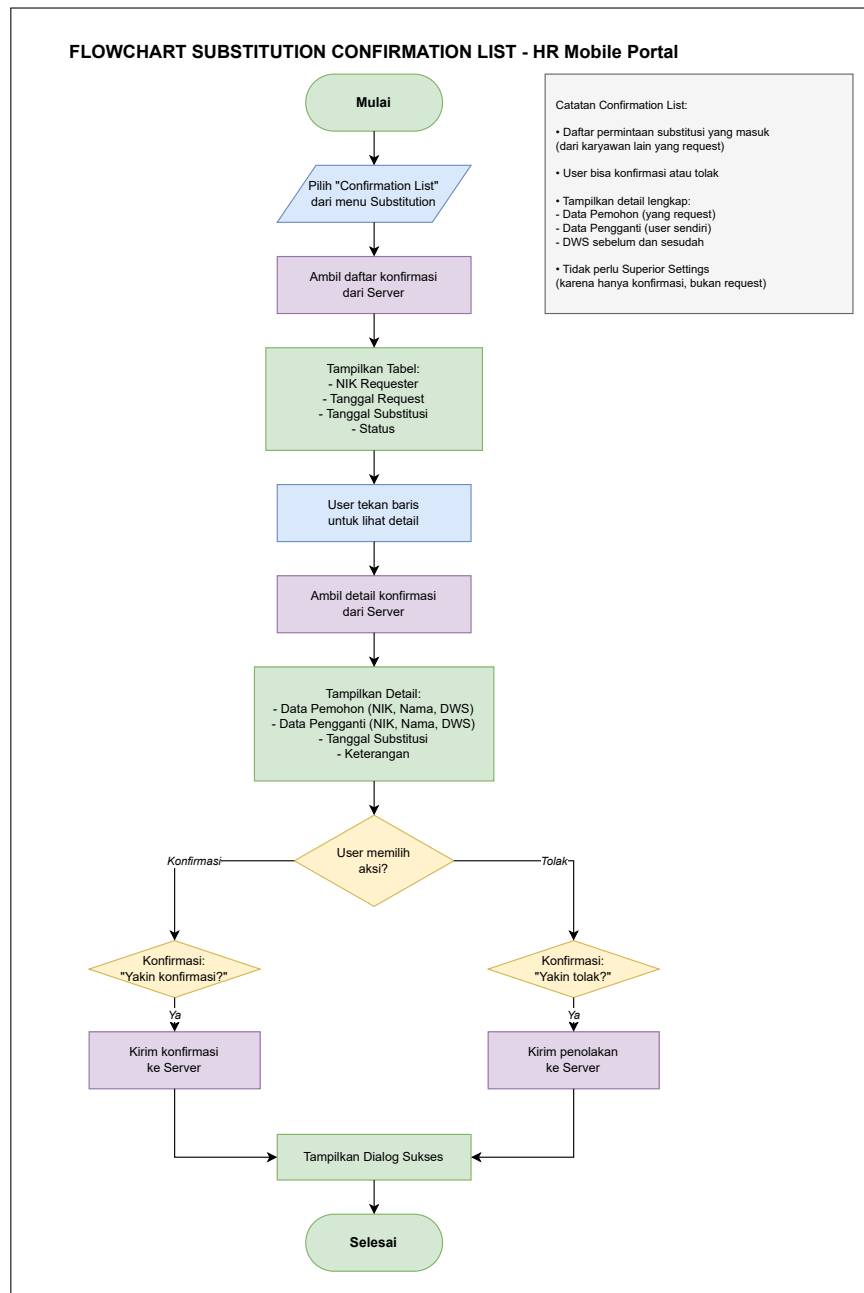
D Flowchart Modul Substitution (Substitusi)



Gambar 3.11. Flowchart Overview Substitution

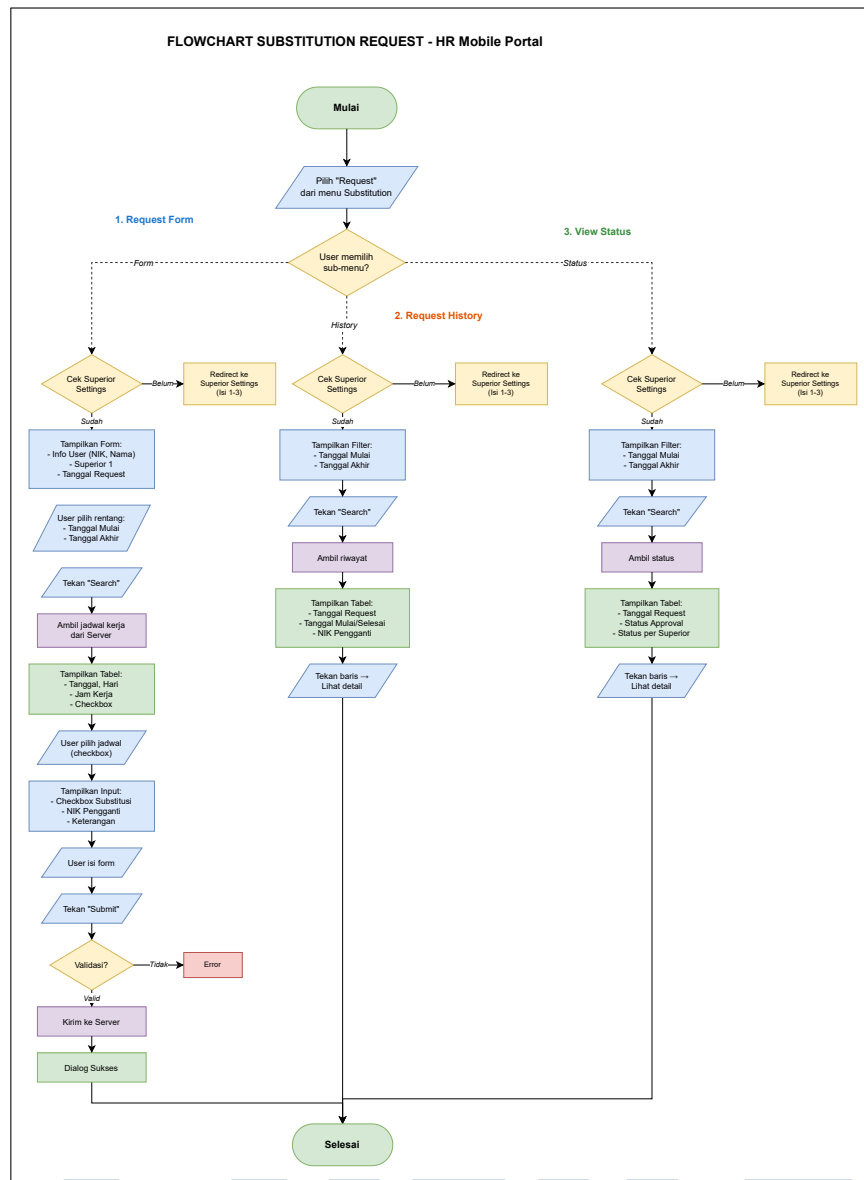
Gambar 3.11 menampilkan struktur menu modul *Substitution* yang terdiri dari empat opsi: *Confirmation List* untuk konfirmasi penerima substitusi, *Request* untuk pengajuan substitusi, *View Status* untuk memantau status pengajuan, dan *Approval* untuk persetujuan atasan.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



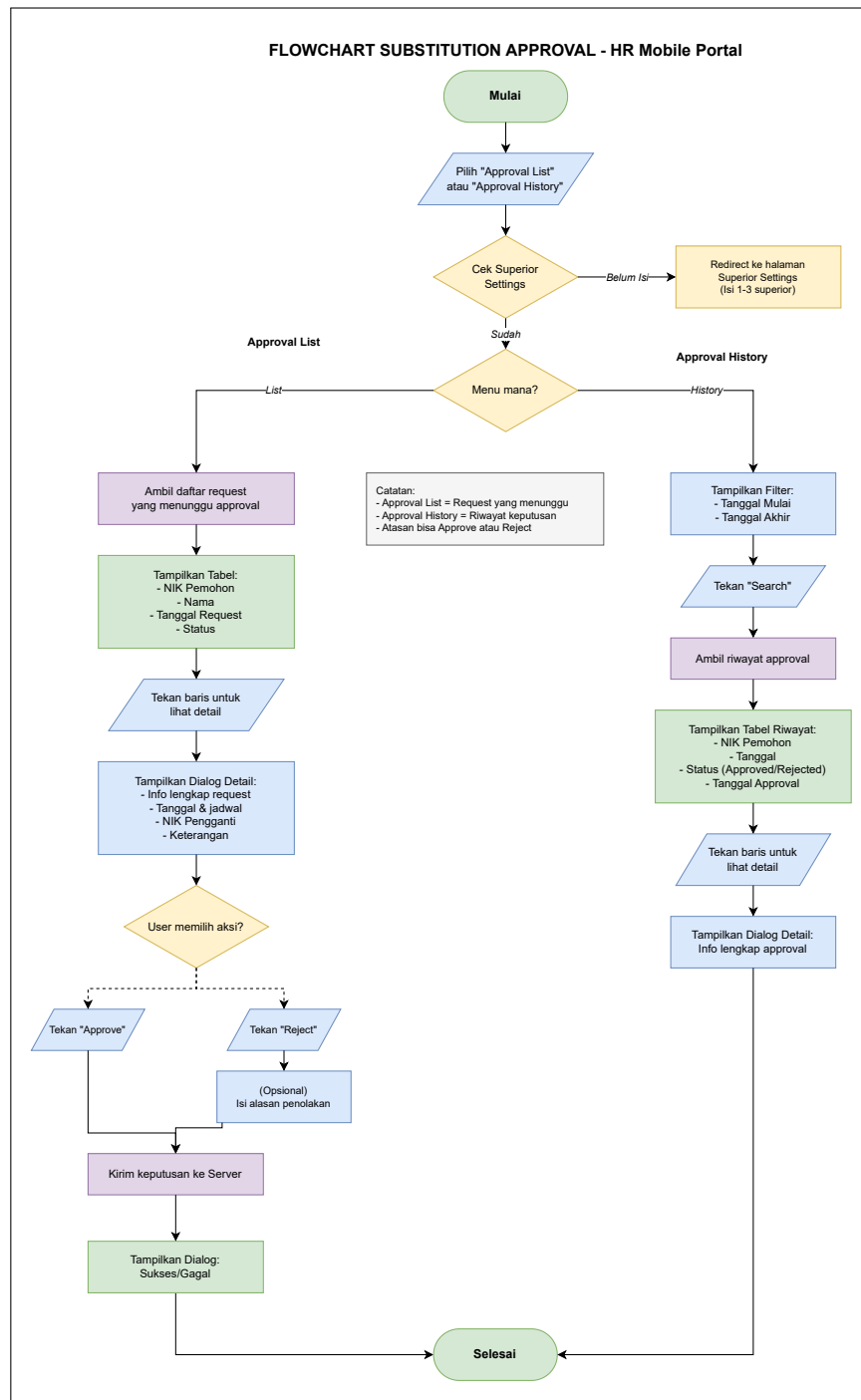
Gambar 3.12. Flowchart Confirmation List Substitution

Gambar 3.12 menampilkan alur *Confirmation List* untuk penerima substitusi. Rekan kerja yang ditunjuk sebagai pengganti dapat melihat daftar permintaan dan mengonfirmasi kesediaannya.



Gambar 3.13. Flowchart Request Substitution

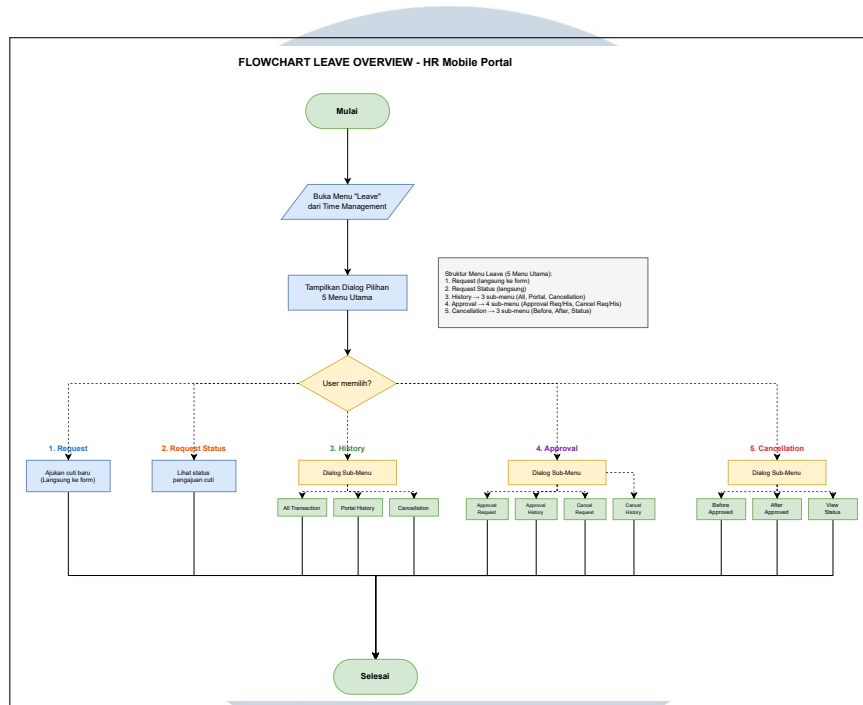
Gambar 3.13 menampilkan alur pengajuan substitusi. Pengguna memilih rekan kerja sebagai pengganti, menentukan periode substitusi, mengisi keterangan, dan mengirimkan pengajuan ke sistem.



Gambar 3.14. Flowchart Approval Substitution

Gambar 3.14 menampilkan alur persetujuan substitusi oleh atasan. Atasan melihat daftar pengajuan, membuka detail, dan memberikan keputusan berupa persetujuan atau penolakan.

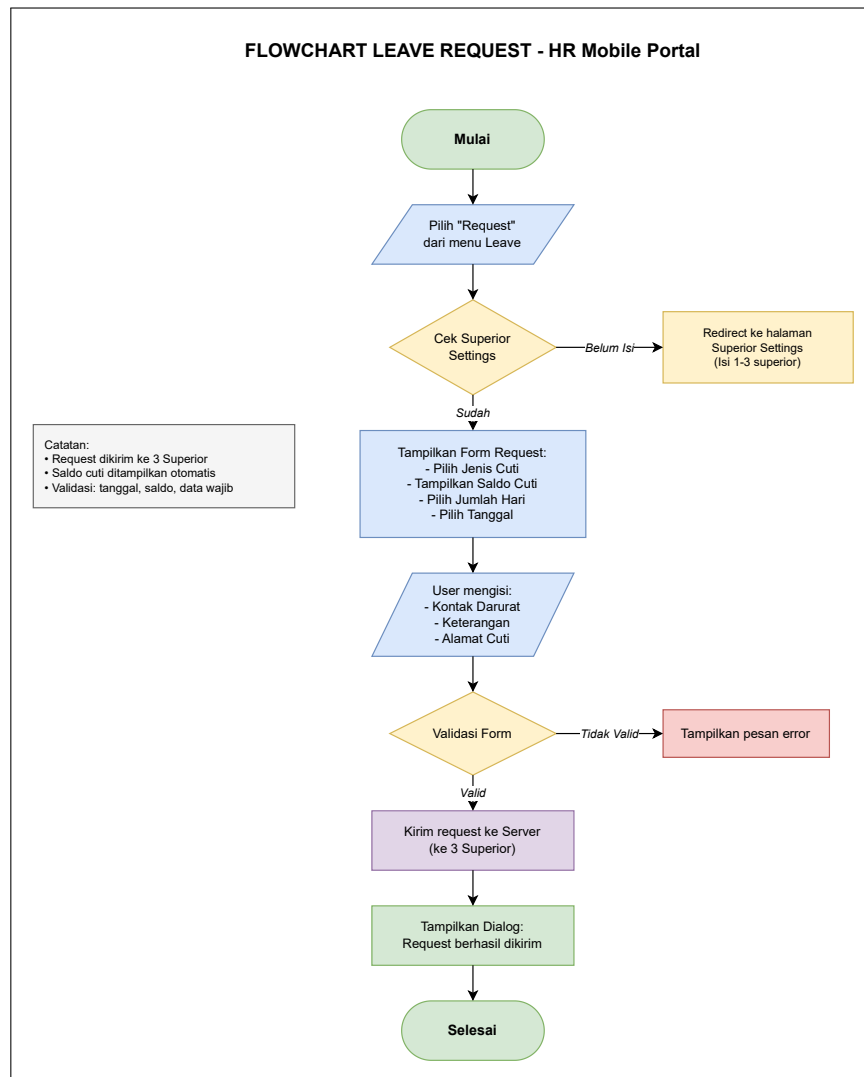
E Flowchart Modul Leave (Cuti)



Gambar 3.15. Flowchart Overview Leave

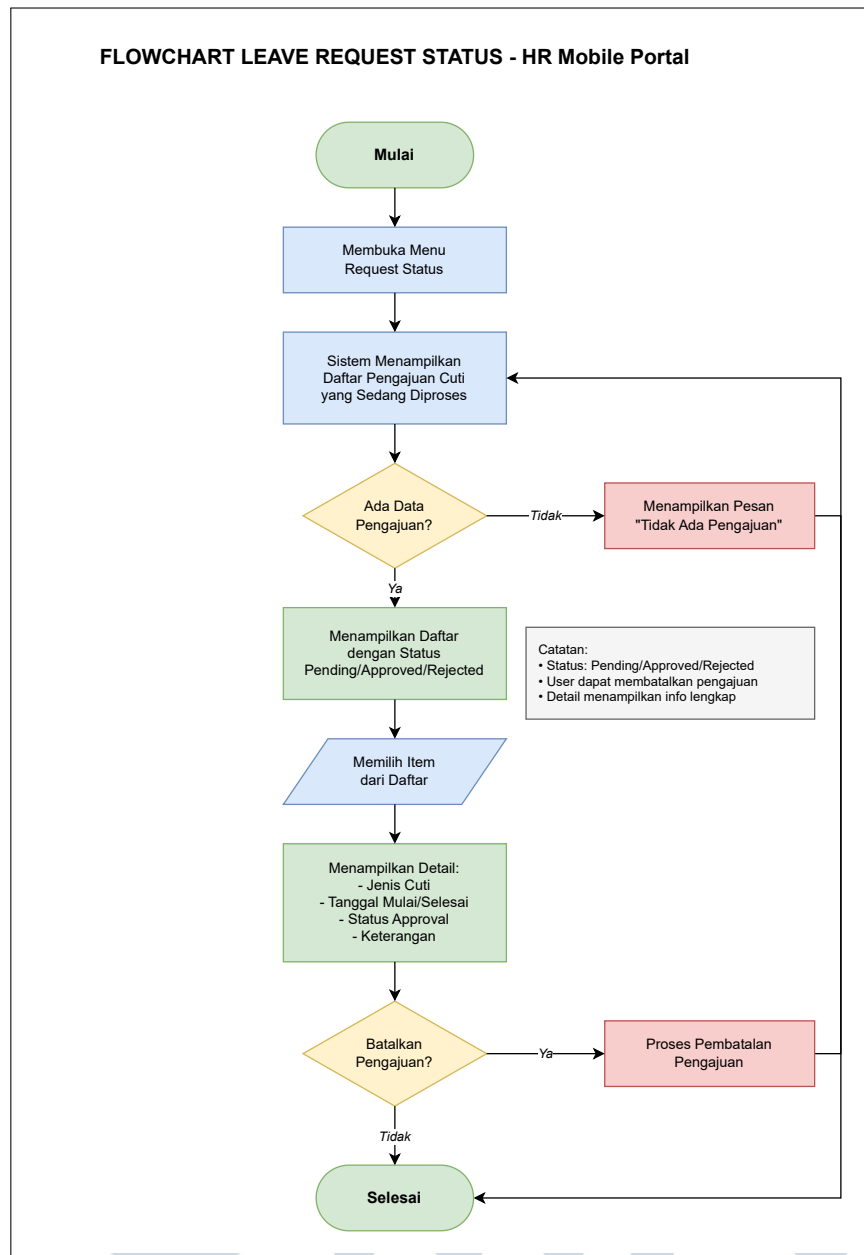
Gambar 3.15 menampilkan struktur menu modul *Leave* yang terdiri dari lima opsi utama: *Request* untuk pengajuan cuti baru, *Request Status* untuk memantau status pengajuan, *Cancellation* untuk pembatalan pengajuan, *History* untuk riwayat cuti, dan *Approval* untuk persetujuan cuti (khusus atasan).

UNIVERSITAS
MULTIMEDIA
NUSANTARA



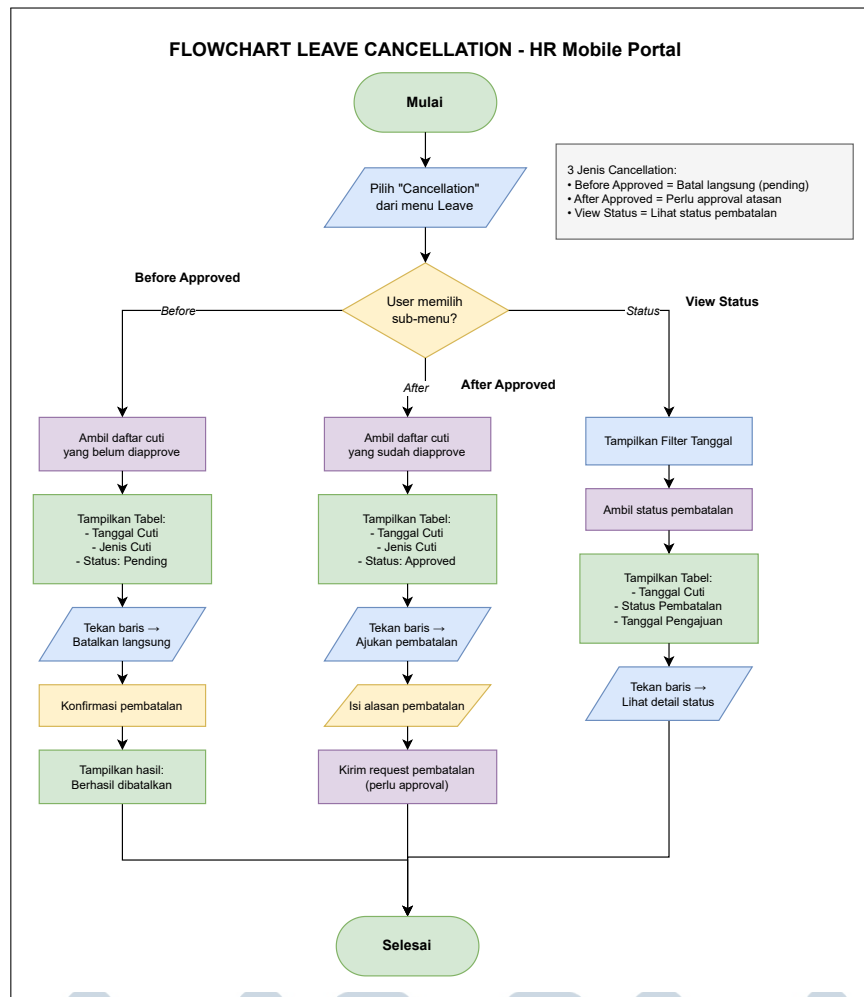
Gambar 3.16. Flowchart Request Leave

Gambar 3.16 menampilkan alur pengajuan cuti. Pengguna memilih jenis cuti, menentukan tanggal mulai dan selesai, mengisi keterangan, dan mengirimkan pengajuan. Sistem memvalidasi data dan meneruskan ke atasan untuk persetujuan.



Gambar 3.17. Flowchart Request Status Leave

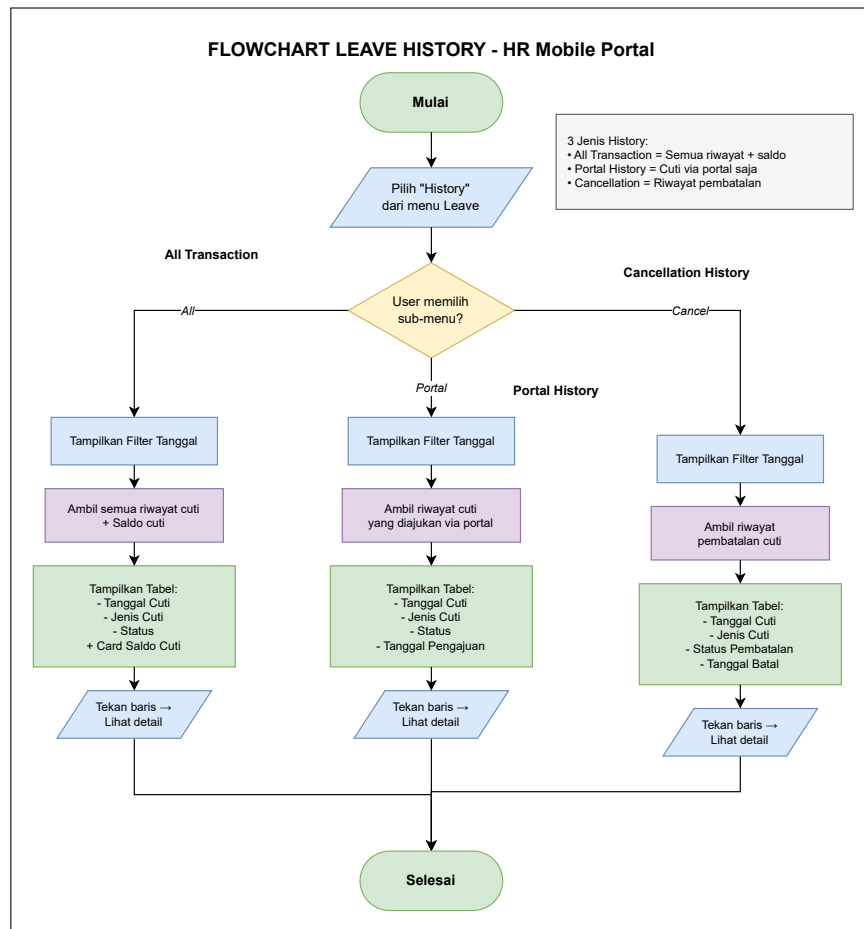
Gambar 3.17 menampilkan alur pemantauan status pengajuan cuti. Pengguna dapat melihat daftar pengajuan yang sedang diproses beserta statusnya (*Pending*, *Approved*, atau *Rejected*), dan membuka detail untuk melihat informasi lengkap pengajuan.



Gambar 3.18. Flowchart Cancellation Leave

Gambar 3.18 menampilkan alur pembatalan cuti. Pengguna dapat membatalkan pengajuan cuti yang masih dalam status *pending* dengan mengisi alasan pembatalan.

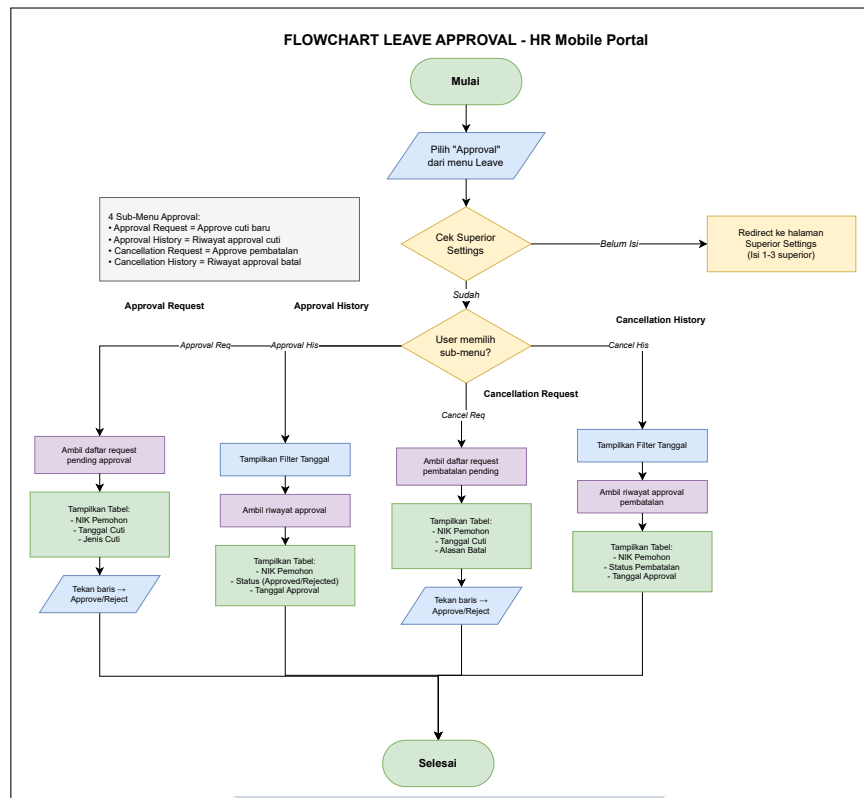
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.19. Flowchart History Leave

Gambar 3.19 menampilkan alur riwayat cuti. Pengguna dapat melihat semua pengajuan cuti yang pernah diajukan dengan fitur *filter* berdasarkan status dan periode.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

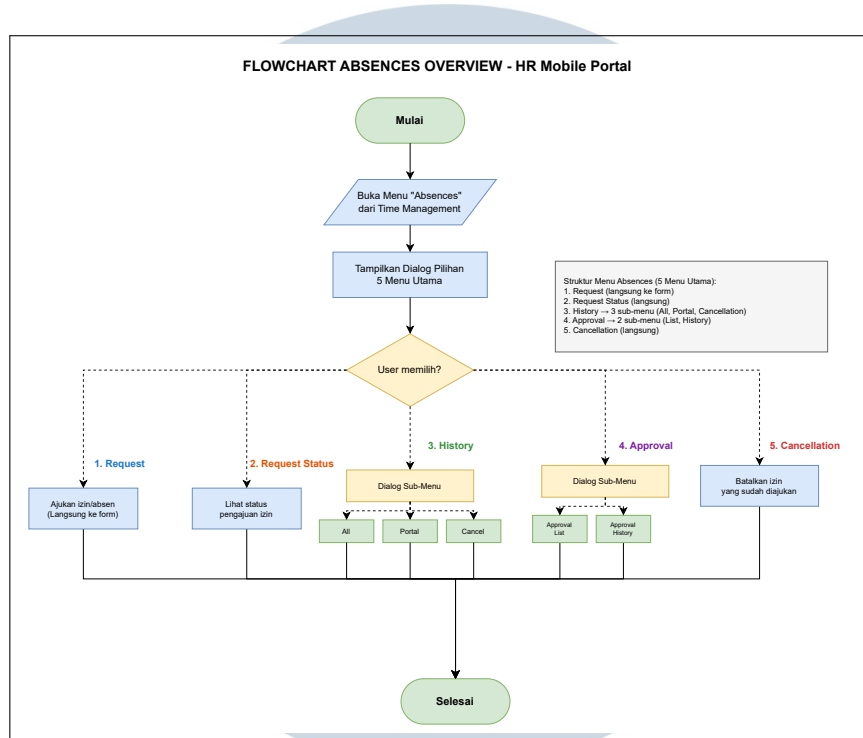


Gambar 3.20. Flowchart Approval Leave

Gambar 3.20 menampilkan alur persetujuan cuti oleh atasan. Atasan melihat daftar pengajuan cuti bawahan, membuka detail pengajuan, dan memberikan keputusan berupa persetujuan atau penolakan beserta keterangan.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

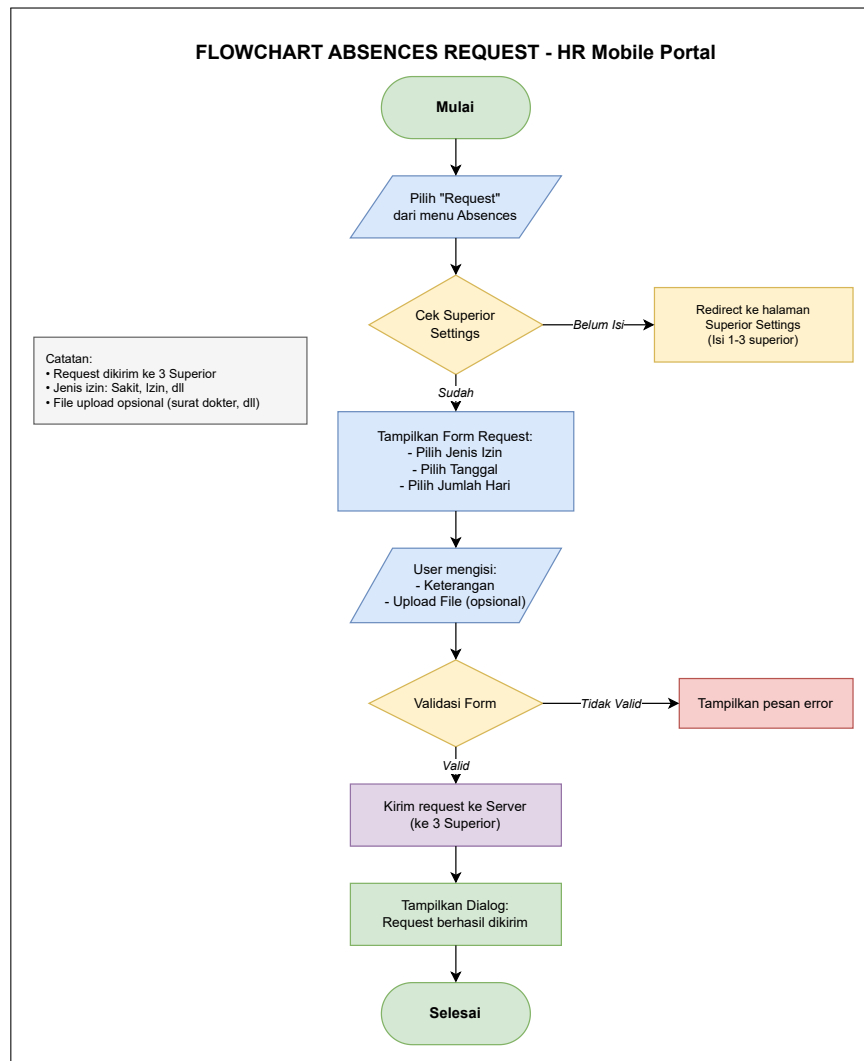
F Flowchart Modul Absences (Ketidakhadiran)



Gambar 3.21. Flowchart Overview Absences

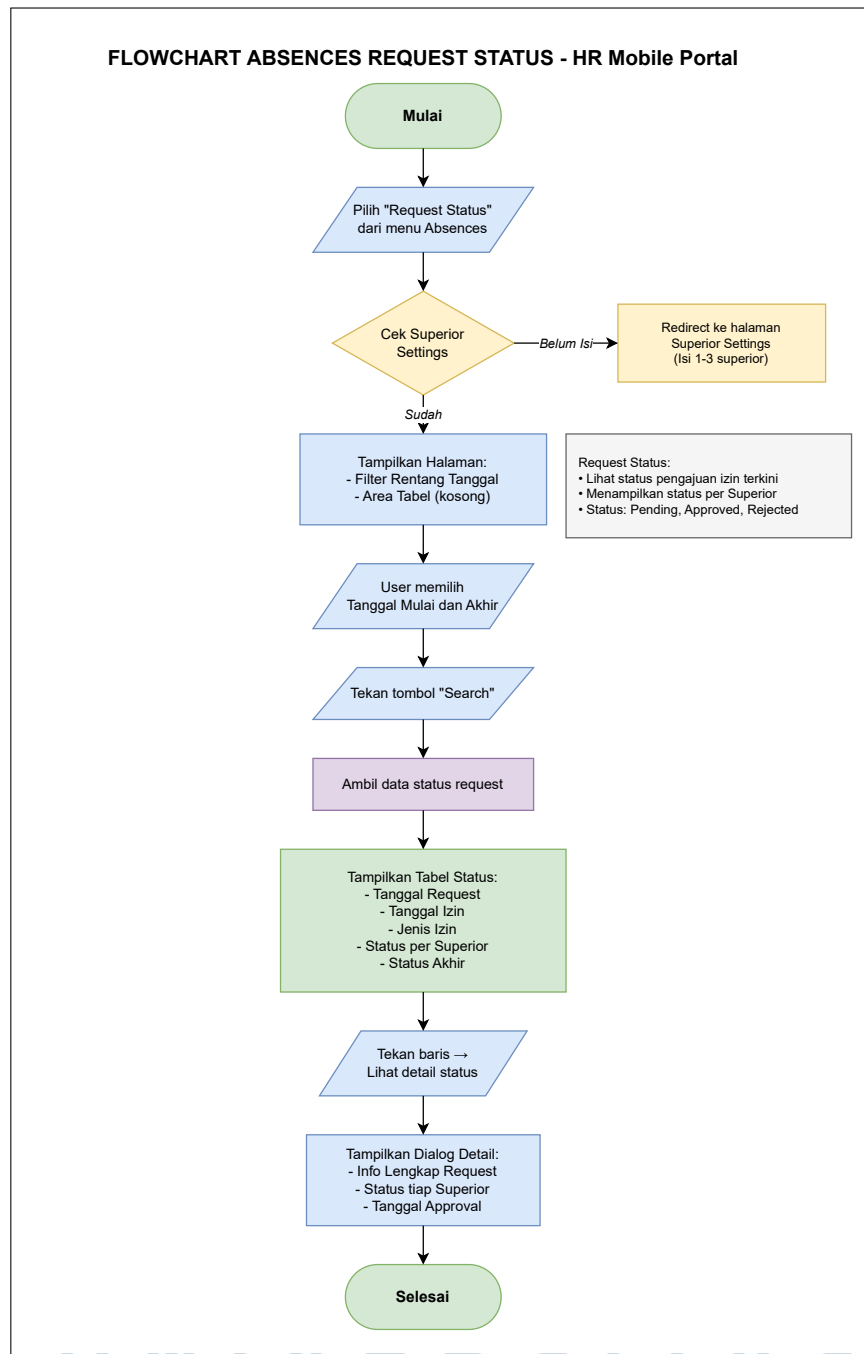
Gambar 3.21 menampilkan struktur menu modul *Absences* yang terdiri dari lima opsi: *Request* untuk mengajukan izin tidak masuk, *Request Status* untuk memantau status pengajuan, *Cancellation* untuk membatalkan pengajuan, *History* untuk melihat riwayat, dan *Approval* untuk atasan menyetujui pengajuan bawahan.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



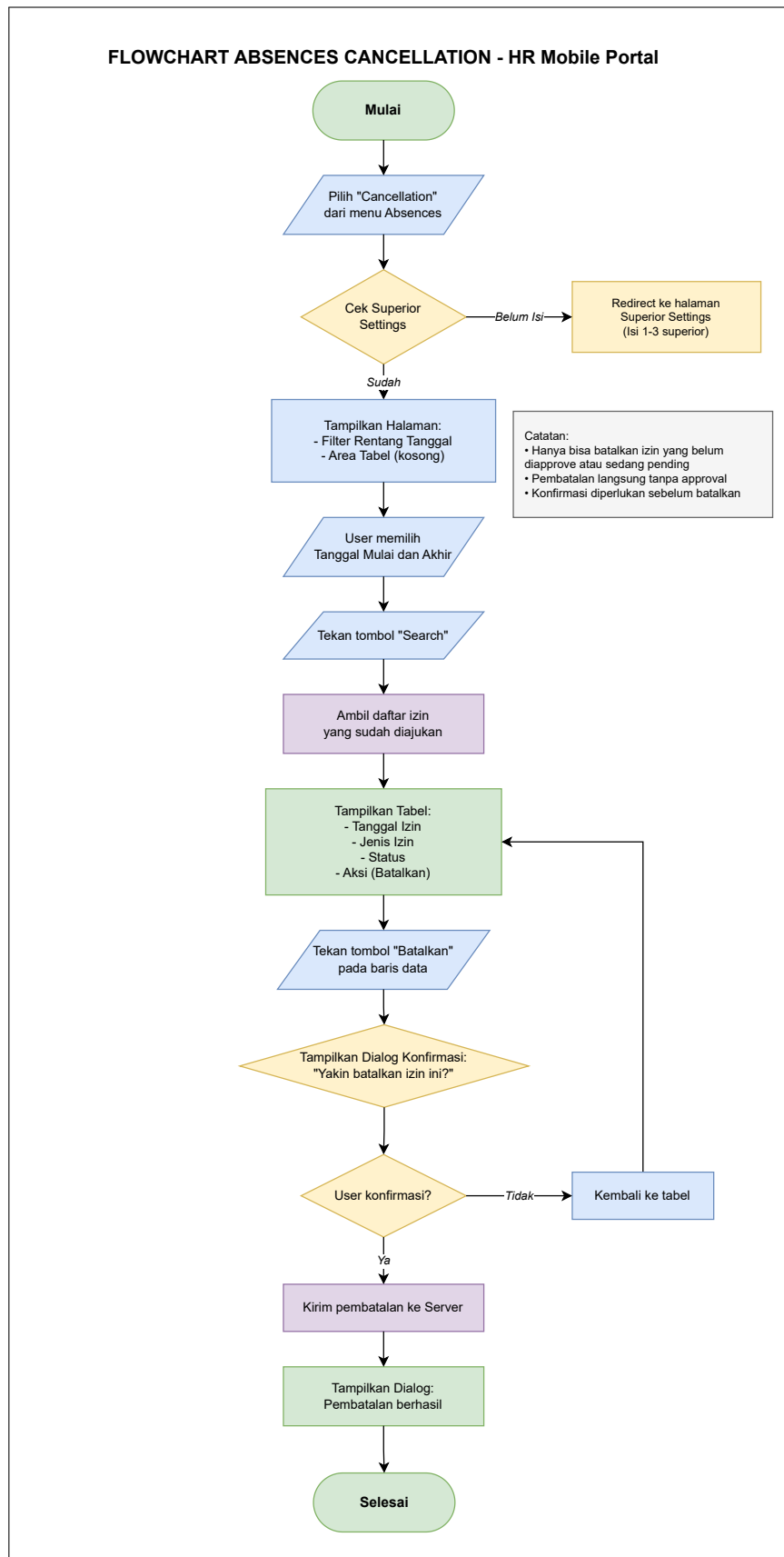
Gambar 3.22. Flowchart Request Absences

Gambar 3.22 menampilkan alur pengajuan ketidakhadiran. Pengguna memilih jenis izin (sakit, keperluan keluarga, dll.), mengisi tanggal dan keterangan, serta mengunggah dokumen pendukung jika diperlukan.



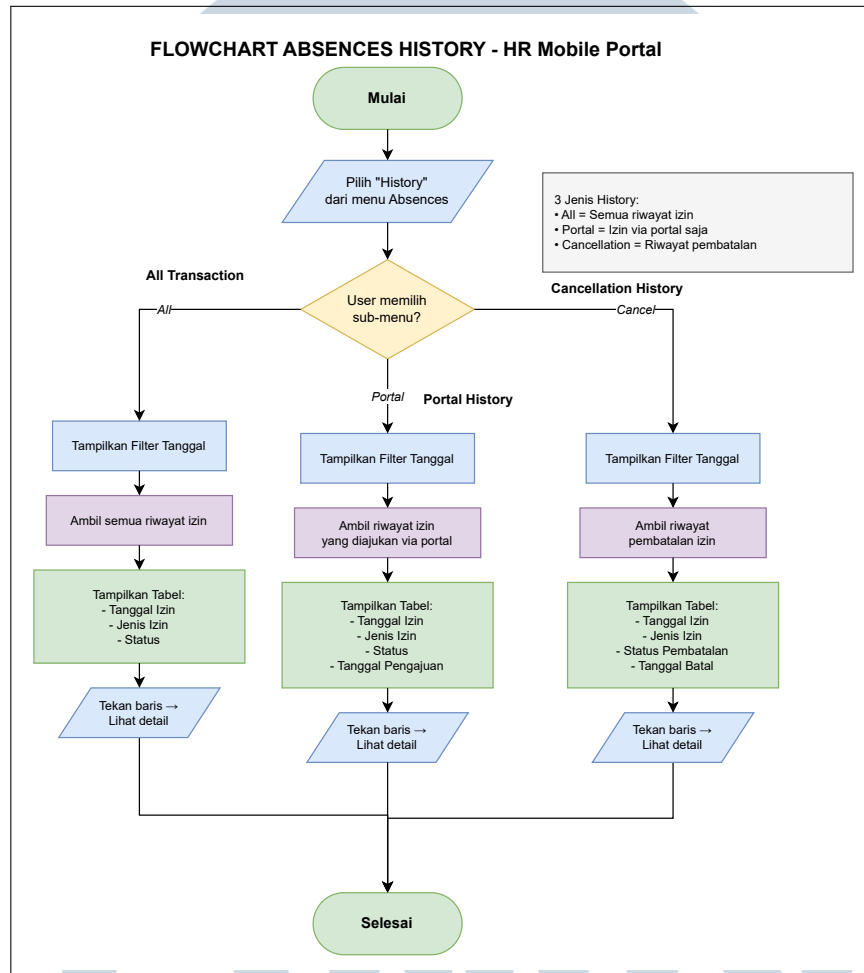
Gambar 3.23. Flowchart Request Status Absences

Gambar 3.23 menampilkan alur pemantauan status pengajuan ketidakhadiran. Pengguna dapat melihat daftar pengajuan yang sedang diproses beserta statusnya.



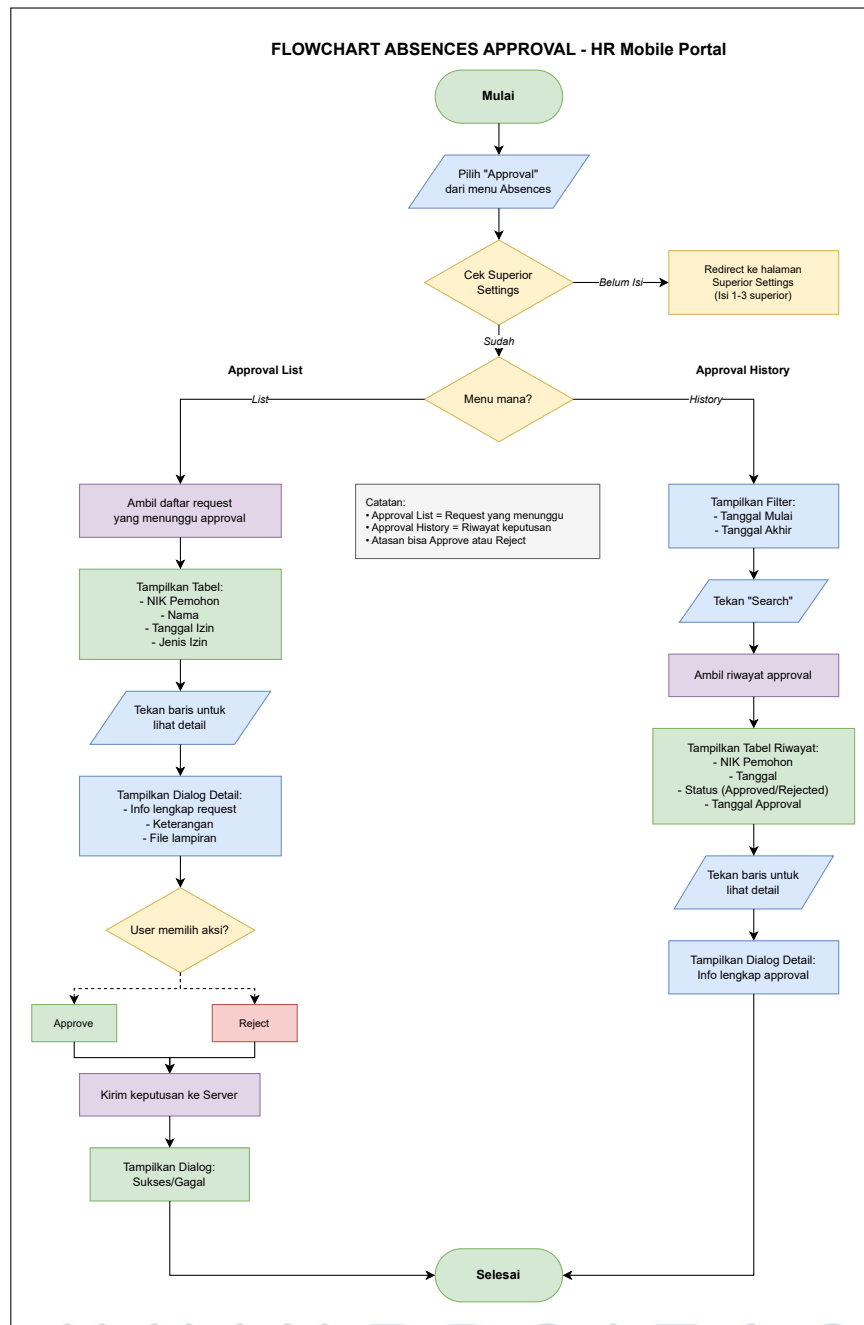
Gambar 3.24. Flowchart Cancellation Absences

Gambar 3.24 menampilkan alur pembatalan pengajuan ketidakhadiran. Pengguna dapat membatalkan pengajuan yang masih dalam status menunggu persetujuan dengan memilih pengajuan dari daftar dan mengonfirmasi pembatalan.



Gambar 3.25. Flowchart History Absences

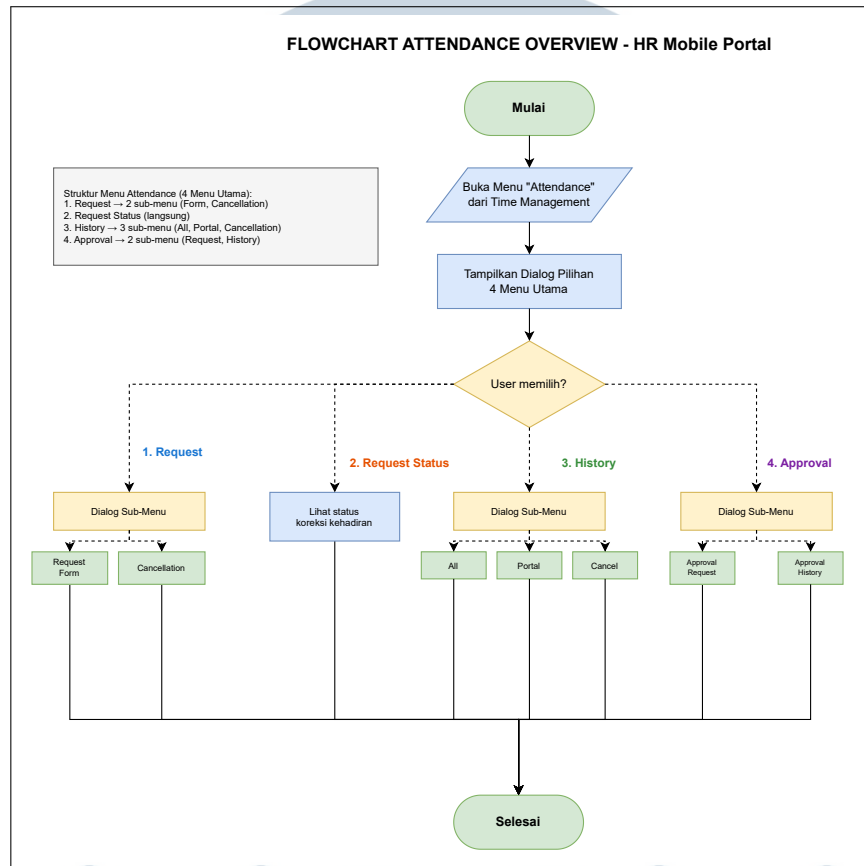
Gambar 3.25 menampilkan alur melihat riwayat ketidakhadiran. Pengguna dapat menelusuri semua pengajuan yang pernah diajukan, memfilter berdasarkan status persetujuan atau periode waktu, dan membuka detail untuk informasi lengkap.



Gambar 3.26. Flowchart Approval Absences

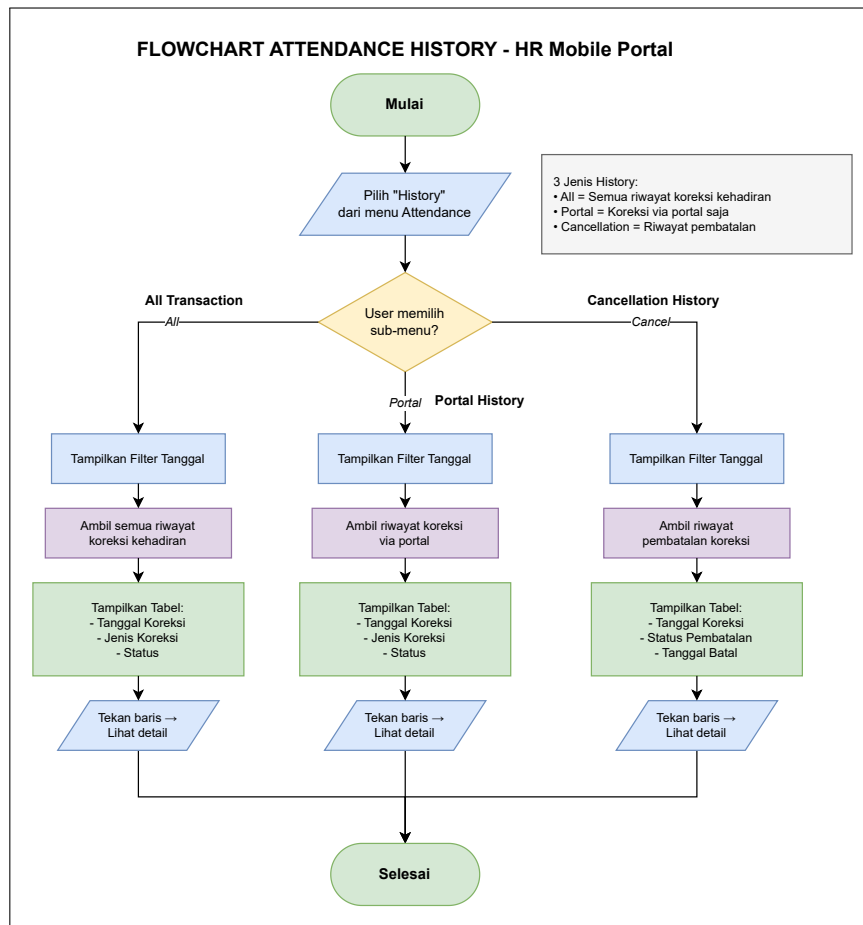
Gambar 3.26 menampilkan alur persetujuan ketidakhadiran oleh atasan. Atasan dapat melihat daftar pengajuan dari bawahannya, membuka detail untuk mempelajari alasan ketidakhadiran, lalu memutuskan untuk menyetujui atau menolak pengajuan tersebut.

G Flowchart Modul Attendance (Kehadiran)



Gambar 3.27. Flowchart Overview Attendance

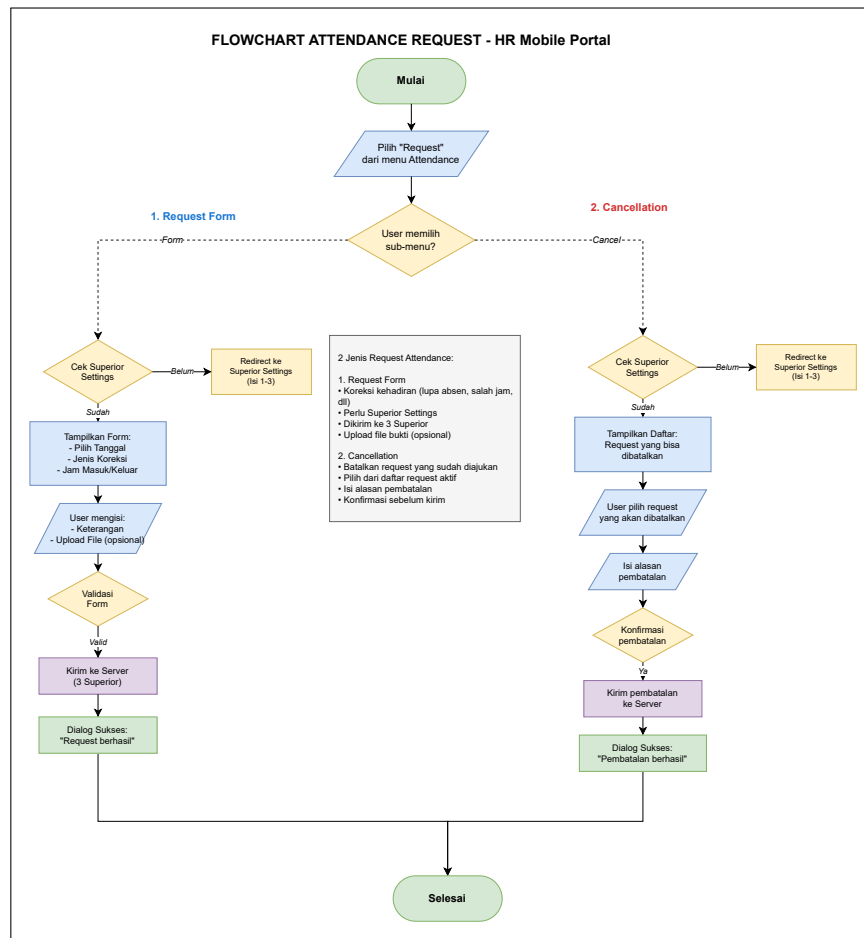
Gambar 3.27 menampilkan struktur menu modul *Attendance*. Modul ini digunakan untuk koreksi data kehadiran jika terjadi kesalahan pencatatan. Tersedia empat opsi: *History* untuk melihat riwayat koreksi, *Request* untuk mengajukan koreksi baru, *Request Status* untuk memantau status, dan *Approval* untuk atasan menyetujui pengajuan.



Gambar 3.28. Flowchart History Attendance

Gambar 3.28 menampilkan alur melihat riwayat koreksi kehadiran. Pengguna dapat menelusuri semua pengajuan koreksi yang pernah dilakukan, melihat statusnya, dan membuka detail untuk informasi lengkap termasuk waktu sebelum dan sesudah dikoreksi.

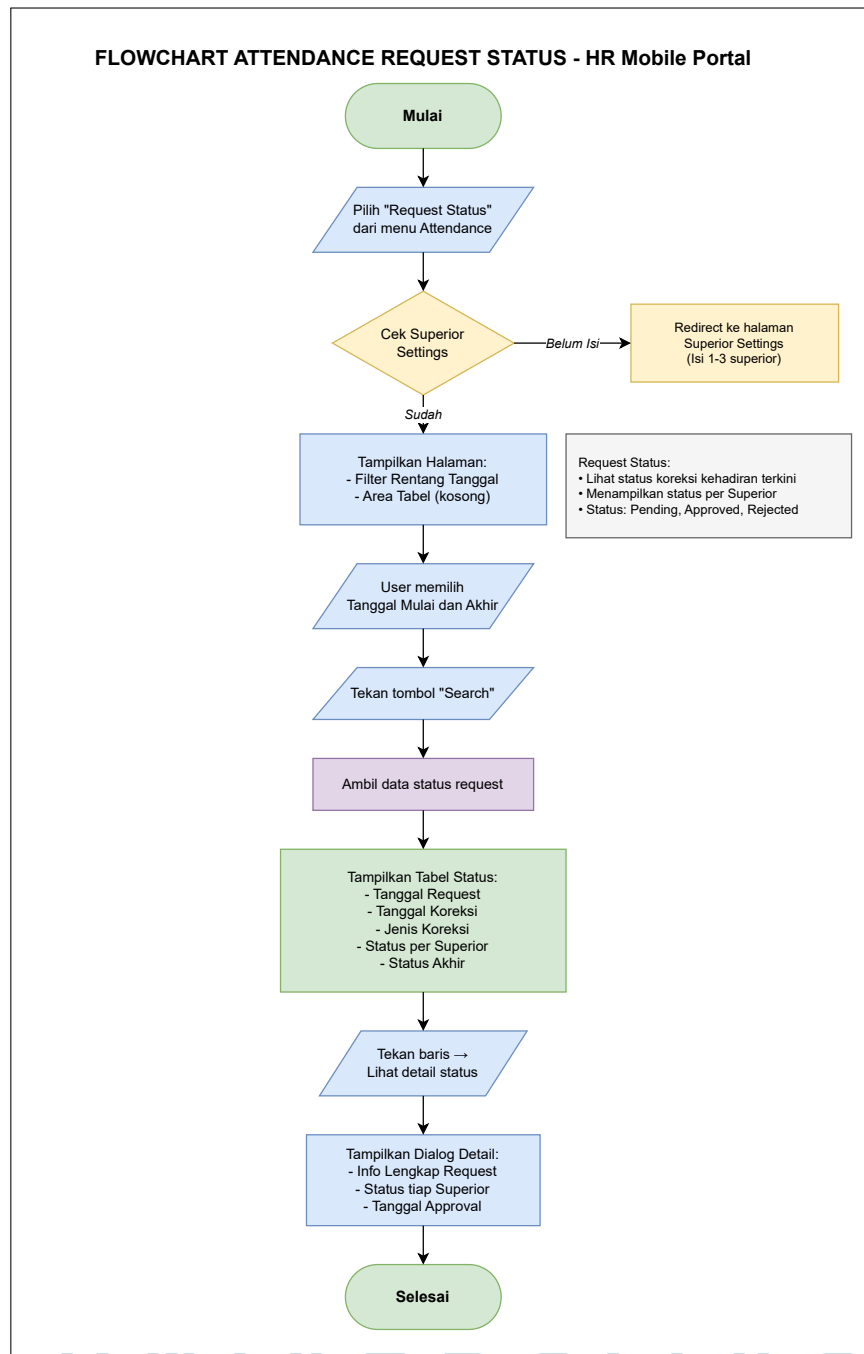
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.29. Flowchart Request Attendance

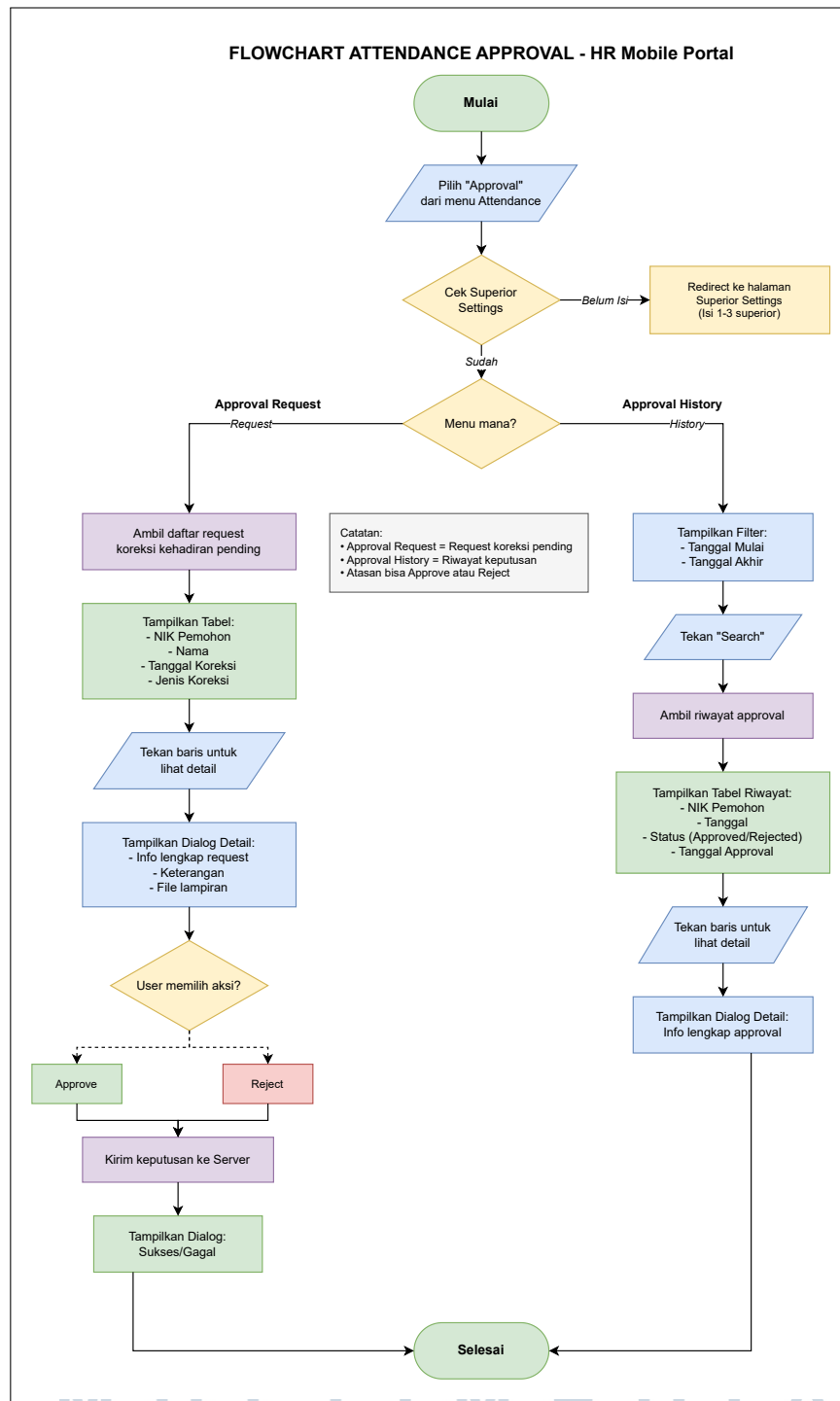
Gambar 3.29 menampilkan alur pengajuan koreksi kehadiran. Pengguna memilih tanggal yang ingin dikoreksi, mengisi waktu masuk dan pulang yang sebenarnya, serta memberikan alasan koreksi.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.30. Flowchart Request Status Attendance

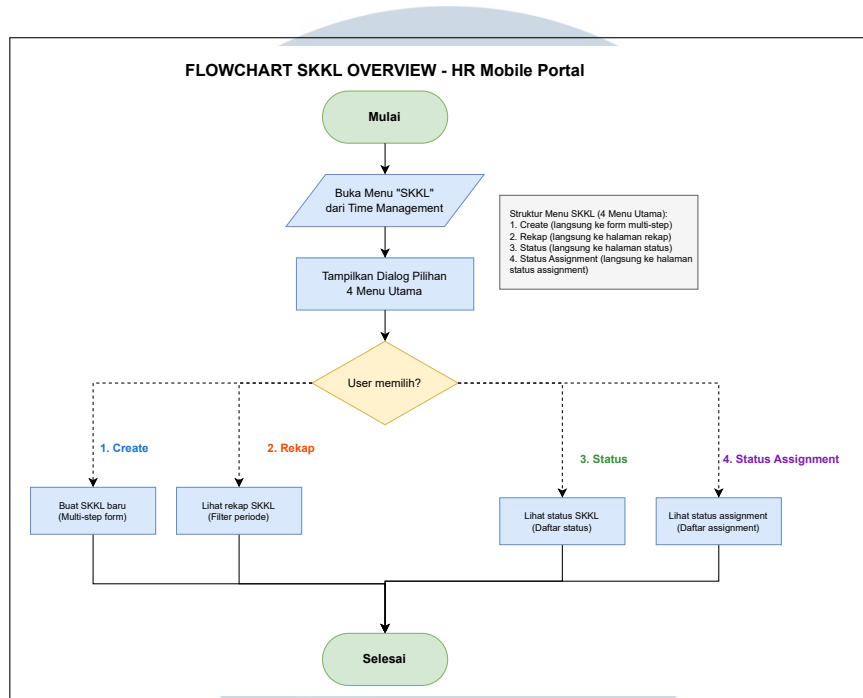
Gambar 3.30 menampilkan alur pemantauan status pengajuan koreksi kehadiran. Pengguna dapat melihat pengajuan mana yang masih menunggu persetujuan, sudah disetujui, atau ditolak oleh atasan.



Gambar 3.31. Flowchart Approval Attendance

Gambar 3.31 menampilkan alur persetujuan koreksi kehadiran oleh atasan. Atasan melihat daftar pengajuan koreksi dari bawahannya, memeriksa alasan dan bukti koreksi, kemudian memberikan keputusan berupa persetujuan atau penolakan.

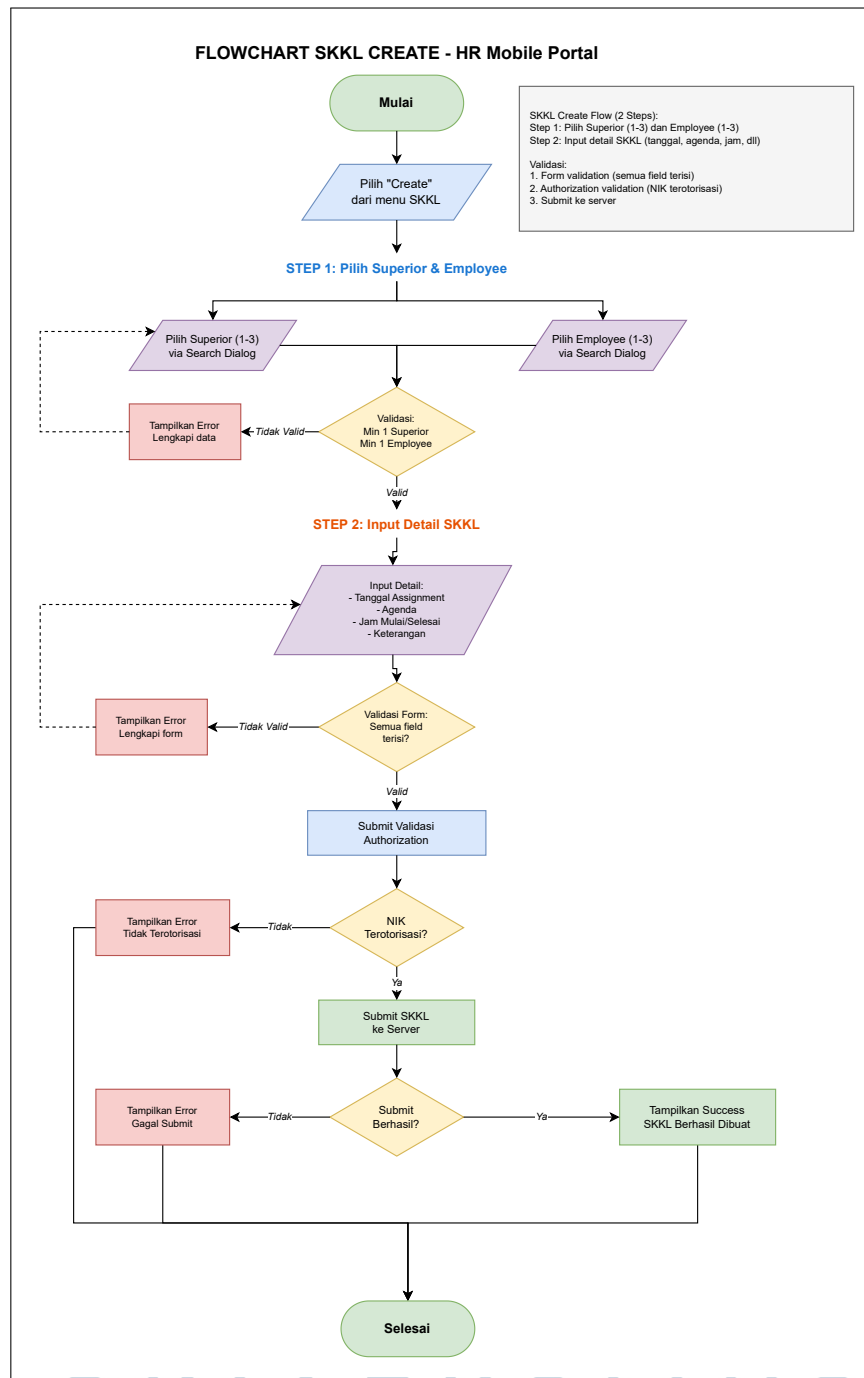
H Flowchart Modul SKKL



Gambar 3.32. Flowchart Overview SKKL

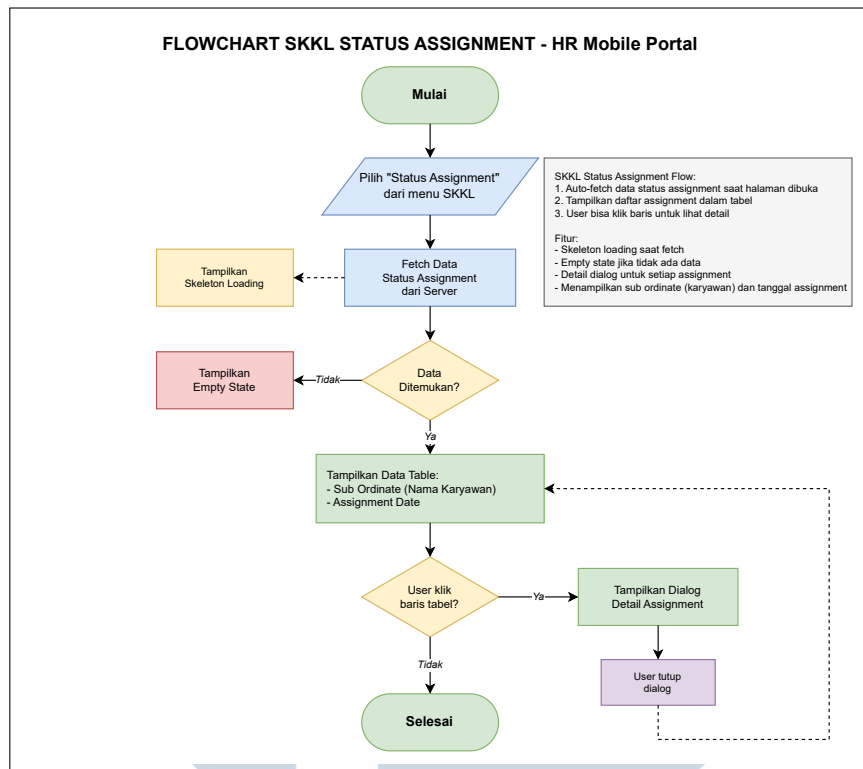
Gambar 3.32 menampilkan struktur menu modul SKKL (*Surat Keterangan Kerja Lembur*) yang terdiri dari empat menu: *Create*, *Status Assignment*, *Status*, dan *Rekap*.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.33. Flowchart Create SKKL

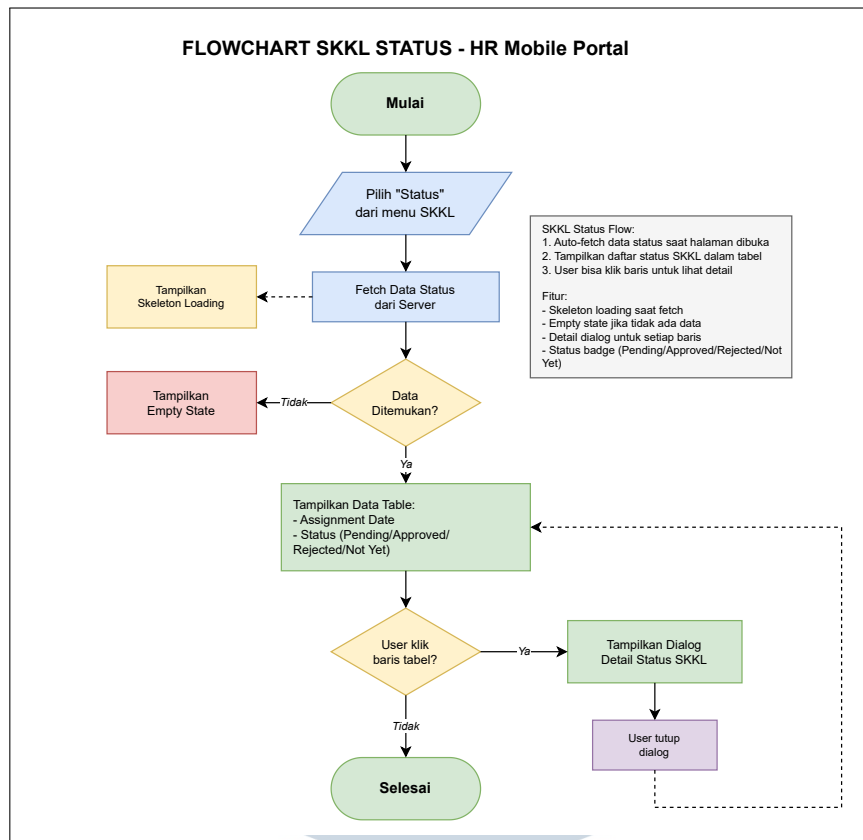
Gambar 3.33 menampilkan alur pembuatan surat lembur baru. Pengguna memilih karyawan, menentukan tanggal dan durasi lembur, mengisi keterangan, dan mengirimkan pengajuan.



Gambar 3.34. Flowchart Status Assignment SKKL

Gambar 3.34 menampilkan alur *Status Assignment* untuk melihat penugasan lembur yang diberikan kepada karyawan. Pengguna dapat melihat daftar penugasan beserta statusnya.

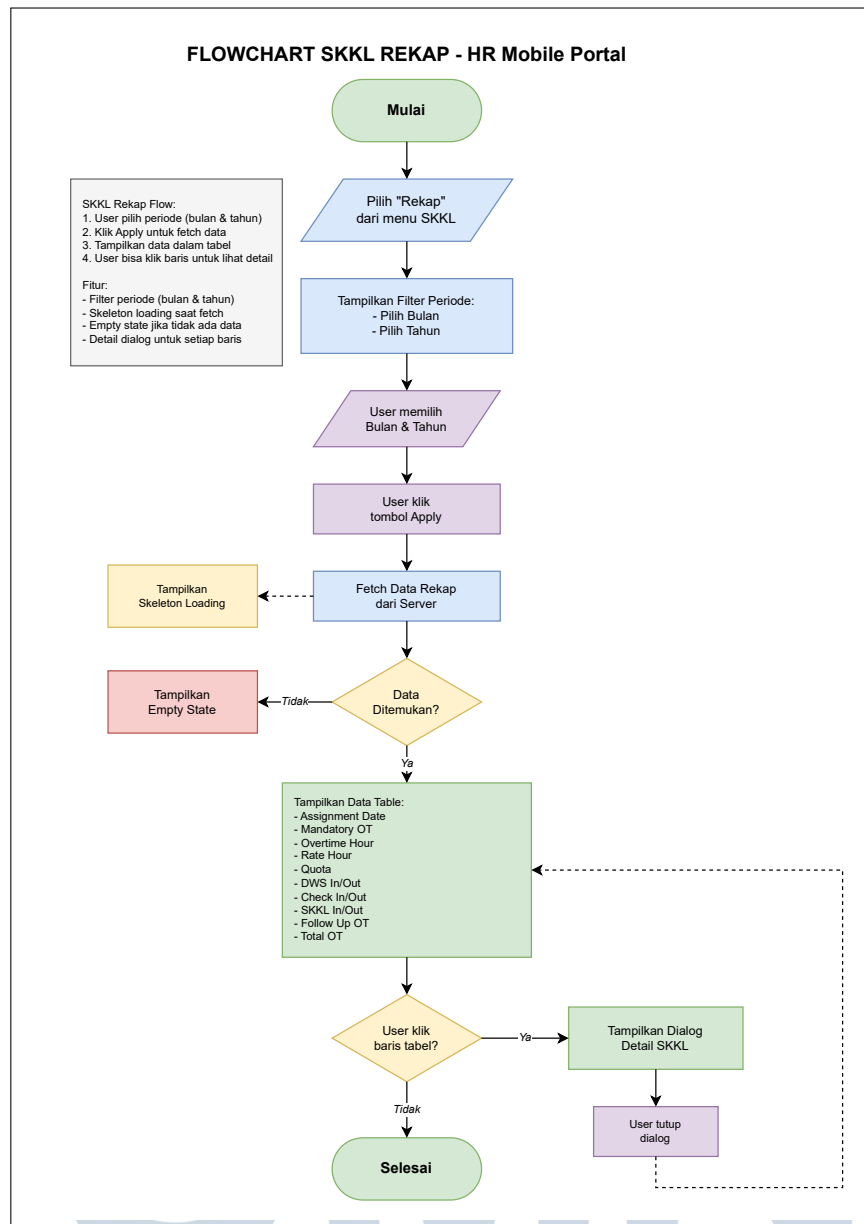
UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.35. Flowchart Status SKKL

Gambar 3.35 menampilkan alur pemantauan status pengajuan SKKL. Pengguna dapat melihat status pengajuan yang sedang diproses, disetujui, atau ditolak.

UMIN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.36. Flowchart Rekap SKKL

Gambar 3.36 menampilkan alur rekapitulasi data lembur. Pengguna dapat melihat ringkasan data lembur berdasarkan periode tertentu dengan *filter* tanggal.

3.3.3 Implementasi Kode Program

Implementasi kode program pada proyek HR Mobile Portal mengikuti arsitektur berlapis yang terstruktur. Setiap layer memiliki tanggung jawab spesifik

untuk memastikan kode yang bersih, terpisah, dan mudah dipelihara. Berikut adalah contoh implementasi kode dari masing-masing layer menggunakan modul *Leave* sebagai studi kasus, karena modul ini merupakan modul paling kompleks dengan total 35 *file*.

A Types - Definisi Tipe Data TypeScript

Layer *types* berisi definisi tipe data menggunakan TypeScript *interfaces*. Tipe data ini memastikan *type-safety* di seluruh aplikasi dan memberikan dokumentasi yang jelas tentang struktur data yang digunakan.

```
1  /**
2   * Types untuk Leave History
3   */
4
5  /**
6   * Leave Transaction - data dari table list API
7   */
8  export interface LeaveHistoryItem {
9    id: string;
10   requestDate: string; // Format: "18 January 2024"
11   leaveType: string;
12   superior1Status: string;
13   superior2Status: string;
14   superior3Status: string;
15 }
16
17 /**
18  * Leave Balance Summary - ringkasan saldo cuti
19  */
20 export interface LeaveBalanceSummary {
21   annualLeave: {
22     balance: number;
23     maxRequest: number;
24     pendingApproval: number;
25   };
26   longLeave: {
27     balance: number;
28     maxRequest: number;
29     pendingApproval: number;
30   };
31 }
```

```

32
33 /**
34  * Query Params untuk filter data
35  */
36 export interface LeaveHistoryQueryParams {
37   startDate: string; // Format: YYYY-MM
38   endDate: string;   // Format: YYYY-MM
39 }

```

Kode 3.1: Definisi Interface untuk Leave History

Kode 3.1 menampilkan definisi *interface* TypeScript untuk modul *Leave History*. Interface *LeaveHistoryItem* mendefinisikan struktur data untuk setiap item transaksi cuti, sedangkan *LeaveBalanceSummary* mendefinisikan struktur ringkasan saldo cuti.

B Parser - Ekstraksi Data dari HTML

Layer *parser* merupakan komponen unik dalam proyek ini. Karena *backend* menggunakan PHP dan mengembalikan respons dalam format HTML, diperlukan fungsi *parser* untuk mengekstrak data dari elemen HTML dan mengonversinya menjadi objek TypeScript.

```

1 import { parse } from 'node-html-parser';
2 import type { LeaveHistoryItem } from '@types/time-management/
   leave';
3
4 /**
5  * Parse Leave History List dari HTML table response
6  */
7 export function parseLeaveHistoryListHTML(html: string):
   LeaveHistoryItem[] {
8   const items: LeaveHistoryItem[] = [];
9
10  try {
11    const root = parse(html);
12    const rows = root.querySelectorAll('tbody tr');
13
14    for (const row of rows) {
15      const cells = row.querySelectorAll('td');
16      if (cells.length < 2) continue;
17
18      const id = row.getAttribute('data-value')?.trim() || '';

```

```

19     const superior1Status = row.getAttribute('data-superior1')?.
trim() || '-';
20     const superior2Status = row.getAttribute('data-superior2')?.
trim() || '-';
21     const superior3Status = row.getAttribute('data-superior3')?.
trim() || '-';
22     const requestDate = cells[0]?.text.trim() || '';
23     const leaveType = cells[1]?.text.trim() || '';
24
25     if (id && requestDate) {
26         items.push({
27             id,
28             requestDate,
29             leaveType,
30             superior1Status,
31             superior2Status,
32             superior3Status,
33         });
34     }
35 }
36 } catch (_error) {
37     // Silent fail - return empty array
38 }
39
40 return items;
41 }

```

Kode 3.2: Parser Function untuk Leave History List

Kode 3.2 menampilkan fungsi *parser* yang menggunakan *library* `node-html-parser` untuk mengekstrak data dari respons HTML. Fungsi ini membaca elemen `<tr>` dalam `<tbody>`, mengekstrak atribut `data-*` dan teks dari sel-sel tabel, kemudian mengonversinya menjadi array objek `LeaveHistoryItem`.

C Service - Integrasi API

Layer *service* bertanggung jawab untuk melakukan pemanggilan API ke *backend*. Layer ini menangani autentikasi, pembuatan *request*, dan penanganan *error*.

```

1 import { API_ENDPOINTS } from '@/constants/api';
2 import type { LeaveHistoryItem, LeaveHistoryQueryParams } from '@/
types/time-management/leave';

```

```

3 import { createFormData, getNIK } from '@utils/api-helpers';
4 import { createCombinedSignal, REQUEST_HEADERS } from '../..//
  shared';
5 import { parseLeaveHistoryListHTML } from './parsers';
6
7 /**
8  * Fetch Leave History List
9  */
10 export async function fetchLeaveHistoryList (
11   params: LeaveHistoryQueryParams,
12   signal?: AbortSignal
13 ): Promise<LeaveHistoryItem[]> {
14   const nik = await getNIK();
15   if (!nik) {
16     throw new Error('User not authenticated. Please login first.')
17   };
18
19   const { signal: combinedSignal, cleanup } = createCombinedSignal
    (signal);
20
21   try {
22     const formData = createFormData({
23       NIK: nik,
24       start_date: params.startDate,
25       end_date: params.endDate,
26     });
27
28     const response = await fetch(API_ENDPOINTS.
    GET_LEAVE_HISTORY_LIST, {
29       method: 'POST',
30       headers: REQUEST_HEADERS,
31       body: formData.toString(),
32       signal: combinedSignal,
33     });
34
35     if (!response.ok) {
36       throw new Error('HTTP error! status: ${response.status}');
37     }
38
39     const html = await response.text();
40     return parseLeaveHistoryListHTML(html);
41   } finally {

```

```

42     cleanup();
43   }
44 }

```

Kode 3.3: Service Function untuk Fetch Leave History

Kode 3.3 menampilkan fungsi *service* yang melakukan pemanggilan API. Fungsi ini mengambil NIK pengguna dari *secure storage*, membuat *form data* dengan parameter yang diperlukan, mengirim *request* POST ke *endpoint* API, dan mengonversi respons HTML menjadi objek menggunakan fungsi *parser*.

D Hook - State Management dengan TanStack Query

Layer *hook* menggunakan TanStack Query (React Query) untuk mengelola *state* data dari server. Layer ini menyediakan fitur *caching*, *refetching*, dan penanganan *loading/error state* secara otomatis.

```

1  import { useQuery } from '@tanstack/react-query';
2  import { fetchLeaveHistoryList } from '@services/time-management/
   leave';
3  import type { LeaveHistoryQueryParams } from '@types/time-
   management/leave';
4
5  /**
6   * Query keys untuk Leave History
7   */
8  const leaveHistoryKeys = {
9    all: ['leave-history'] as const,
10   list: (params: LeaveHistoryQueryParams) =>
11     [...leaveHistoryKeys.all, 'list', params] as const,
12   detail: (id: string) =>
13     [...leaveHistoryKeys.all, 'detail', id] as const,
14 };
15
16 /**
17  * Hook untuk fetch leave history list
18  */
19 export function useLeaveHistoryList(
20   params: LeaveHistoryQueryParams | null,
21   enabled = false
22 ) {
23   return useQuery({
24     queryKey: params

```

```

25     ? leaveHistoryKeys.list(params)
26     : ['leave-history', 'list'],
27     queryFn: ({ signal }) => {
28       if (!params) throw new Error('Params required');
29       return fetchLeaveHistoryList(params, signal);
30     },
31     enabled: enabled && params !== null,
32     staleTime: 2 * 60 * 1000, // 2 minutes
33   });
34 }

```

Kode 3.4: Custom Hook untuk Leave History dengan TanStack Query

Kode 3.4 menampilkan *custom hook* yang membungkus fungsi *service* dengan TanStack Query. Pola `queryKey` digunakan untuk *caching* dan invalidasi data, sedangkan `staleTime` menentukan durasi data dianggap masih valid sebelum perlu di-*fetch* ulang.

E Component - Antarmuka Pengguna React Native

Layer *component* berisi komponen-komponen antarmuka pengguna yang dapat digunakan ulang. Setiap komponen menerima data melalui *props* dan bertanggung jawab untuk menampilkan data tersebut kepada pengguna.

```

1 import { useState } from 'react';
2 import { FlatList, Pressable, StyleSheet, Text, View } from 'react-
  -native';
3 import { Card } from '@components/ui/card';
4 import type { LeaveTransaction, SortOrder } from '@types/time-
  management/leave';
5
6 interface LeaveHistoryTableProps {
7   data: LeaveTransaction[];
8   isLoading: boolean;
9   onRowPress?: (item: LeaveTransaction) => void;
10 }
11
12 /**
13  * Leave History Table Component
14  */
15 export function LeaveHistoryTable({
16   data,
17   isLoading,

```



```

18   onPress,
19 }: LeaveHistoryTableProps) {
20   const [sortBy, setSortBy] = useState<'requestDate' | 'leaveType'>
    >('requestDate');
21   const [sortOrder, setSortOrder] = useState<SortOrder>('asc');
22
23   const sortedData = [...data].sort((a, b) => {
24     if (sortBy === 'requestDate') {
25       const dateA = new Date(a.requestDate).getTime();
26       const dateB = new Date(b.requestDate).getTime();
27       return sortOrder === 'asc' ? dateA - dateB : dateB - dateA;
28     }
29     return a.leaveType.localeCompare(b.leaveType) * (sortOrder ===
    'asc' ? 1 : -1);
30   });
31
32   return (
33     <Card style={styles.container}>
34       <FlatList
35         data={sortedData}
36         keyExtractor={({item}) => item.id}
37         renderItem={({item}) => (
38           <Pressable onPress={() => onPress?.(item)}>
39             <View style={styles.row}>
40               <Text>{item.requestDate}</Text>
41               <Text>{item.leaveType}</Text>
42             </View>
43           </Pressable>
44         )}
45         removeClippedSubviews={true}
46         maxToRenderPerBatch={10}
47       />
48     </Card>
49   );
50 }
51
52 const styles = StyleSheet.create({
53   container: { marginBottom: 16 },
54   row: { flexDirection: 'row', padding: 12 },
55 });

```

Kode 3.5: Component Table untuk Leave History

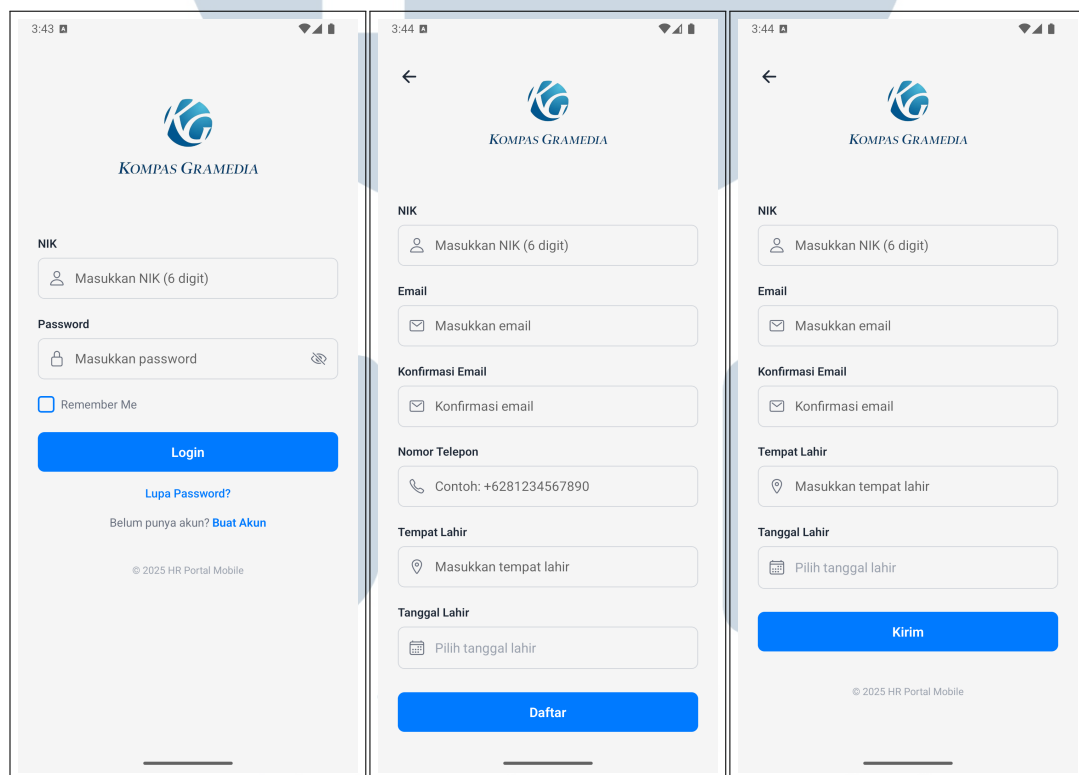
Kode 3.5 menampilkan komponen tabel yang menggunakan FlatList dari

React Native untuk menampilkan daftar transaksi cuti. Komponen ini memiliki fitur *sorting* dan optimasi performa melalui properti `removeClippedSubviews` dan `maxToRenderPerBatch`.

3.3.4 Tampilan Antarmuka Aplikasi

Tampilan antarmuka aplikasi HR Mobile Portal dirancang dengan memperhatikan aspek pengalaman pengguna (*user experience*) dan konsistensi visual. Setiap modul menampilkan dua jenis tampilan utama: tampilan daftar (*list view*) dan tampilan detail (*detail view*).

A Tampilan Autentikasi



(a) Halaman Login

(b) Halaman Register

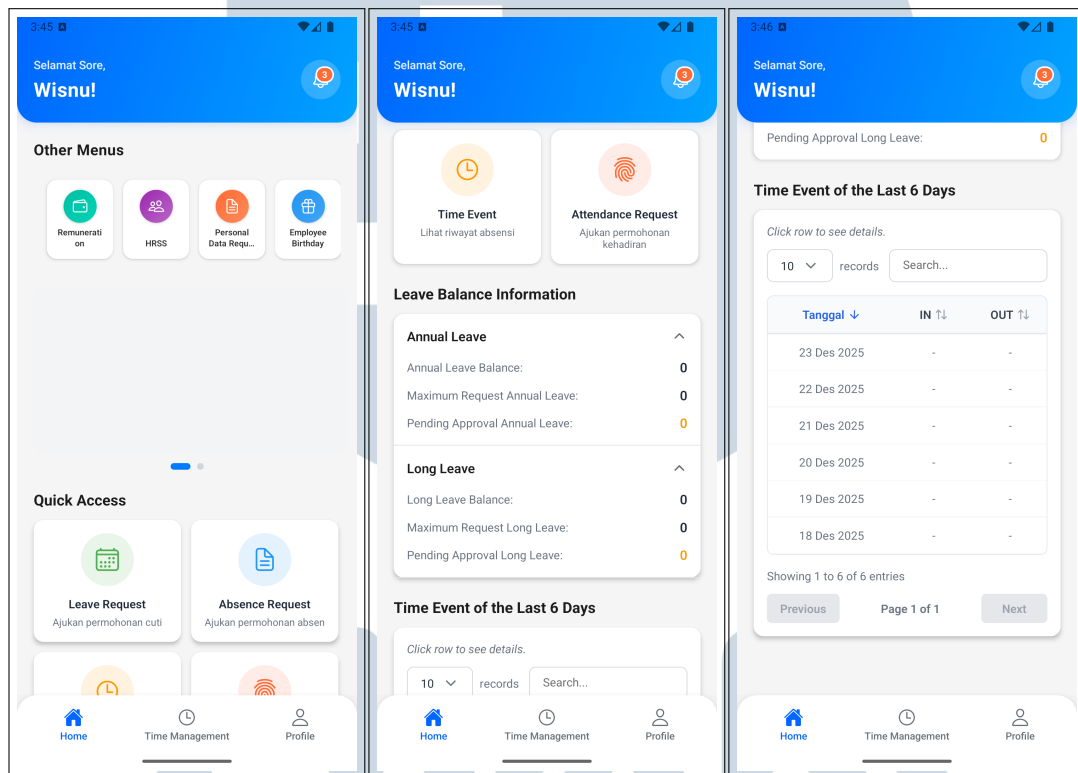
(c) Halaman Lupa Password

Gambar 3.37. Tampilan Modul Autentikasi

Gambar 3.37 menampilkan antarmuka modul autentikasi. Halaman *Login* (Gambar 3.37a) menampilkan form dengan input NIK dan *password*, dilengkapi logo perusahaan dan opsi navigasi ke halaman *Register* serta *Forgot*

Password. Halaman *Register* (Gambar 3.37b) menampilkan form pendaftaran dengan field NIK, email, dan *password*. Halaman *Forgot Password* (Gambar 3.37c) menampilkan form untuk memasukkan email guna menerima tautan *reset password*.

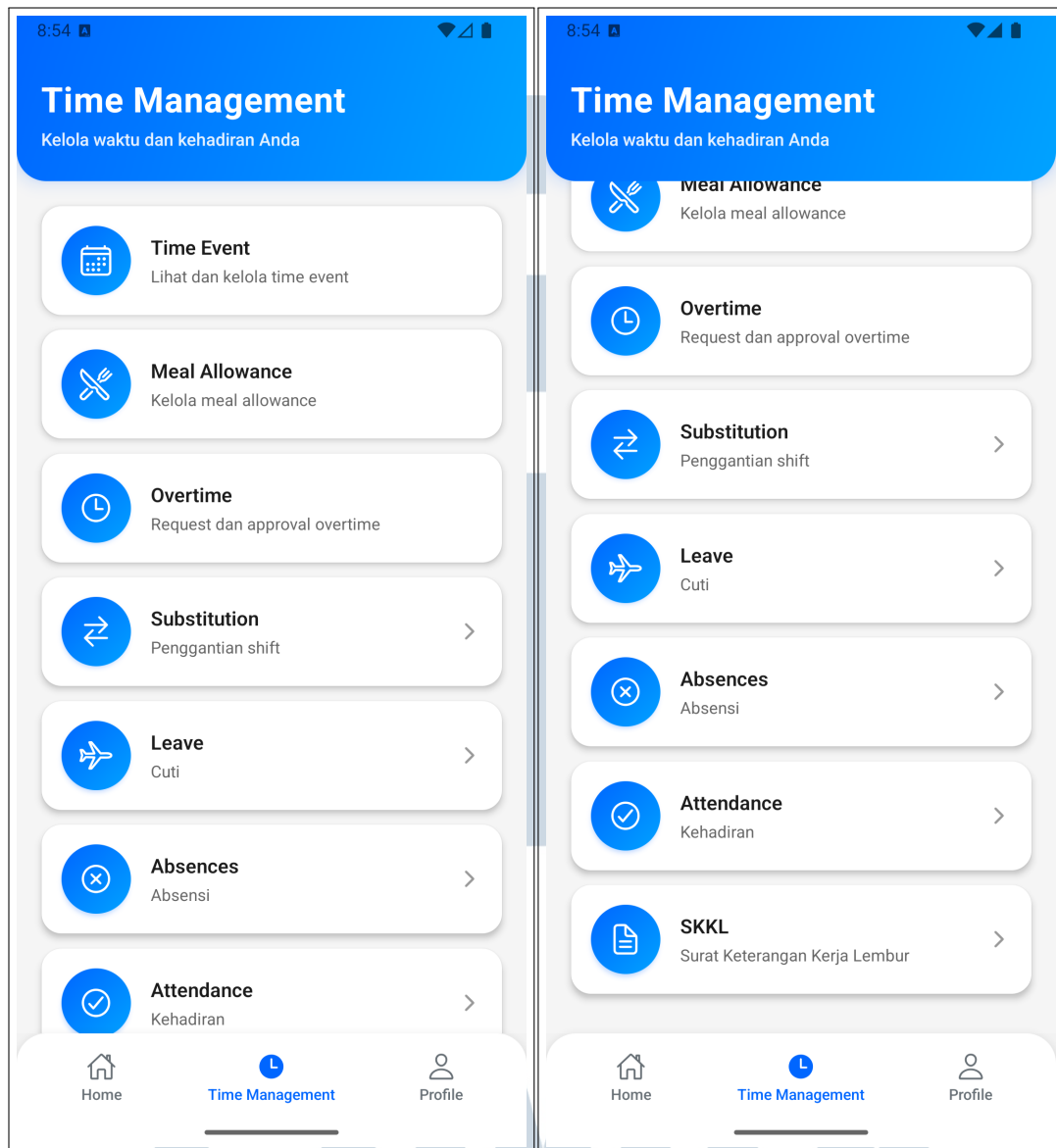
B Tampilan Home dan Time Management



(a) Halaman Home (Atas) (b) Halaman Home (Tengah) (c) Halaman Home (Bawah)

Gambar 3.38. Tampilan Halaman Home

Gambar 3.38 menampilkan antarmuka halaman *Home* yang berfungsi sebagai *dashboard* utama. Bagian atas (Gambar 3.38a) menampilkan *slideshow* informasi dan ringkasan data karyawan. Bagian tengah (Gambar 3.38b) menampilkan *quick access* ke fitur-fitur utama termasuk *Time Management*. Bagian bawah (Gambar 3.38c) menampilkan informasi tambahan dan *Bottom Navigation*.



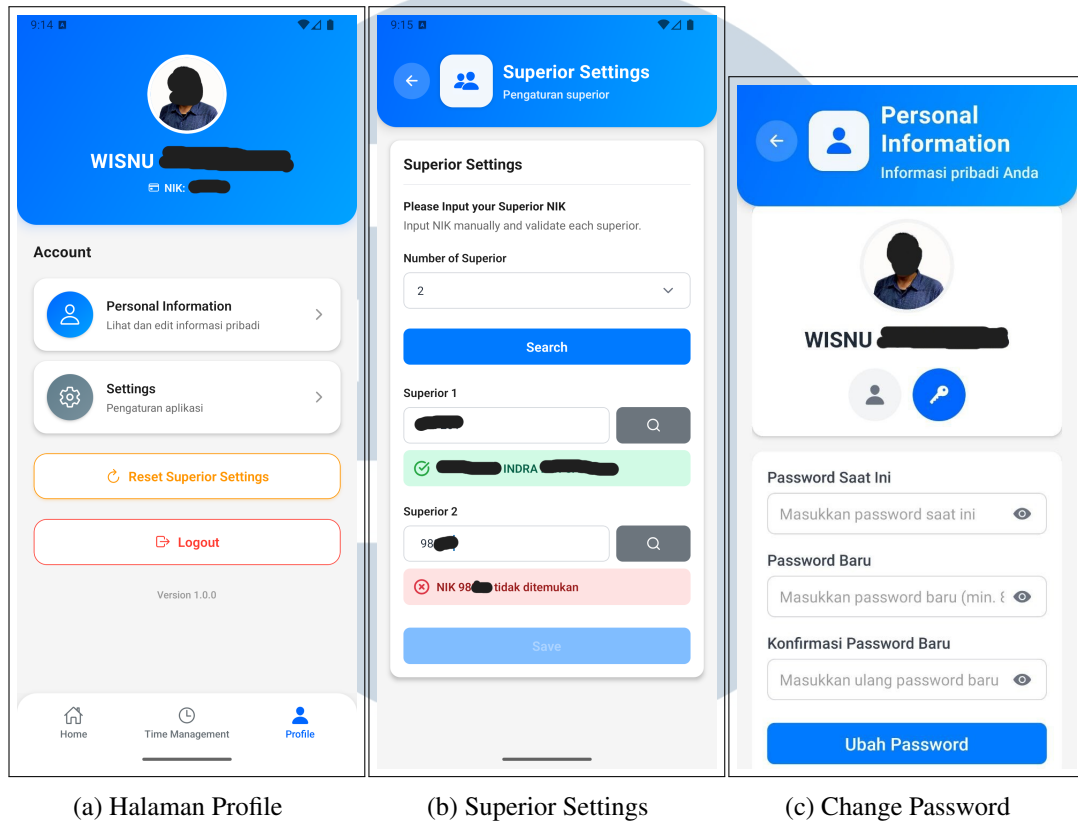
(a) Halaman Time Management (Atas)

(b) Halaman Time Management (Bawah)

Gambar 3.39. Tampilan Halaman Time Management

Gambar 3.39 menampilkan halaman utama modul *Time Management*. Gambar 3.39a menunjukkan bagian atas dengan menu-menu seperti *Leave*, *Absences*, *Attendance*, dan *Substitution*. Gambar 3.39b menampilkan menu tambahan seperti *SKKL*, *Overtime*, *Meal Allowance*, dan *Time Event*.

C Tampilan Profile dan Settings

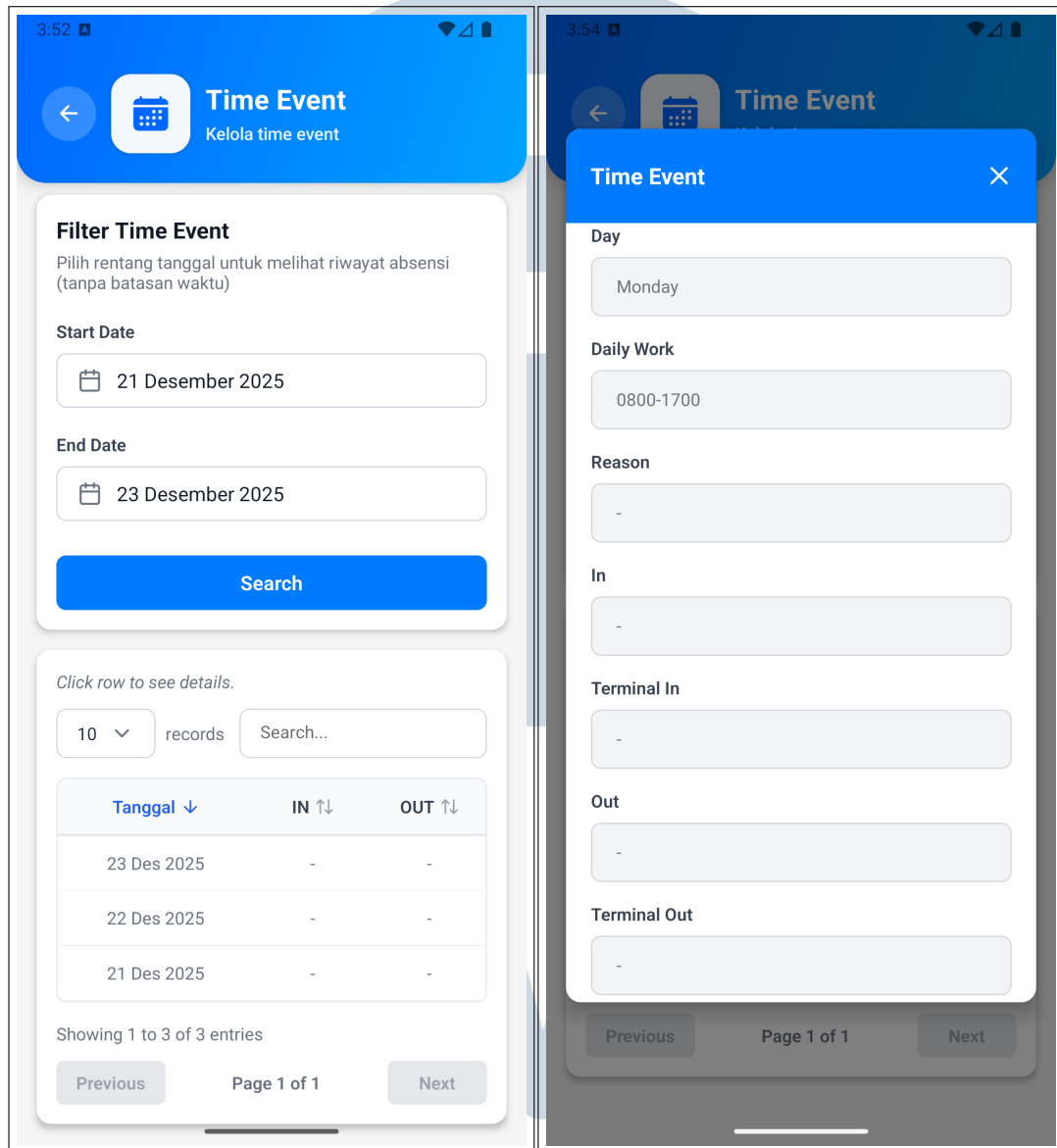


Gambar 3.40. Tampilan Profile dan Settings

Gambar 3.40 menampilkan antarmuka halaman *Profile* dan pengaturan. Halaman *Profile* (Gambar 3.40a) menampilkan informasi pengguna seperti nama, NIK, email, posisi, dan departemen. *Superior Settings* (Gambar 3.40b) menyediakan opsi untuk mengatur atasan pengganti dan *bypass approval*. Halaman *Change Password* (Gambar 3.40c) menampilkan form untuk mengubah *password* akun.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

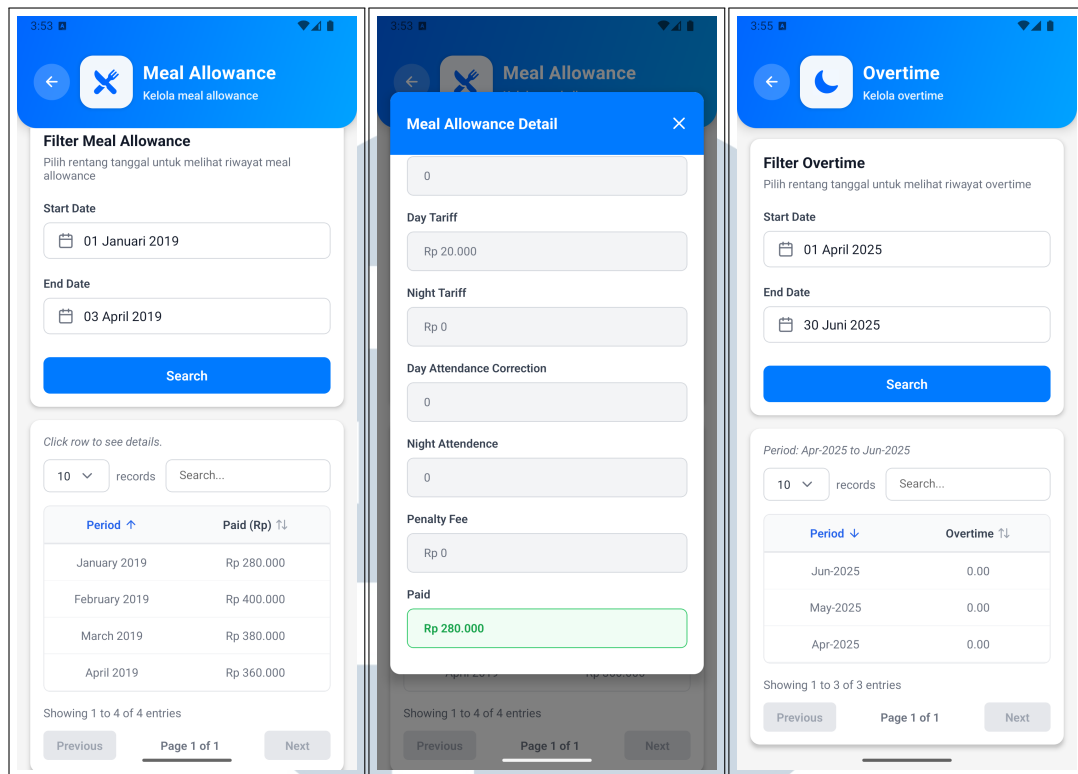
D Tampilan Modul Time Event, Meal Allowance, dan Overtime



(a) Time Event List

(b) Detail Time Event

Gambar 3.41. Tampilan Modul Time Event



(a) Meal Allowance List

(b) Detail Meal Allowance

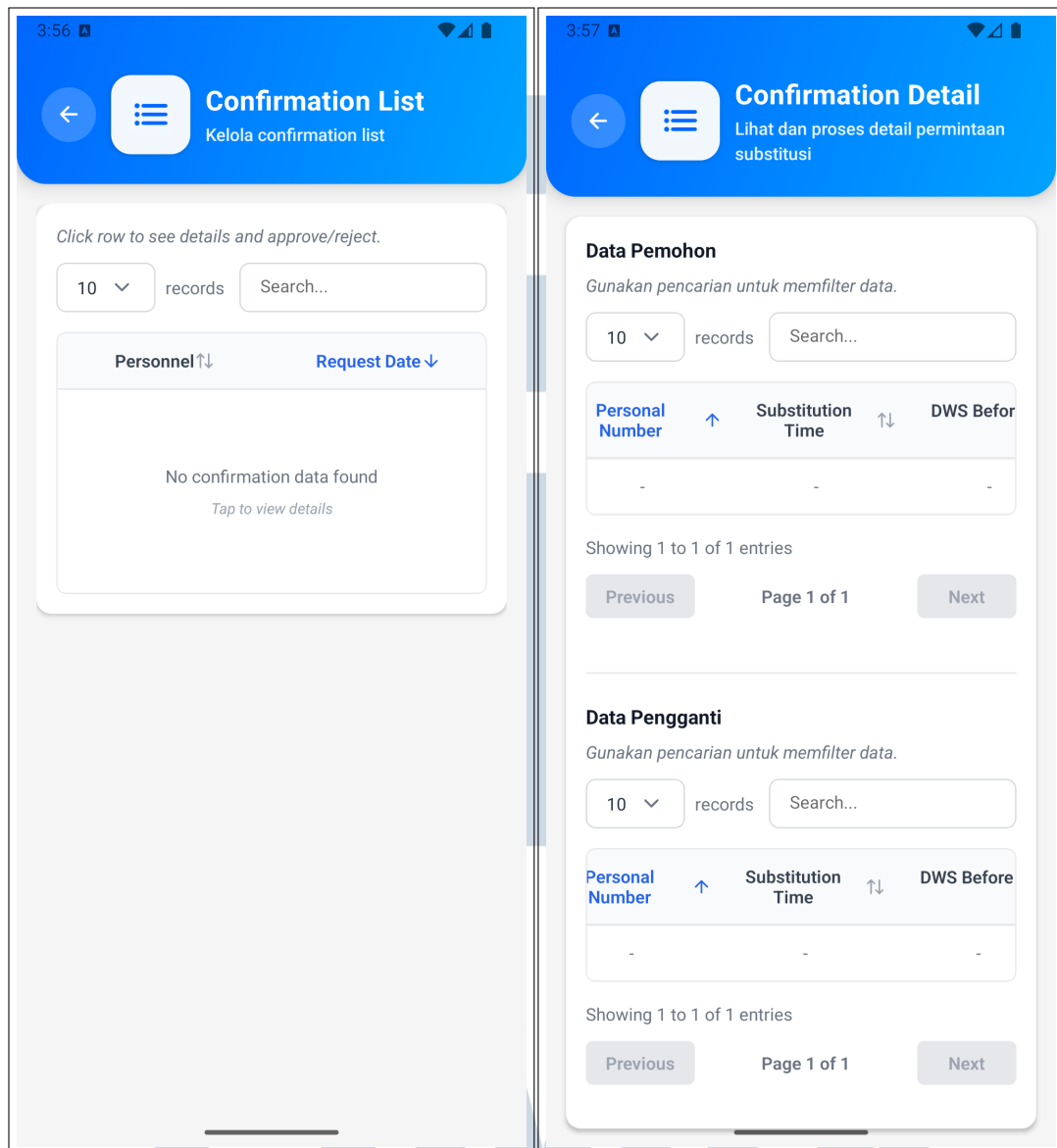
(c) Halaman Overtime

Gambar 3.42. Tampilan Modul Meal Allowance dan Overtime

Gambar 3.41 dan 3.42 menampilkan modul-modul sederhana dalam *Time Management*. *Time Event* (Gambar 3.41a) menampilkan catatan waktu kerja dengan detail (Gambar 3.41b) yang mencakup jam masuk dan pulang. *Meal Allowance* (Gambar 3.42a) menampilkan informasi tunjangan makan dengan detail per hari (Gambar 3.42b). Halaman *Overtime* (Gambar 3.42c) menampilkan data lembur karyawan.

E Tampilan Modul Substitution

Modul *Substitution* memiliki tiga menu utama: *Confirmation List*, *Request*, dan *Approval*. Berikut adalah tampilan masing-masing menu sesuai dengan urutannya dalam aplikasi.

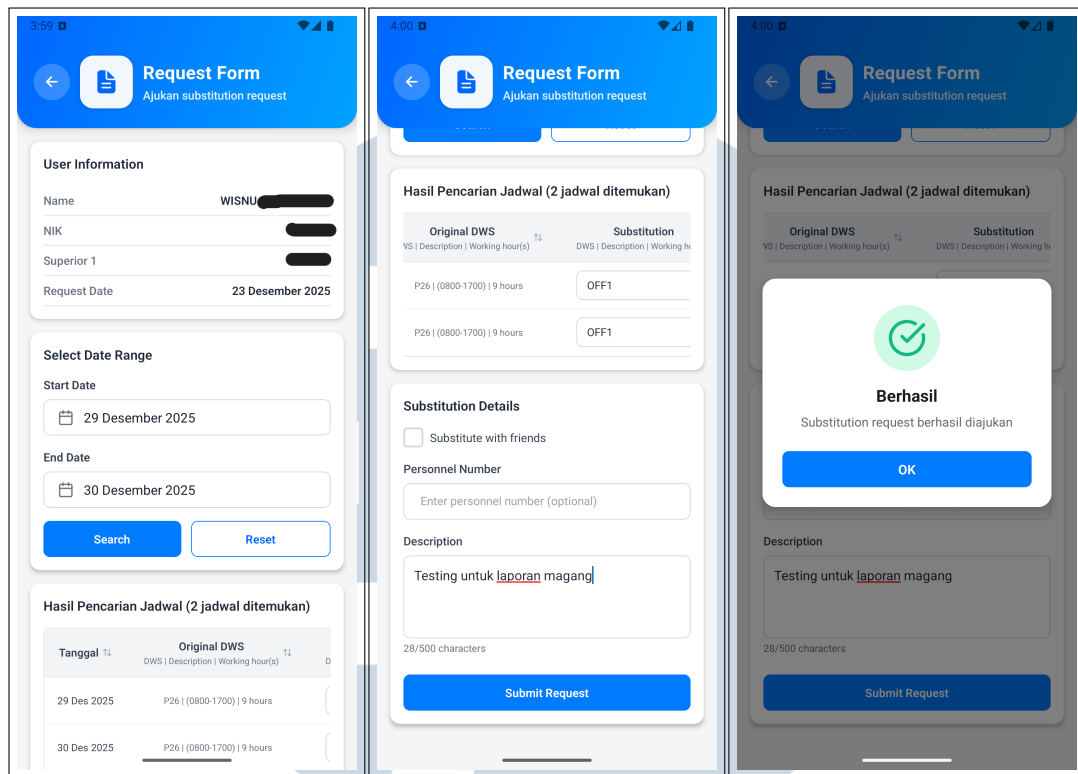


(a) Confirmation List

(b) Detail Confirmation

Gambar 3.43. Tampilan Modul Substitution - Confirmation List

Gambar 3.43 menampilkan menu *Confirmation List* yang merupakan menu pertama dalam modul *Substitution*. Daftar konfirmasi (Gambar 3.43a) menampilkan permintaan substitusi yang ditujukan kepada pengguna untuk dikonfirmasi. Detail konfirmasi (Gambar 3.43b) menampilkan informasi lengkap permintaan beserta opsi untuk menerima atau menolak.



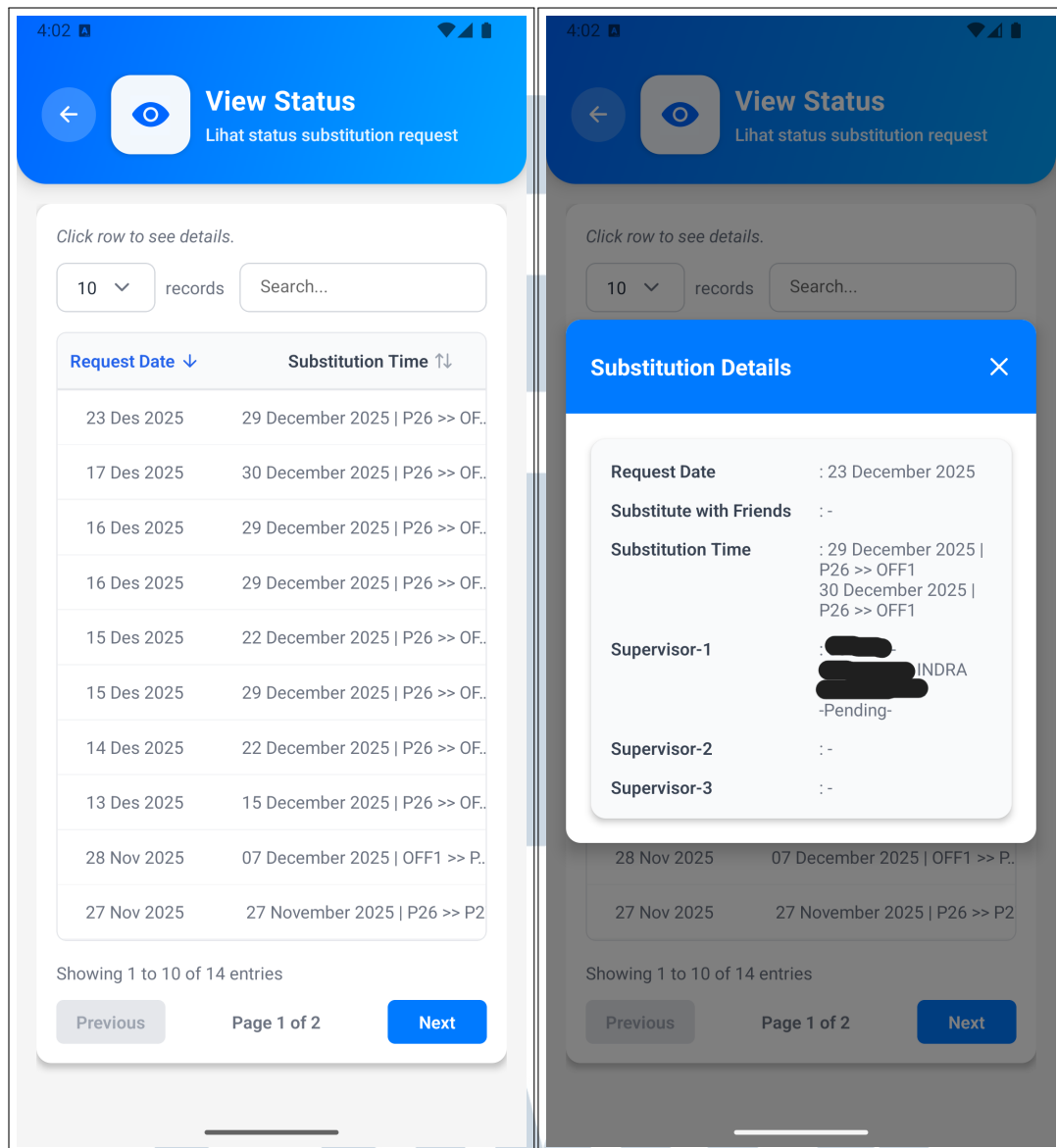
(a) Request Form

(b) Form Terisi

(c) Submit Berhasil

Gambar 3.44. Tampilan Modul Substitution - Request Form

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

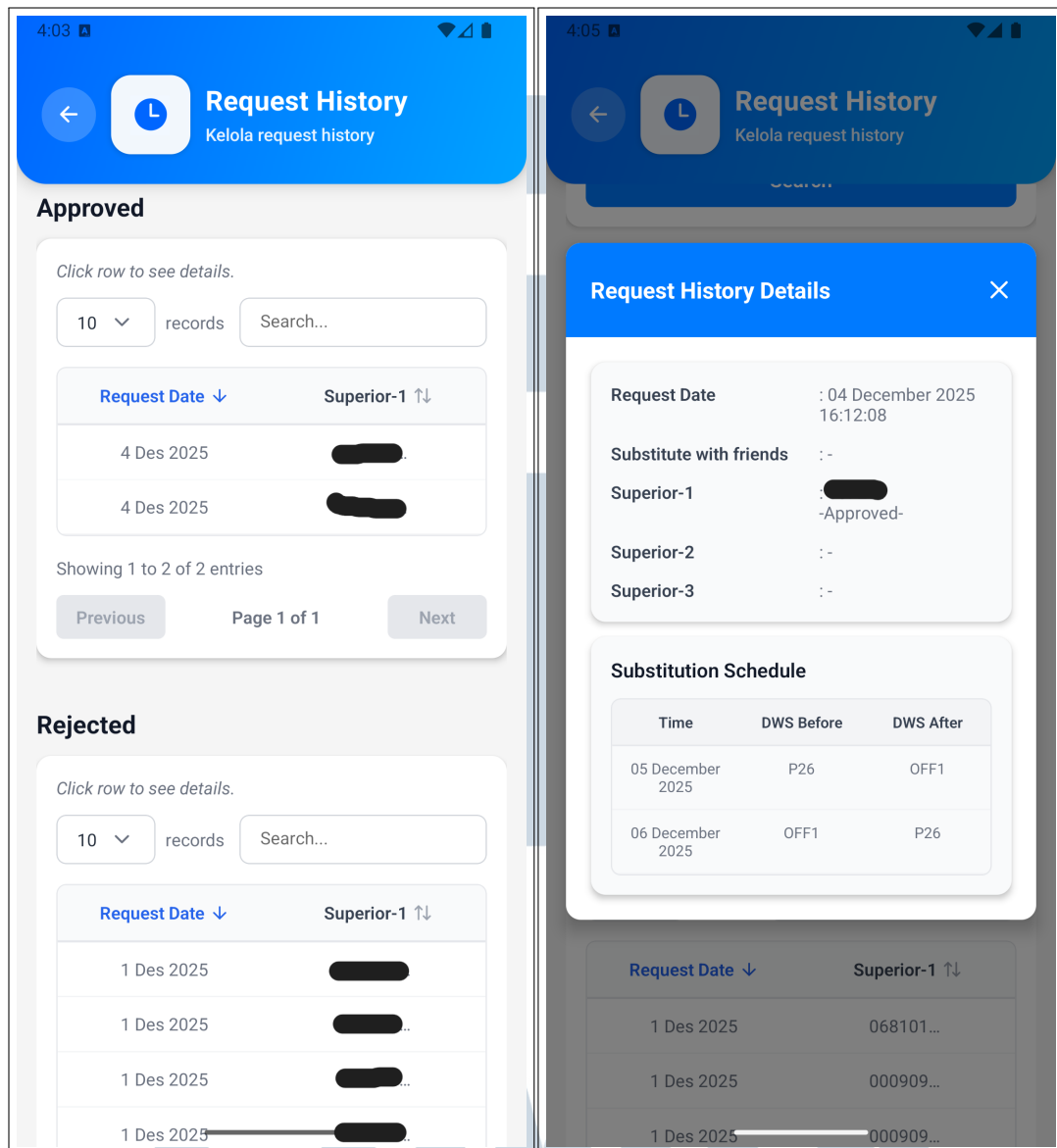


(a) View Status

(b) Detail View Status

Gambar 3.45. Tampilan Modul Substitution - View Status

UNIVERSITAS
MULTIMEDIA
NUSANTARA

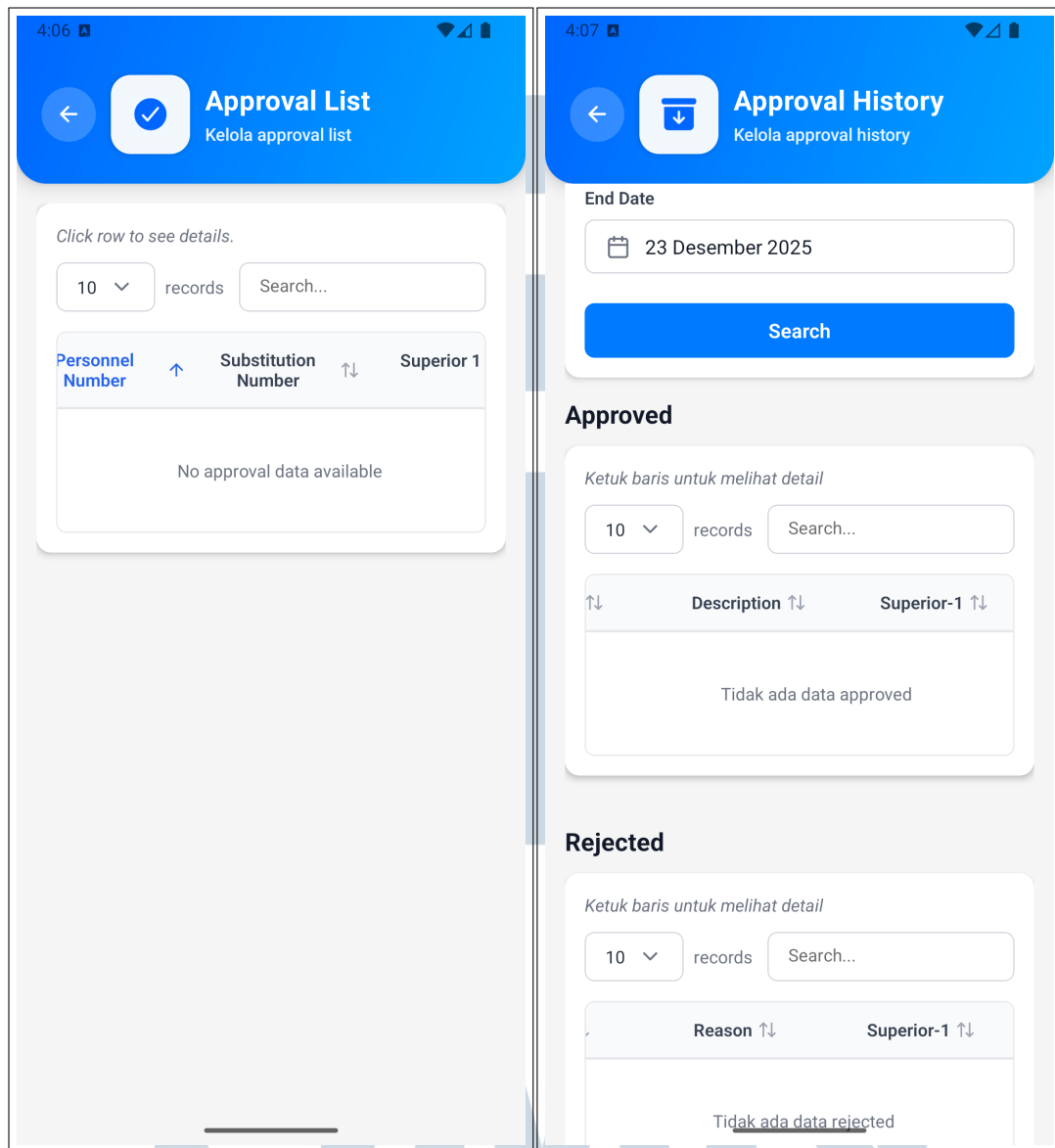


(a) Request History

(b) Detail Request History

Gambar 3.46. Tampilan Modul Substitution - Request History

Gambar 3.44, 3.45, dan 3.46 menampilkan menu *Request* yang memiliki tiga sub-menu. *Request Form* (Gambar 3.44a) digunakan untuk mengajukan substitusi baru, dengan tampilan form terisi (Gambar 3.44b) dan konfirmasi berhasil (Gambar 3.44c). *View Status* (Gambar 3.45a) memantau status pengajuan dengan detail (Gambar 3.45b). *Request History* (Gambar 3.46a) menampilkan riwayat pengajuan dengan detail (Gambar 3.46b).



(a) Approval List

(b) Approval History

Gambar 3.47. Tampilan Modul Substitution - Approval

Gambar 3.47 menampilkan menu *Approval* yang memiliki dua sub-menu. *Approval List* (Gambar 3.47a) menampilkan daftar pengajuan substitusi bawahan yang menunggu persetujuan atasan. *Approval History* (Gambar 3.47b) menampilkan riwayat persetujuan yang telah diberikan oleh atasan, baik yang disetujui maupun ditolak.

F Tampilan Modul Leave

Modul *Leave* memiliki lima menu utama: *Leave Request*, *Request Status*, *Cancellation*, *History*, dan *Approval*. Berikut adalah tampilan masing-masing menu sesuai dengan urutannya dalam aplikasi.

The image displays three screenshots of a mobile application interface for leave management.
Screenshot (a) shows the 'Leave Request' form. It includes fields for 'Superior 1' (INDRA DWI), 'NIK Superior 1', 'Leave Type' (Annual Leave), and 'Number of Leave' (2 Day). Below these is a 'LEAVE DATA' section showing 'Maximum Request Annual Leave: 0', 'Pending Approval Annual Leave: 0', and 'Annual Leave Balance: 0'. A 'PLEASE INPUT NUMBER OF LEAVE' section is also present.
Screenshot (b) shows the 'Form Validation Error' screen. It displays 'DATES' with 'Date Request' (23 Desember 2025), 'Day 1' (29 Desember 2025), and 'Day 2' (30 Desember 2025). It also shows a 'Contact' field (0867623232) and a 'Notes' field (Testing untuk magang). A red error message states: 'Mohon maaf untuk sementara waktu request hanya bisa dilakukan melalui HR Portal Web'.
Screenshot (c) shows the 'Request Status' screen. It displays 'Leave Request Status' with 'Annual Leave' and 'Long Leave' sections. Both sections show 'Annual Leave Balance: 0', 'Maximum Request Annual Leave: 0', and 'Pending Approval Annual Leave: 0'. A table below shows 'Request Date' and 'Leave Type' with a 'No request history found' message.

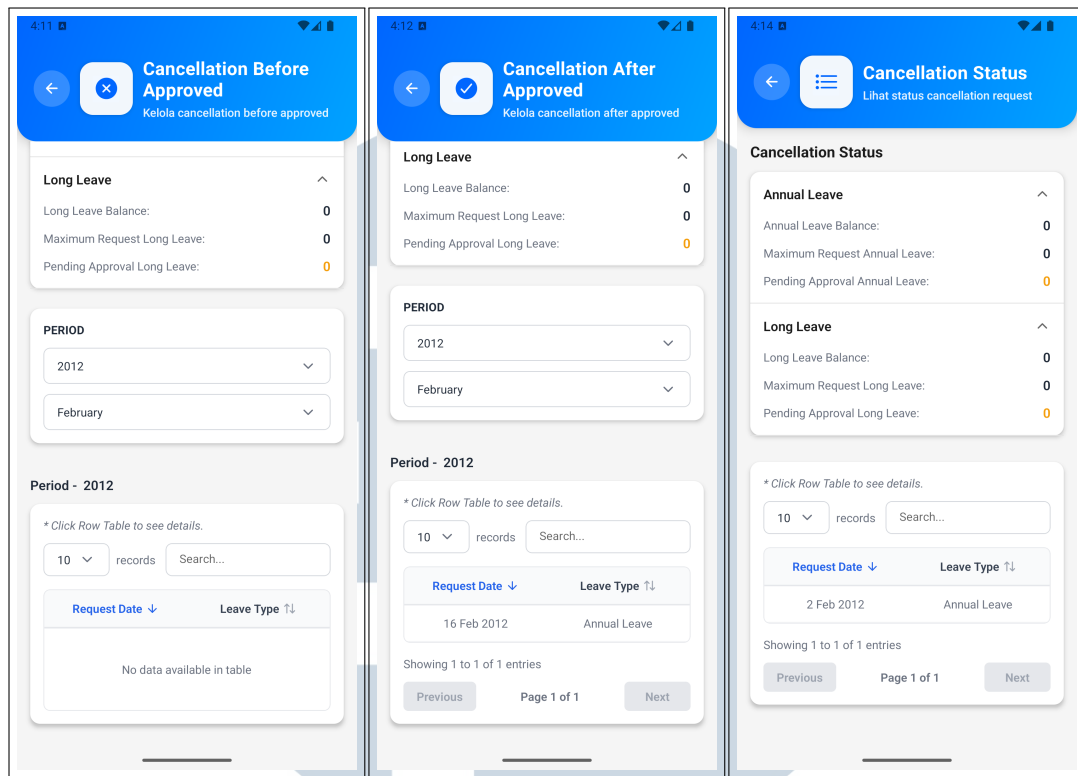
(a) Leave Request Form

(b) Form Validation Error

(c) Request Status

Gambar 3.48. Tampilan Modul Leave - Leave Request dan Request Status

Gambar 3.48 menampilkan dua menu pertama dalam modul *Leave*. *Leave Request Form* (Gambar 3.48a) menyediakan field untuk jenis cuti, tanggal, substitut, dan keterangan dengan tampilan validasi error (Gambar 3.48b). *Request Status* (Gambar 3.48c) menampilkan daftar pengajuan yang sedang diproses.



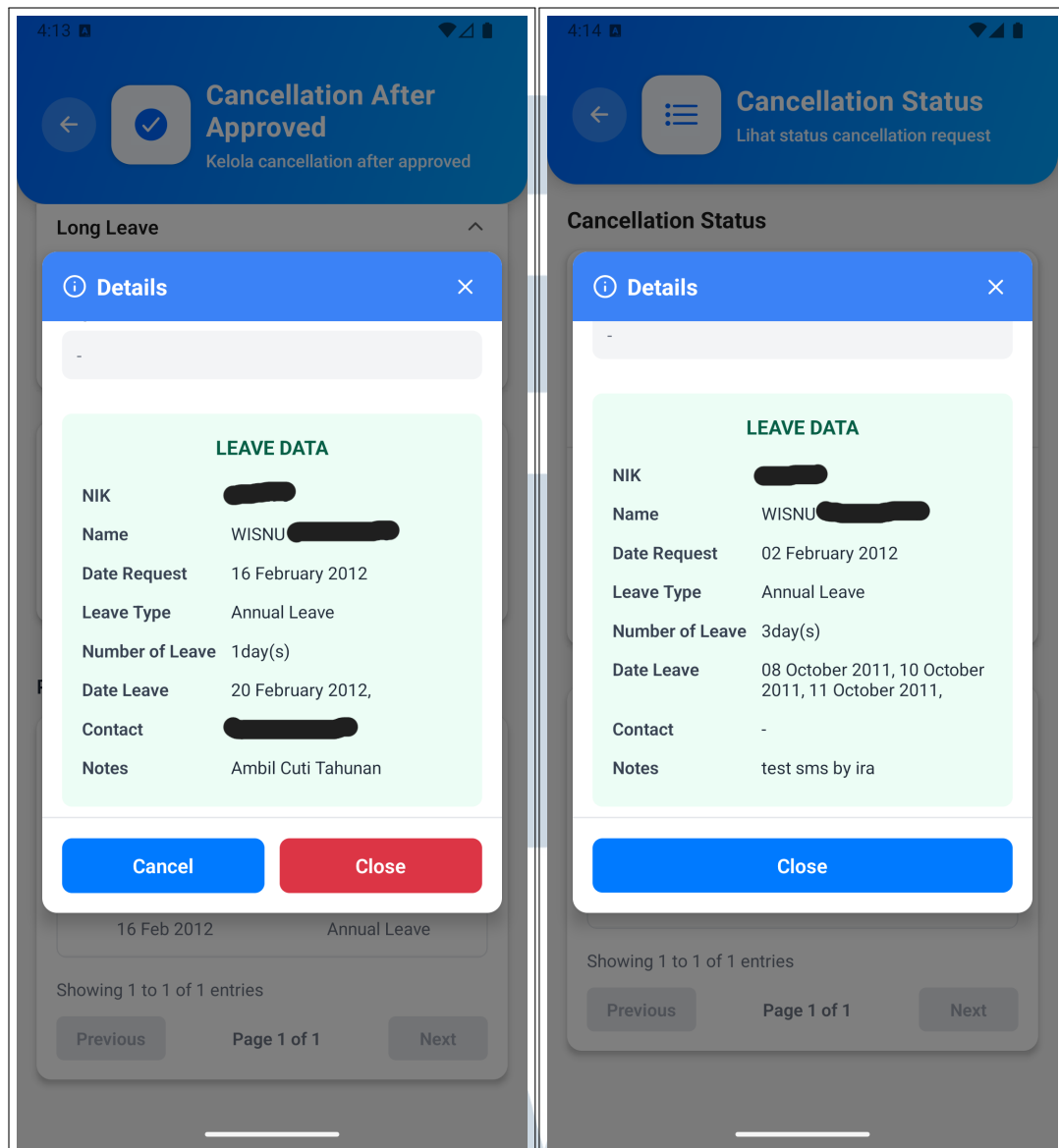
(a) Before Approved

(b) After Approved

(c) Cancellation Status

Gambar 3.49. Tampilan Modul Leave - Cancellation

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

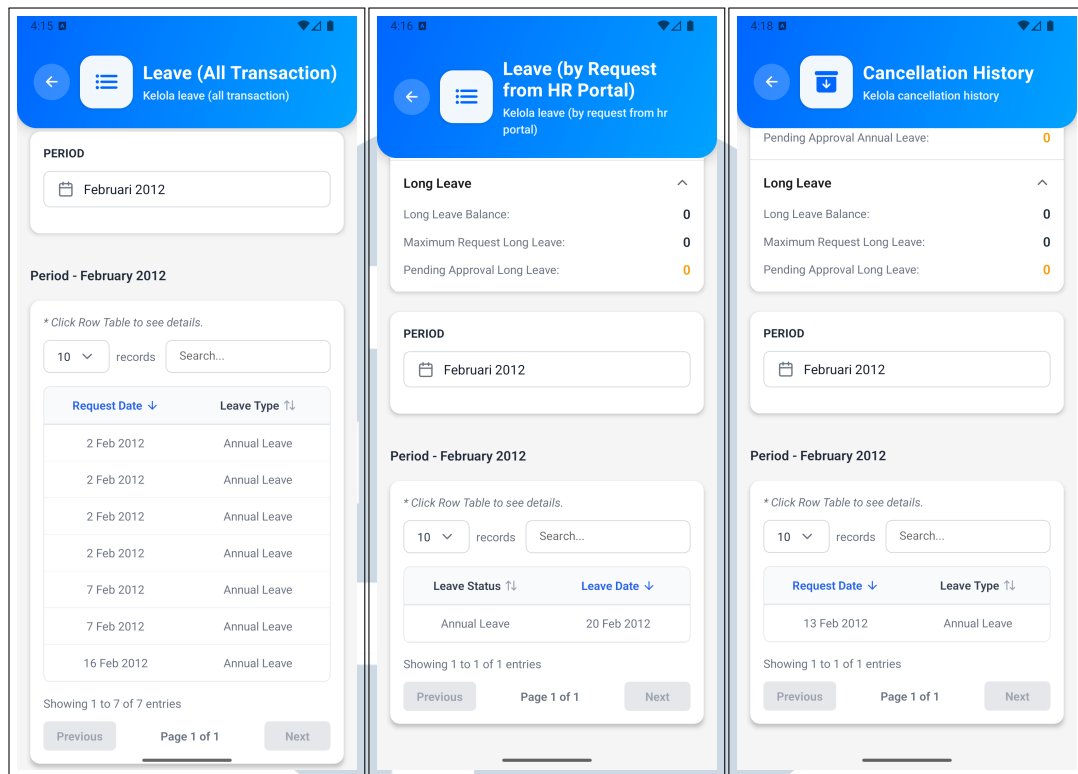


(a) Detail After Approved

(b) Detail Cancellation Status

Gambar 3.50. Tampilan Modul Leave - Detail Cancellation

Gambar 3.49 dan 3.50 menampilkan menu *Cancellation* yang memiliki tiga sub-menu. *Before Approved* (Gambar 3.49a) untuk membatalkan cuti sebelum disetujui. *After Approved* (Gambar 3.49b) untuk membatalkan cuti yang sudah disetujui dengan detail (Gambar 3.50a). *Cancellation Status* (Gambar 3.49c) memantau status pembatalan dengan detail (Gambar 3.50b).



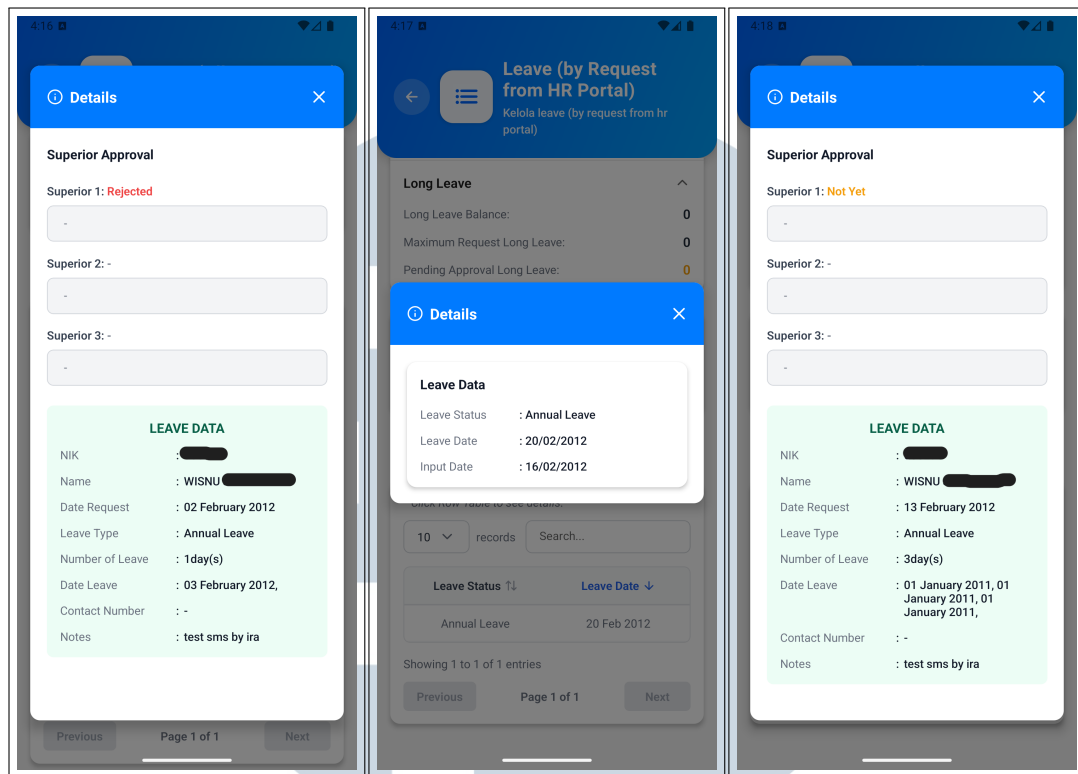
(a) All Transaction

(b) HR Portal

(c) Cancellation History

Gambar 3.51. Tampilan Modul Leave - History

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



(a) Detail All Transaction

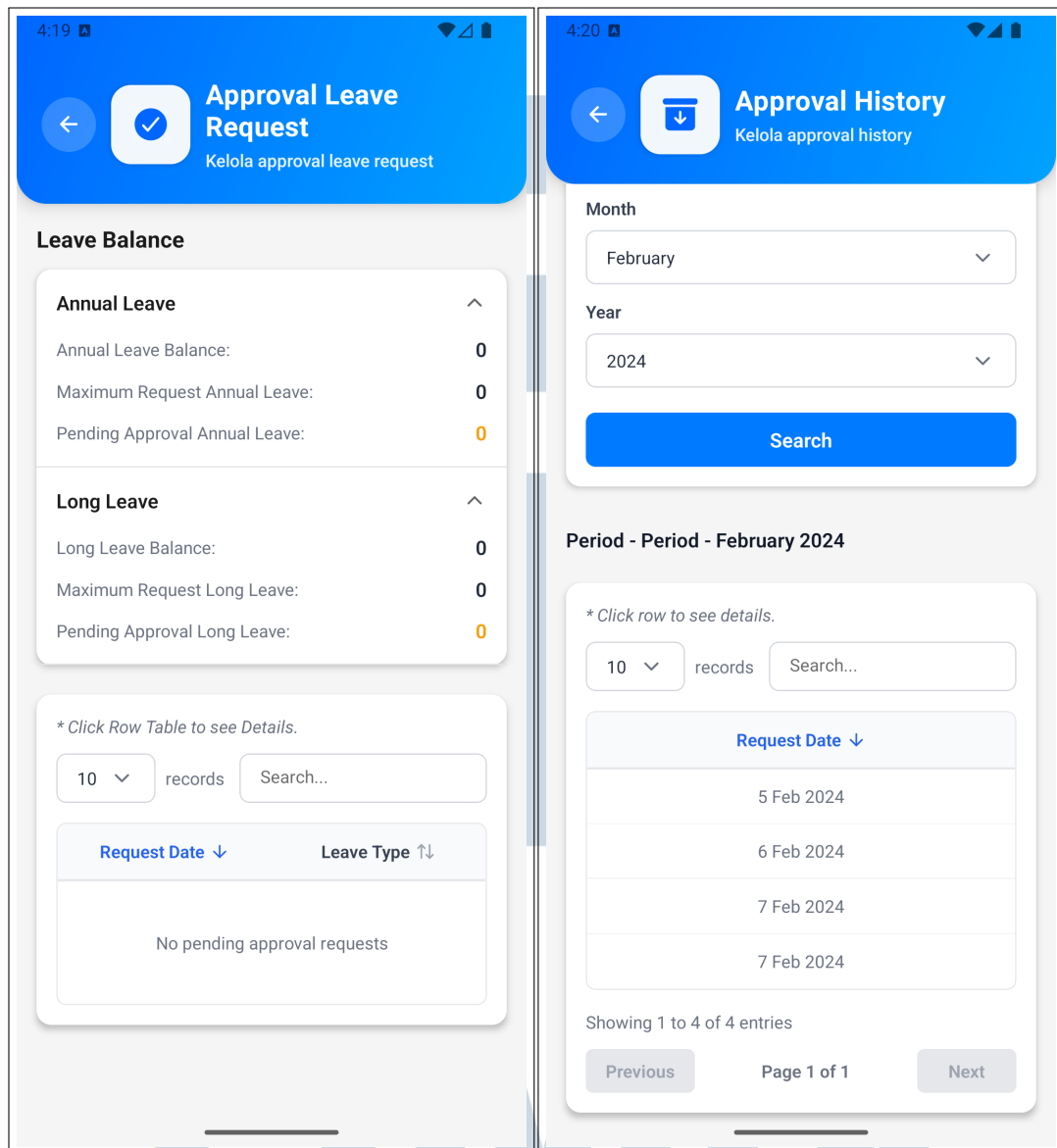
(b) Detail HR Portal

(c) Detail Cancellation History

Gambar 3.52. Tampilan Modul Leave - Detail History

Gambar 3.51 dan 3.52 menampilkan menu *History* yang memiliki tiga sub-menu. *All Transaction* (Gambar 3.51a) menampilkan semua riwayat dengan detail (Gambar 3.52a). *HR Portal* (Gambar 3.51b) menampilkan riwayat dengan detail (Gambar 3.52b). *Cancellation History* (Gambar 3.51c) menampilkan riwayat pembatalan dengan detail (Gambar 3.52c).

UNIVERSITAS
MULTIMEDIA
NUSANTARA

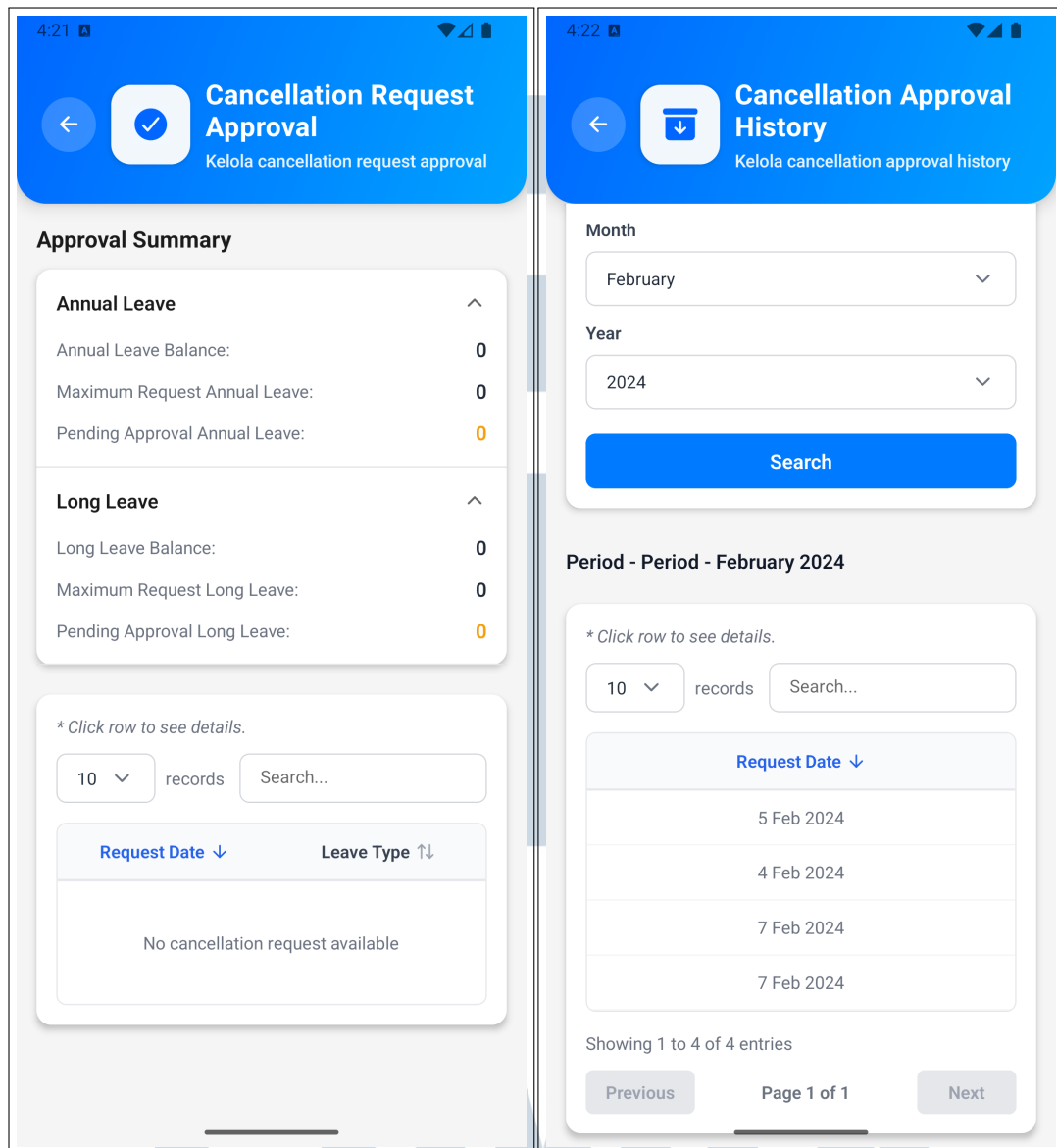


(a) Approval Request

(b) Approval History

Gambar 3.53. Tampilan Modul Leave - Approval (Request dan History)

UNIVERSITAS
MULTIMEDIA
NUSANTARA

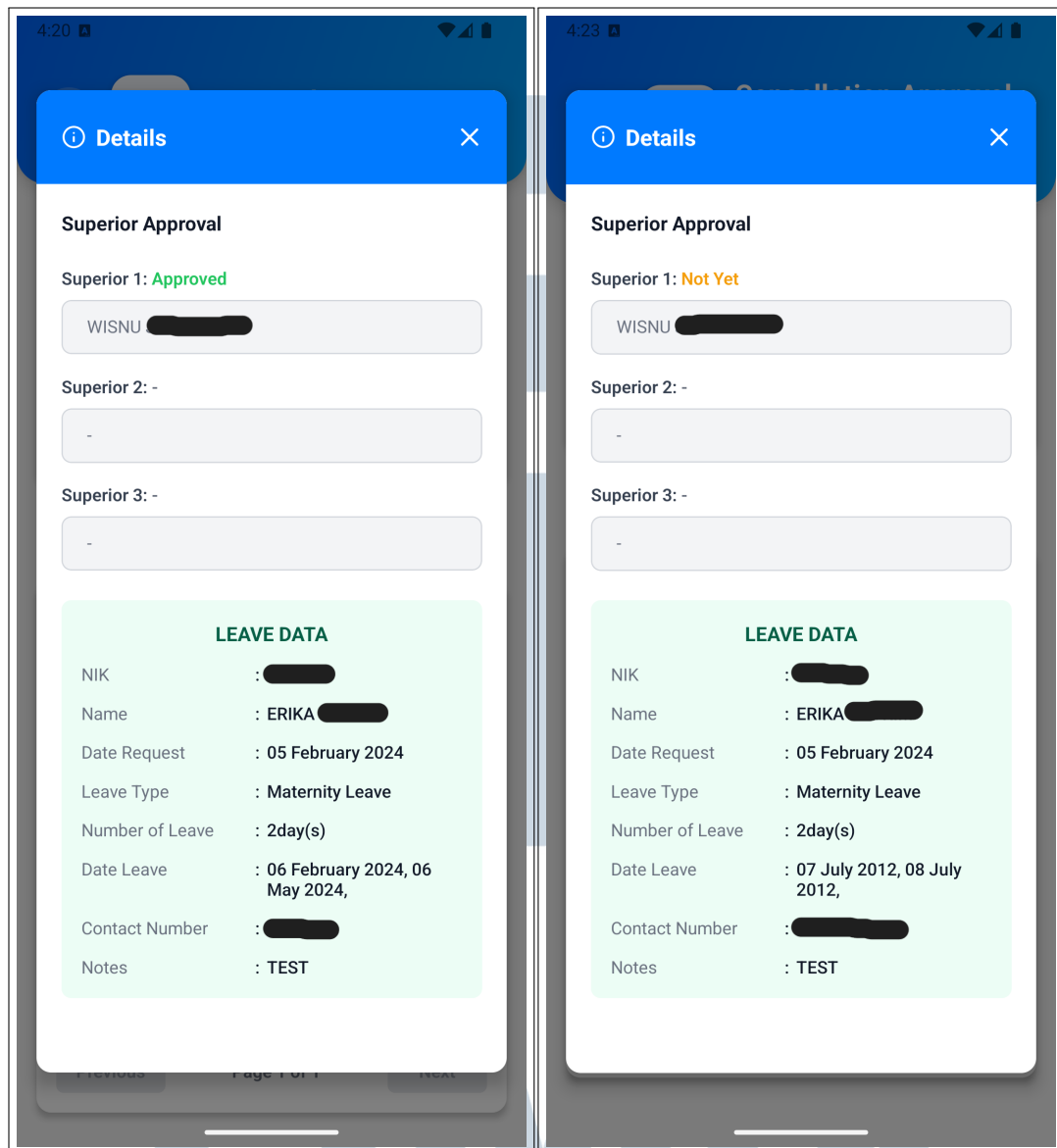


(a) Cancellation Approval Request

(b) Cancellation Approval History

Gambar 3.54. Tampilan Modul Leave - Approval (Cancellation)

UNIVERSITAS
MULTIMEDIA
NUSANTARA



(a) Detail Approval History

(b) Detail Cancellation Approval History

Gambar 3.55. Tampilan Modul Leave - Detail Approval

Gambar 3.53, 3.54, dan 3.55 menampilkan menu *Approval* yang memiliki empat sub-menu. *Approval Request* (Gambar 3.53a) menampilkan daftar pengajuan cuti bawahan. *Approval History* (Gambar 3.53b) menampilkan riwayat persetujuan dengan detail (Gambar 3.55a). *Cancellation Approval Request* (Gambar 3.54a) untuk menyetujui pembatalan cuti. *Cancellation Approval History* (Gambar 3.54b) menampilkan riwayat dengan detail (Gambar 3.55b).

G Tampilan Modul Absences

Modul *Absences* memiliki lima menu utama: *Absences Request*, *Request Status*, *Cancellation*, *History*, dan *Approval*. Berikut adalah tampilan masing-masing menu sesuai dengan urutannya dalam aplikasi.

The image displays two screenshots of the 'Request' form in the Absences module. Both screenshots show a blue header with a back arrow, a plus icon, and the title 'Request' with the subtitle 'Kelola request'.

(a) Absences Request Form: This screenshot shows the form with the following fields:

- Absence Data:**
 - Superior 1: [Redacted] INDRA [Redacted]
 - NIK Superior 1: [Redacted]
 - Date Request: 23 Des 2025
 - Absence Type: Worship Absence (dropdown)
- ABSENCE DATA:**
 - PLEASE INPUT NUMBER OF ABSENCE
 - Absence Type: Worship Absence
 - Number of Absence: 1 Day (dropdown)

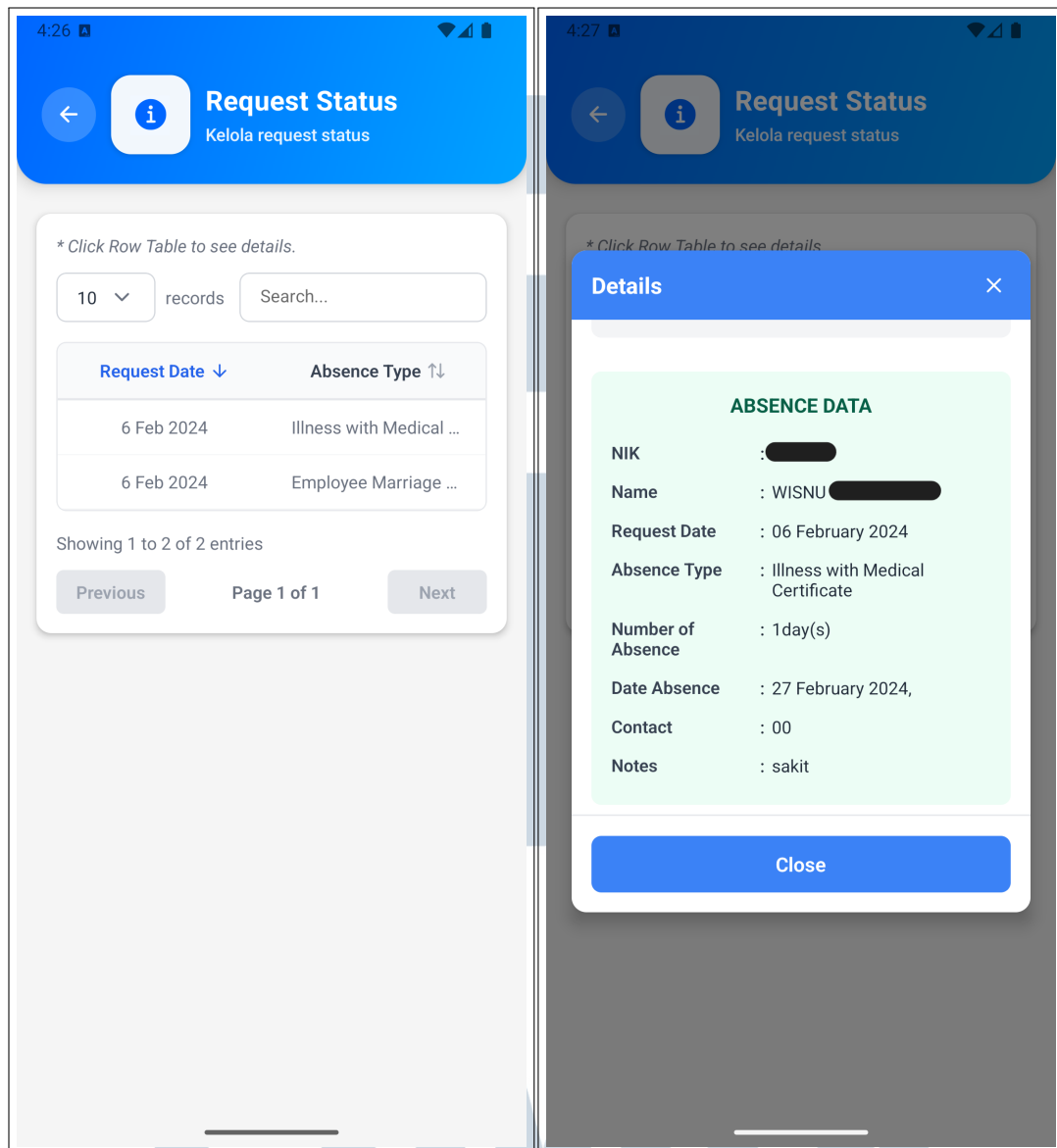
(b) Form Validation Error: This screenshot shows the form with a red message box indicating a validation error:

- Absence Type:** Worship Absence
- DATES:**
 - Date Request: 23 Desember 2025
 - Day 1: 29 Desember 2025
- Contact Number:** 082392392
- Notes:** Testing laporan magang
- Validation Error:** Mohon maaf untuk sementara waktu request hanya bisa dilakukan melalui HR Portal Web
- SUBMIT:** Button

(a) Absences Request Form

(b) Form Validation Error

Gambar 3.56. Tampilan Modul Absences - Request Form

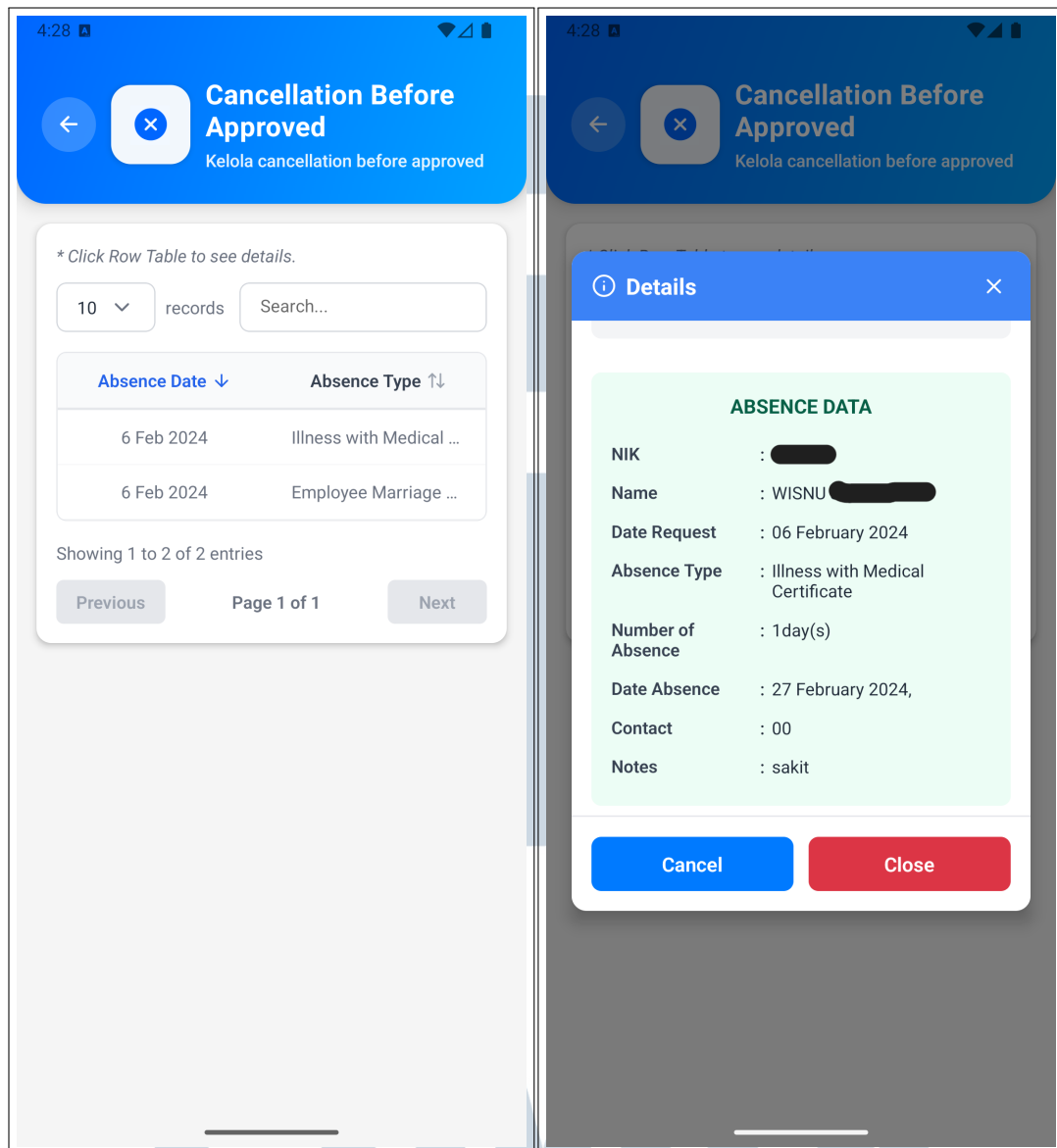


(a) Request Status

(b) Detail Request Status

Gambar 3.57. Tampilan Modul Absences - Request Status

Gambar 3.56 dan 3.57 menampilkan dua menu pertama dalam modul Absences. *Absences Request Form* (Gambar 3.56a) menyediakan field untuk jenis ketidakhadiran, tanggal, dan alasan dengan tampilan validasi error (Gambar 3.56b). *Request Status* (Gambar 3.57a) menampilkan daftar pengajuan dengan detail (Gambar 3.57b).

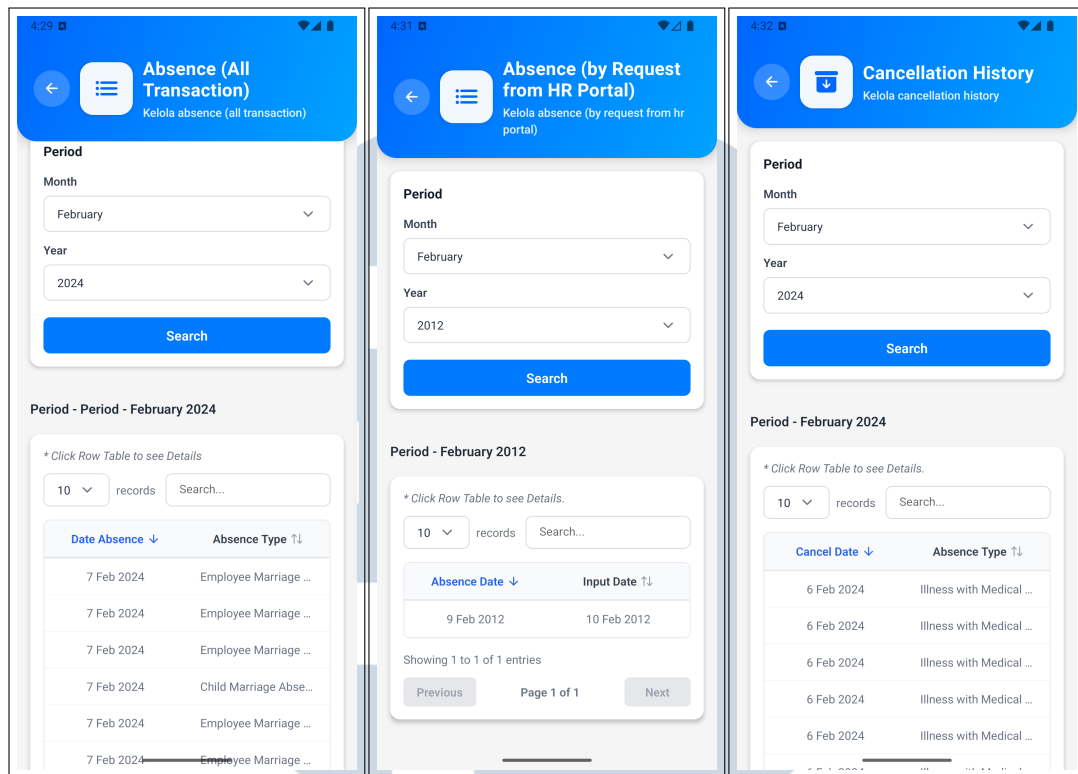


(a) Cancellation Before Approved

(b) Detail Cancellation

Gambar 3.58. Tampilan Modul Absences - Cancellation

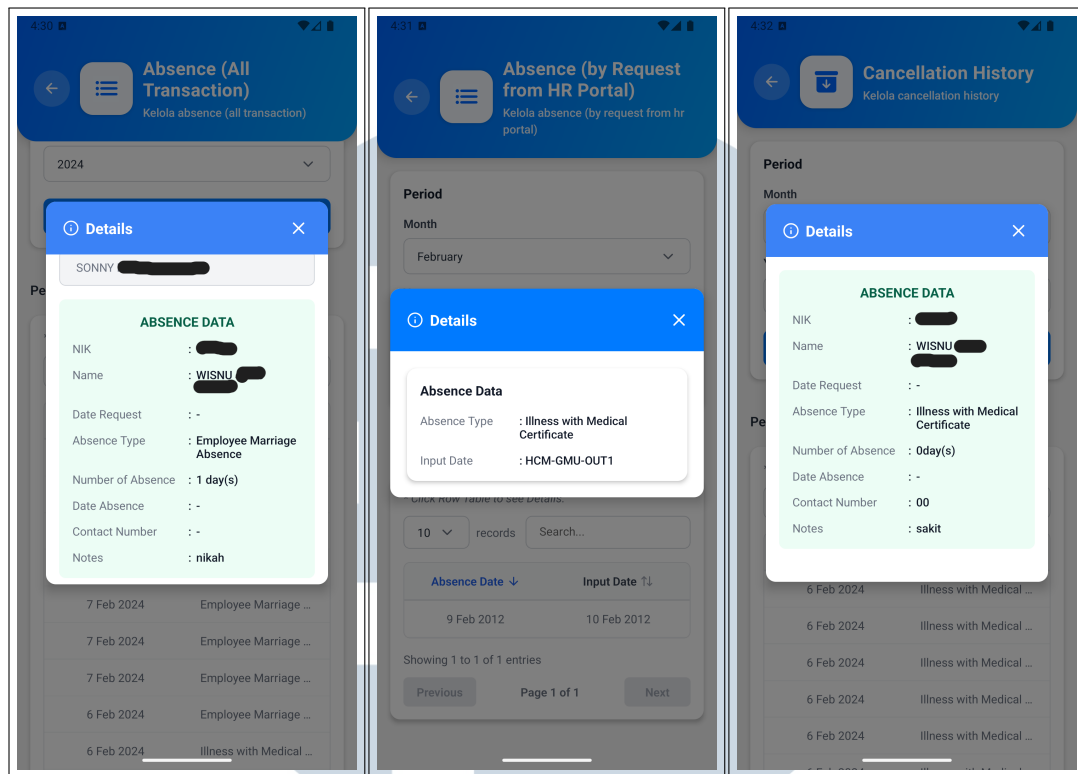
Gambar 3.58 menampilkan menu *Cancellation* untuk membatalkan pengajuan sebelum disetujui. Daftar pembatalan (Gambar 3.58a) menampilkan pengajuan yang dapat dibatalkan dengan detail (Gambar 3.58b).



(a) All Transaction (b) HR Portal (c) Cancellation History

Gambar 3.59. Tampilan Modul Absences - History

UMN
 UNIVERSITAS
 MULTIMEDIA
 NUSANTARA



(a) Detail All Transaction

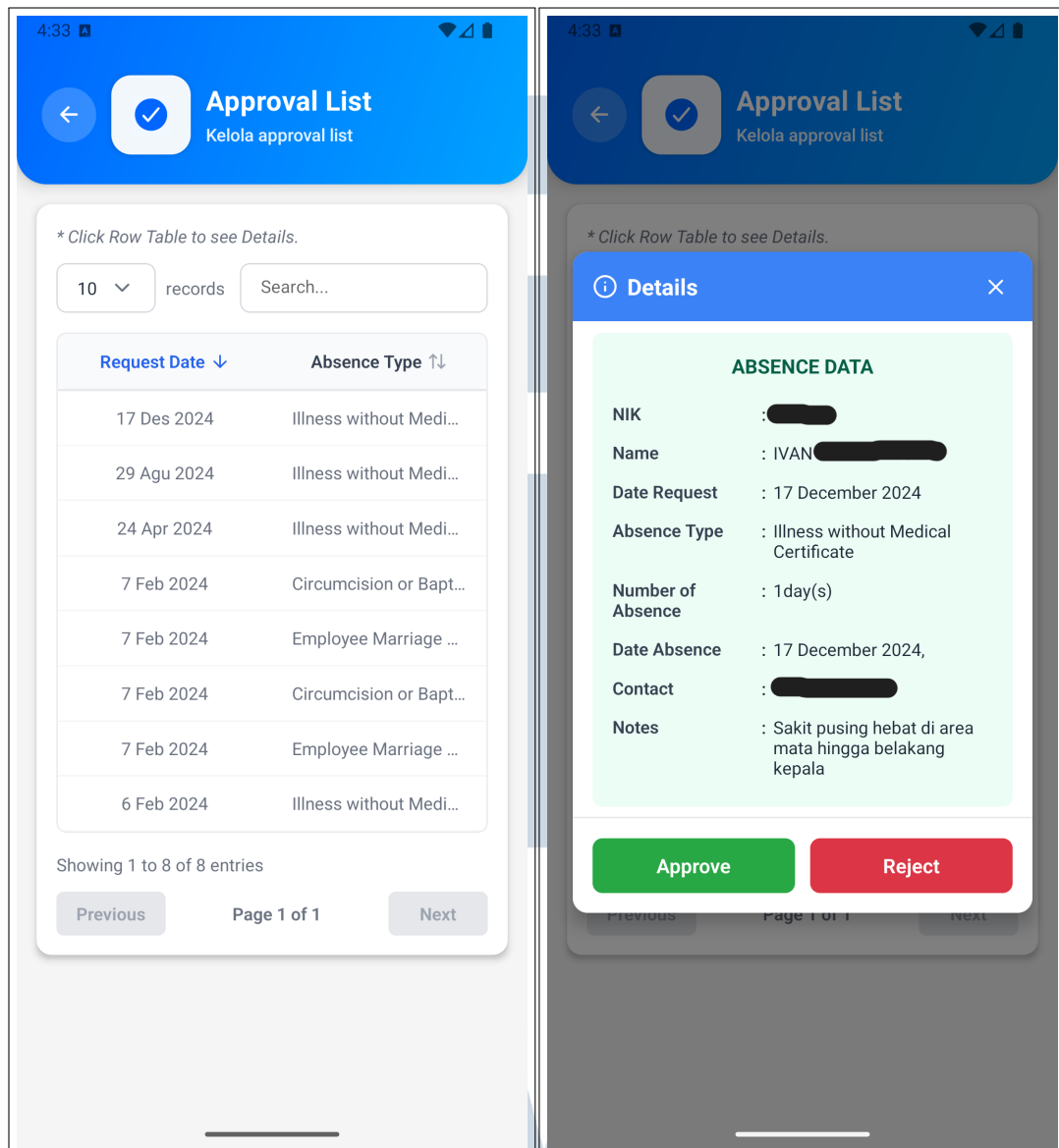
(b) Detail HR Portal

(c) Detail Cancellation History

Gambar 3.60. Tampilan Modul Absences - Detail History

Gambar 3.59 dan 3.60 menampilkan menu *History* yang memiliki tiga sub-menu. *All Transaction* (Gambar 3.59a) menampilkan semua riwayat dengan detail (Gambar 3.60a). *HR Portal* (Gambar 3.59b) menampilkan riwayat dengan detail (Gambar 3.60b). *Cancellation History* (Gambar 3.59c) menampilkan riwayat pembatalan dengan detail (Gambar 3.60c).

UNIVERSITAS
MULTIMEDIA
NUSANTARA

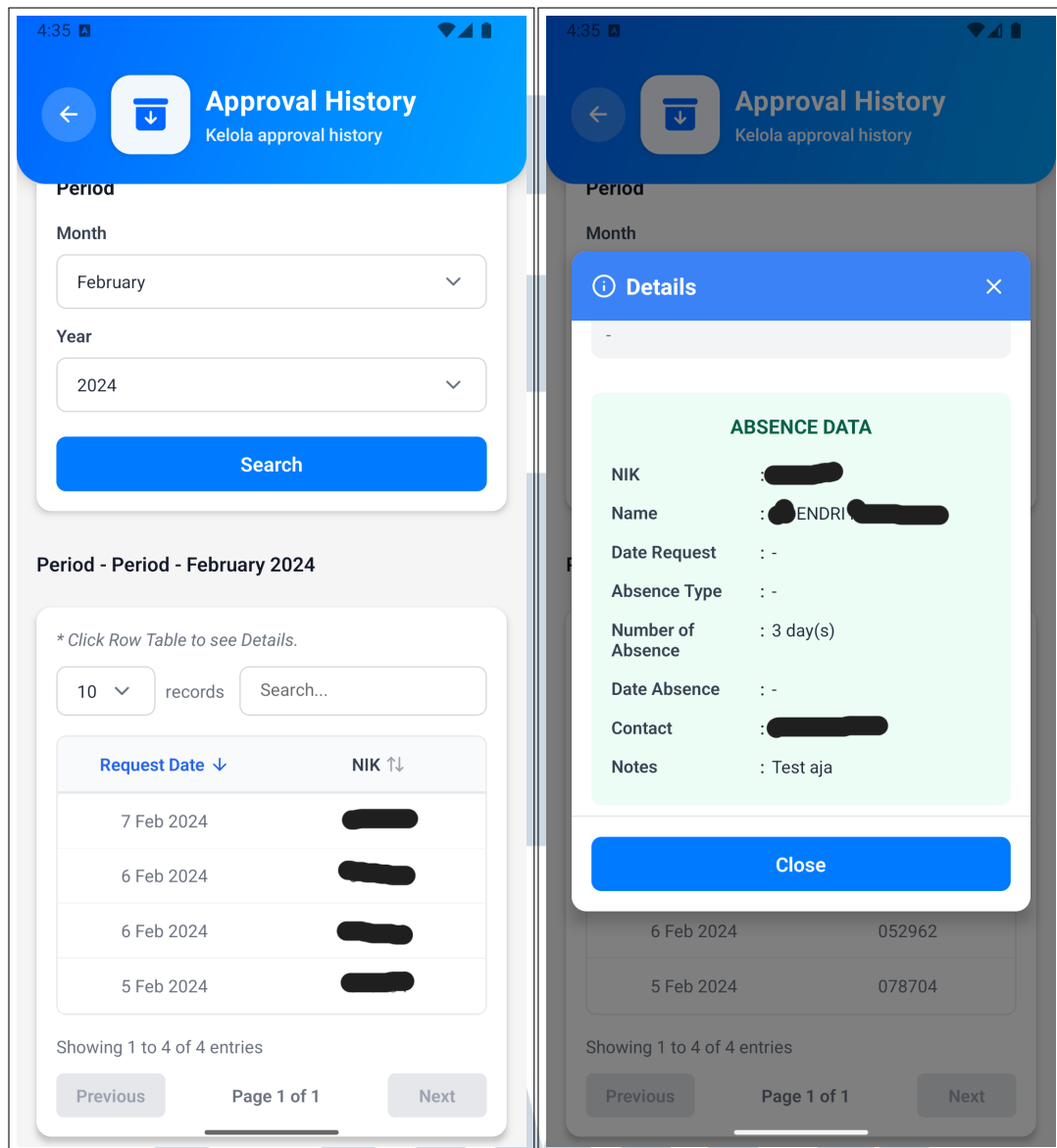


(a) Approval List

(b) Detail Approval

Gambar 3.61. Tampilan Modul Absences - Approval List

UNIVERSITAS
MULTIMEDIA
NUSANTARA



(a) Approval History

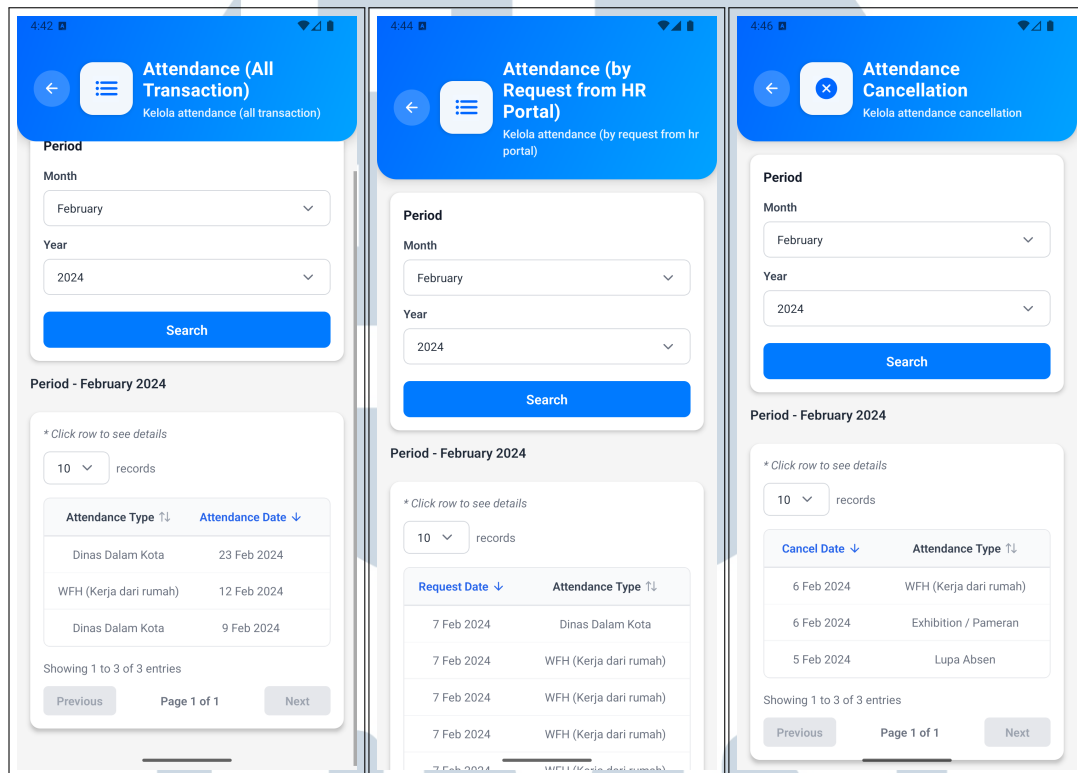
(b) Detail Approval History

Gambar 3.62. Tampilan Modul Absences - Approval History

Gambar 3.61 dan 3.62 menampilkan menu *Approval* yang memiliki dua sub-menu. *Approval List* (Gambar 3.61a) menampilkan daftar pengajuan bawahan yang menunggu persetujuan dengan detail (Gambar 3.61b). *Approval History* (Gambar 3.62a) menampilkan riwayat persetujuan dengan detail (Gambar 3.62b).

H Tampilan Modul Attendance

Modul *Attendance* memiliki empat menu utama: *History*, *Request*, *Request Status*, dan *Approval*. Berikut adalah tampilan masing-masing menu sesuai dengan urutannya dalam aplikasi.



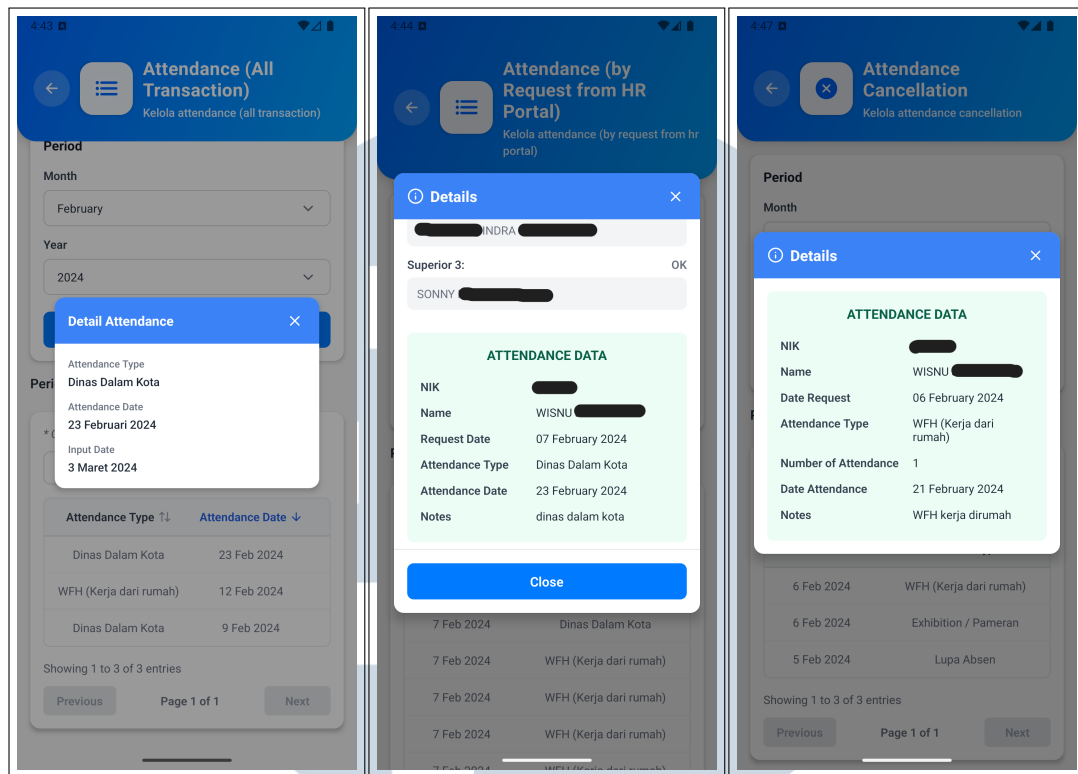
(a) All Transaction

(b) HR Portal

(c) Cancellation History

Gambar 3.63. Tampilan Modul Attendance - History

UNIVERSITAS
MULTIMEDIA
NUSANTARA



(a) Detail All Transaction

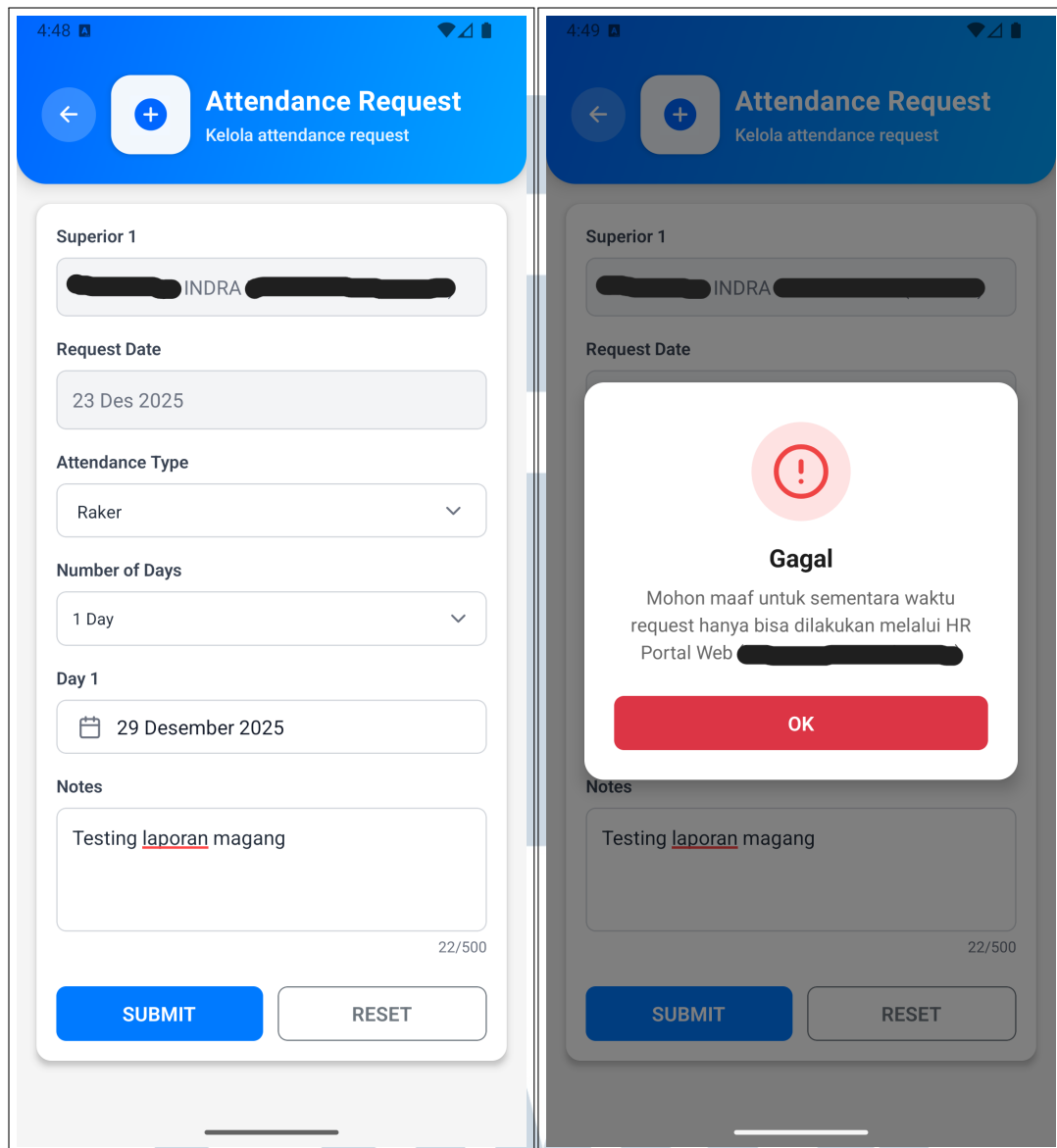
(b) Detail HR Portal

(c) Detail Cancellation History

Gambar 3.64. Tampilan Modul Attendance - Detail History

Gambar 3.63 dan 3.64 menampilkan menu *History*. *All Transaction* (Gambar 3.63a) menampilkan riwayat dengan detail (Gambar 3.64a). *HR Portal* (Gambar 3.63b) menampilkan riwayat dengan detail (Gambar 3.64b). *Cancellation History* (Gambar 3.63c) menampilkan riwayat pembatalan dengan detail (Gambar 3.64c).

UNIVERSITAS
MULTIMEDIA
NUSANTARA

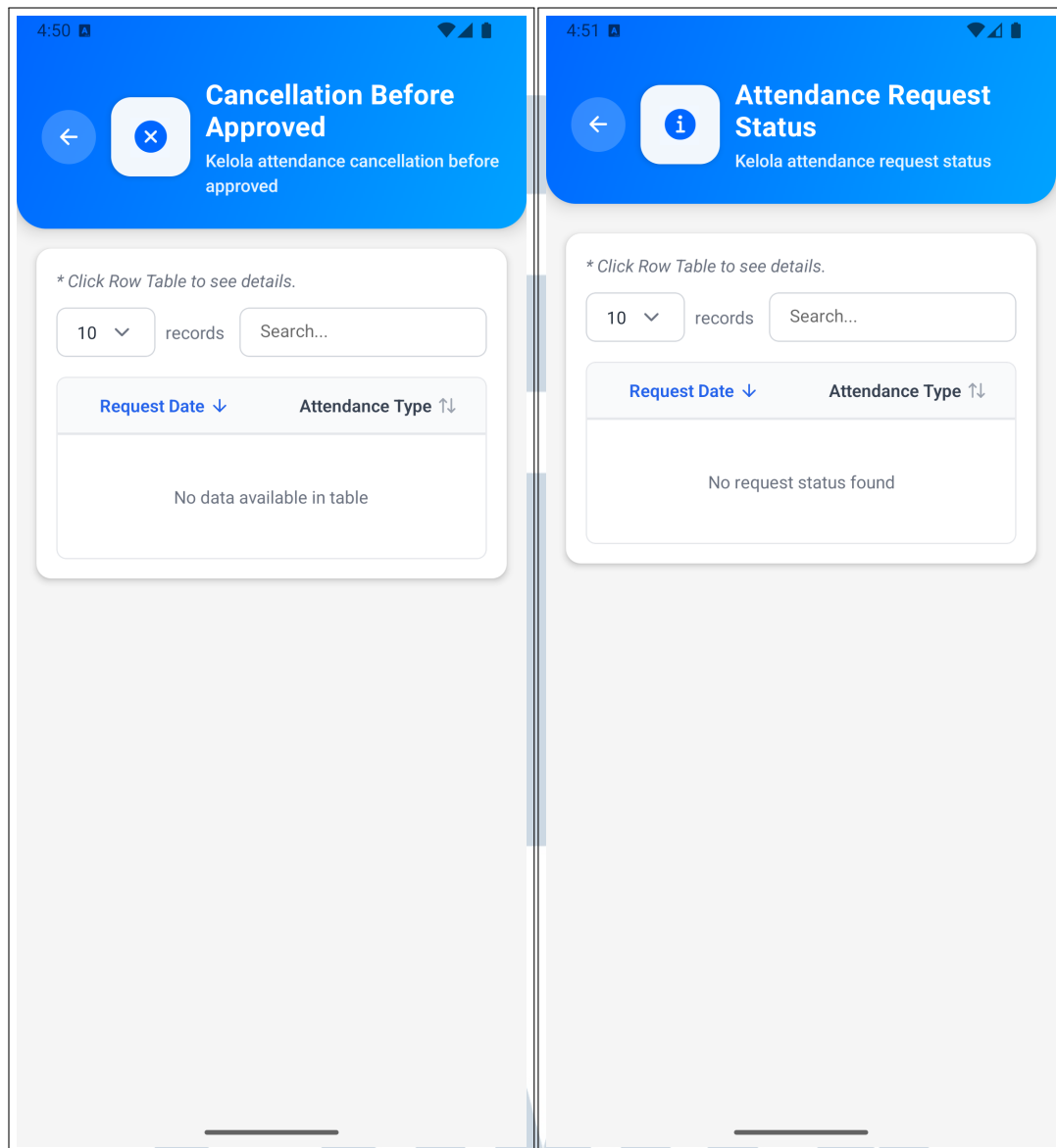


(a) Attendance Request

(b) Form Validation Error

Gambar 3.65. Tampilan Modul Attendance - Request Form

UNIVERSITAS
MULTIMEDIA
NUSANTARA

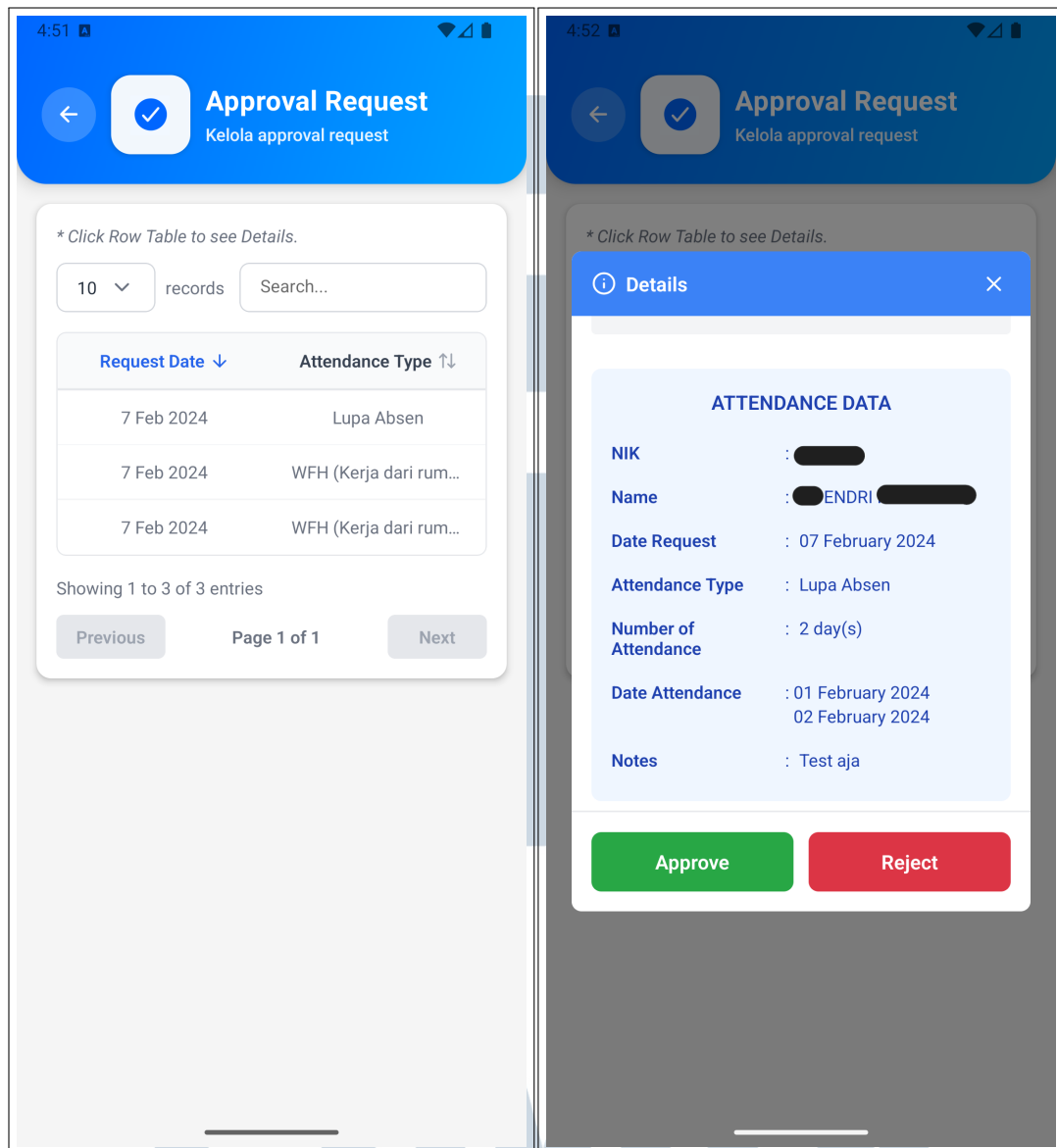


(a) Cancellation Before Approve

(b) Request Status

Gambar 3.66. Tampilan Modul Attendance - Cancellation dan Request Status

Gambar 3.65 dan 3.66 menampilkan menu *Request* dan *Request Status*. *Attendance Request* (Gambar 3.65a) untuk mengajukan koreksi dengan tampilan validasi error (Gambar 3.65b). *Cancellation Before Approve* (Gambar 3.66a) untuk membatalkan pengajuan. *Request Status* (Gambar 3.66b) memantau status pengajuan.

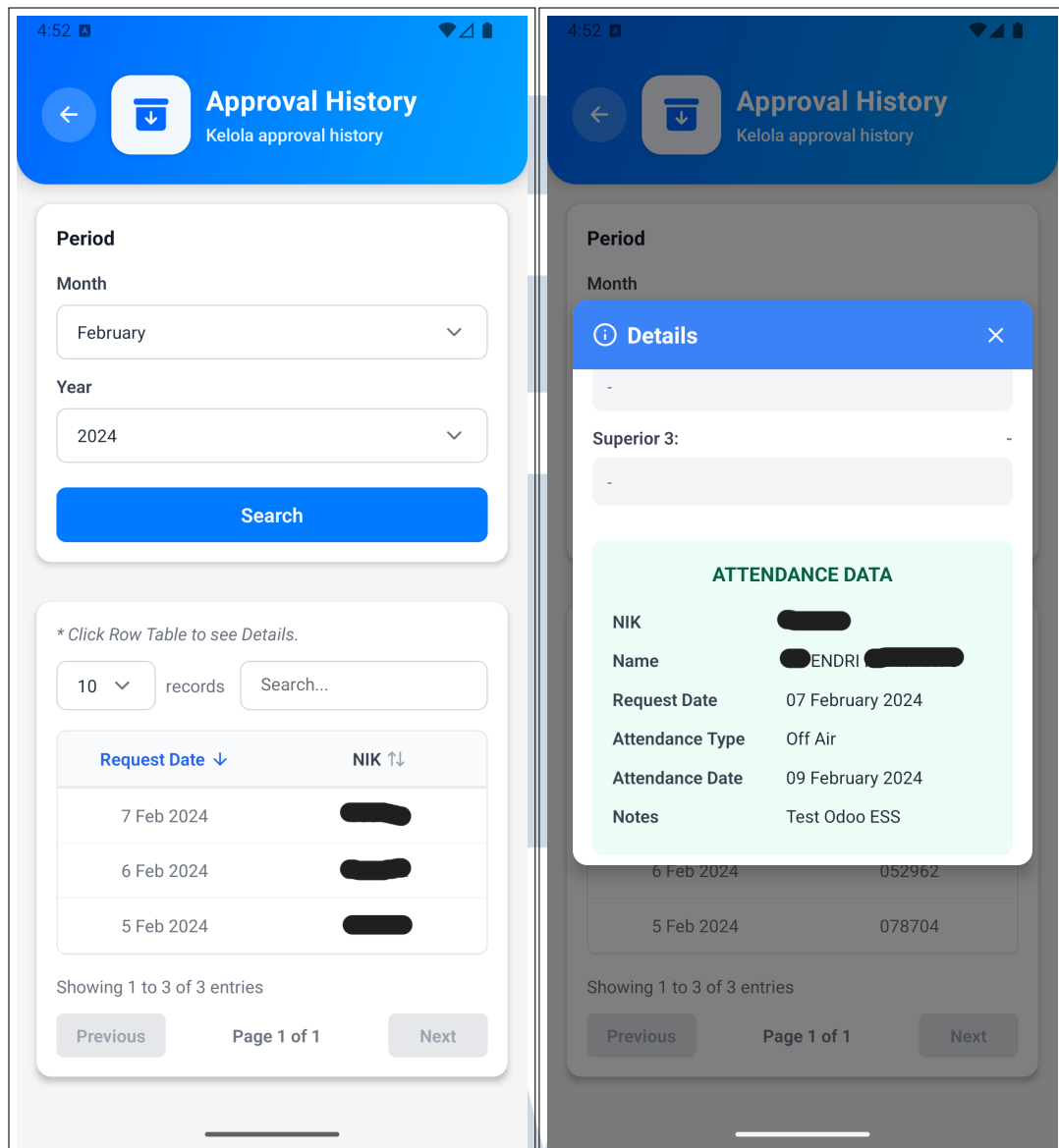


(a) Approval Request

(b) Detail Approval Request

Gambar 3.67. Tampilan Modul Attendance - Approval Request

UNIVERSITAS
MULTIMEDIA
NUSANTARA



(a) Approval History

(b) Detail Approval History

Gambar 3.68. Tampilan Modul Attendance - Approval History

Gambar 3.67 dan 3.68 menampilkan menu *Approval* yang memiliki dua sub-menu. *Approval Request* (Gambar 3.67a) menampilkan daftar pengajuan bawahan dengan detail (Gambar 3.67b) untuk memberikan keputusan. *Approval History* (Gambar 3.68a) menampilkan riwayat persetujuan dengan detail (Gambar 3.68b).

I Tampilan Modul SKKL

Modul SKKL (*Surat Keterangan Kerja Lembur*) memiliki empat menu: *Create SKKL*, *SKKL Status Assignment*, *SKKL Status*, dan *SKKL Rekap*. Berikut adalah tampilan masing-masing menu sesuai dengan urutannya dalam aplikasi.

4:56

← + Create SKKL
Pilih Employee & Superior

ASSIGN TO
Number of Employee
-SELECT-

SUPERIOR TO APPROVE
Number of Superior
-SELECT-

NEXT

4:56

← SKKL Status Assignment
Kelola SKKL status assignment

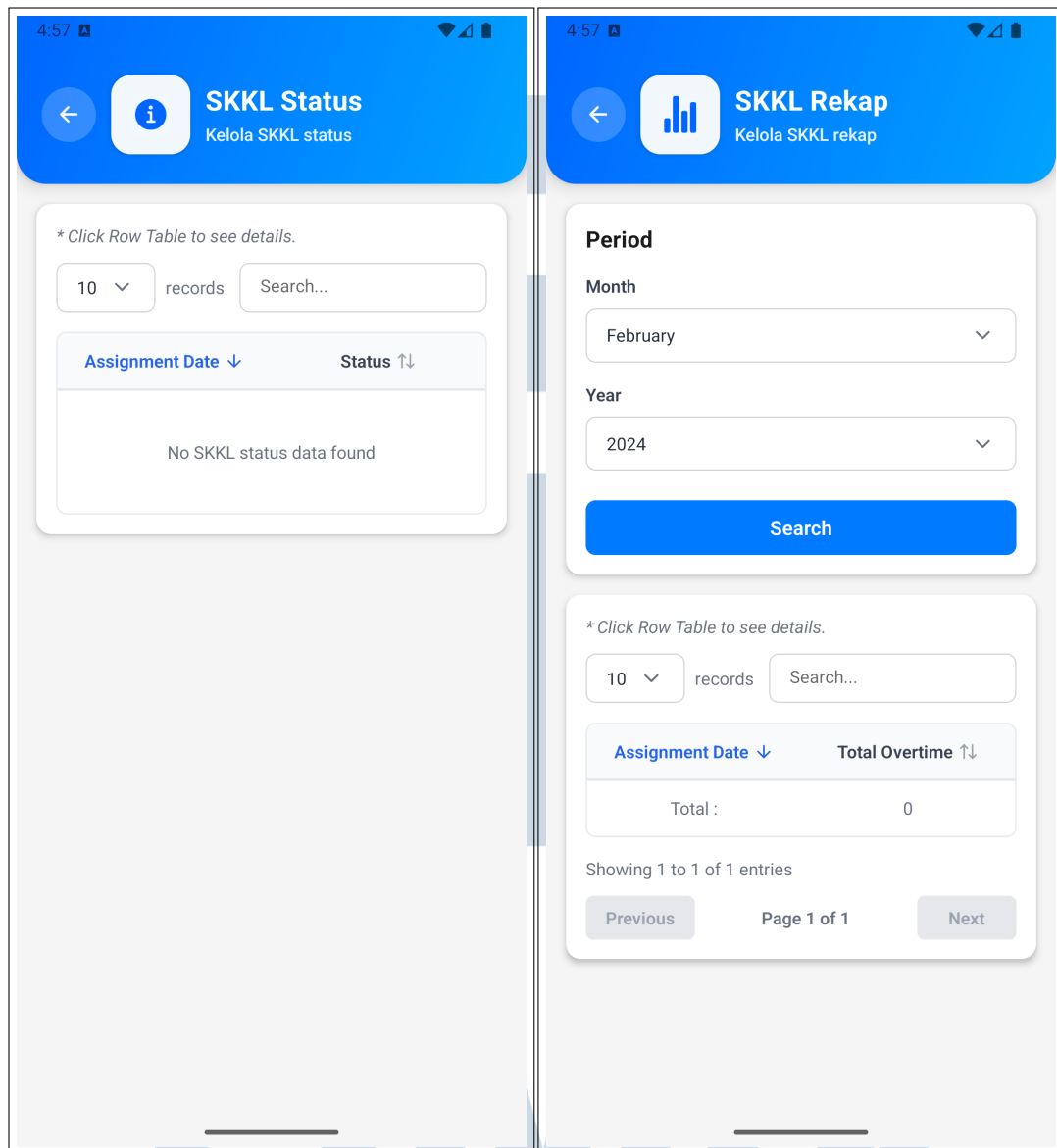
Period
Month
February
Year
2024
Search

10 records Search...
Sub Ordinate ↑↓ Assignment Date ↓
No SKKL status assignment data found for selected period

(a) Create SKKL

(b) SKKL Status Assignment

Gambar 3.69. Tampilan Modul SKKL - Create dan Status Assignment



(a) SKKL Status

(b) SKKL Rekap

Gambar 3.70. Tampilan Modul SKKL - Status dan Rekap

Gambar 3.69 dan 3.70 menampilkan antarmuka modul SKKL. *Create SKKL* (Gambar 3.69a) untuk pembuatan surat lembur baru dengan memilih karyawan dan jadwal. *SKKL Status Assignment* (Gambar 3.69b) menampilkan daftar penugasan lembur. *SKKL Status* (Gambar 3.70a) memantau status pengajuan surat lembur. *SKKL Rekap* (Gambar 3.70b) menampilkan rekapitulasi data lembur berdasarkan periode.

3.4 Kendala dan Solusi

Selama menjalani kerja magang, beberapa kendala teknis dan operasional muncul dalam proses pengembangan aplikasi HR Mobile Portal. Bagian ini mendokumentasikan kendala-kendala tersebut beserta solusi yang diterapkan.

3.4.1 Kendala

1. **Integrasi dengan *Backend* PHP yang mengembalikan HTML.** *Backend* sistem HR Portal yang sudah ada menggunakan PHP dan mengembalikan respons dalam format HTML, bukan JSON yang lazim digunakan dalam pengembangan aplikasi *mobile* modern. Kondisi ini memerlukan pendekatan khusus dalam mengekstrak dan memproses data dari respons server.
2. **Struktur data yang bervariasi antar *endpoint*.** Setiap *endpoint* memiliki struktur HTML yang berbeda-beda, sehingga diperlukan *parser* yang spesifik untuk masing-masing *endpoint*. Hal ini menambah kompleksitas pengembangan dan memerlukan waktu lebih untuk memahami struktur data dari setiap halaman.
3. **Dokumentasi API yang terbatas.** Karena sistem *backend* merupakan *legacy system* yang sudah lama beroperasi, dokumentasi API tidak selalu tersedia atau tidak *up-to-date*. Diperlukan eksplorasi langsung terhadap respons server untuk memahami struktur dan perilaku masing-masing *endpoint*.
4. **Perbedaan perilaku antar platform.** Meskipun React Native mendukung pengembangan *cross-platform*, terdapat beberapa perbedaan perilaku antara Android dan iOS yang memerlukan penyesuaian khusus, terutama dalam hal tampilan dan interaksi pengguna.

3.4.2 Solusi

1. **Implementasi *Parser* menggunakan *node-html-parser*.** Dikembangkan *parser functions* yang memanfaatkan *library node-html-parser* untuk mengekstrak data dari elemen HTML dan mengkonversinya menjadi objek TypeScript yang terstruktur. Pendekatan ini memungkinkan pengolahan data dari respons HTML secara sistematis dan konsisten.

2. **Pembuatan *Parser* modular untuk setiap *endpoint*.** Setiap *endpoint* memiliki *parser* tersendiri yang didefinisikan dalam *file* terpisah di dalam folder *services*. Struktur ini memudahkan pemeliharaan dan pengujian masing-masing *parser* secara independen.
3. **Eksplorasi langsung dan dokumentasi internal.** Dilakukan eksplorasi langsung terhadap respons server menggunakan *tools* seperti Postman dan *browser developer tools*. Temuan-temuan didokumentasikan dalam *file* internal untuk memudahkan pengembangan selanjutnya dan menjadi referensi bagi anggota tim lainnya.
4. **Penggunaan komponen dan styling yang *platform-agnostic*.** Dipilih komponen-komponen dari NativeWind dan React Native yang memiliki perilaku konsisten di kedua platform. Untuk kasus-kasus yang memerlukan penyesuaian, diimplementasikan *conditional styling* berdasarkan platform yang terdeteksi.

