

BAB III

PELAKSANAAN MBKM

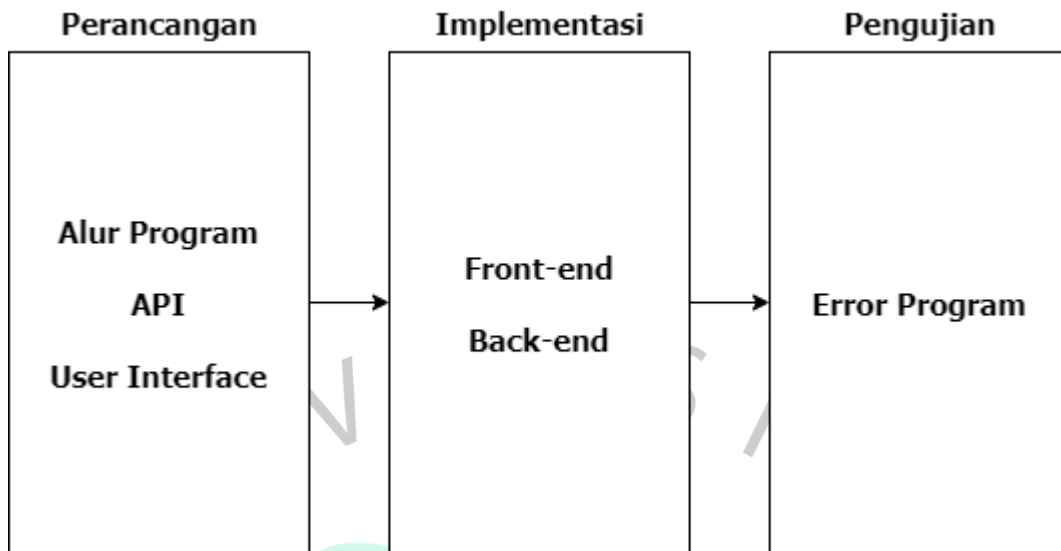
3.1 Bidang Kerja

Dalam melaksanakan kegiatan MBKM yang dilaksanakan di PT Bank Rakyat Indonesia (Persero) Tbk, praktikan ditempatkan di bagian Tim IT SAU yang termasuk ke dalam Human Capital Development Division sebagai Programmer. Salah satu proyek yang sedang dikembangkan oleh praktikan yaitu aplikasi MySAU dimana aplikasi ini merupakan portal sistem informasi pagi pekerja di SAU dan salah satu fitur aplikasi ini yang digunakan sehari-hari adalah fitur presensi harian untuk pekerja.

Aplikasi MySAU merupakan aplikasi berbasis web yang dikembangkan menggunakan bahasa pemrograman PHP dengan framework CodeIgniter 4. Kemudian praktikan mengembangkan aplikasi tersebut menjadi berbasis mobile menggunakan framework React Native sebagaimana sudah dipelajari diperkuliahan sebelumnya, dengan demikian bisa disebutkan bahwa pekerjaan ini relevan dengan studi yang diampu oleh praktikan sebagai mahasiswa Informatika. Tidak hanya itu, praktikan juga mendapatkan ilmu baru seperti pemahaman perancangan aplikasi, library dan bahasa pemrograman yang belum pernah digunakan selama di perkuliahan serta bagaimana mencari solusi untuk memecahkan suatu masalah menggunakan bahasa pemrograman.

3.2 Pelaksanaan Kerja

Pada pelaksanaan kerja untuk membangun aplikasi berbasis mobile, praktikan mempunyai tahap-tahap yang dilaksanakan secara sistematis dan terstruktur agar hasil yang didapatkan dapat maksimal. Aplikasi dikembangkan dengan beberapa tahap mulai dari analisis dan perancangan yang terbagi menjadi perancangan alur program, perancangan API untuk koneksi ke database yang sebelumnya sudah ada dan digunakan oleh aplikasi MySAU berbasis web, dan perancangan user interface. Kemudian tahap selanjutnya adalah implementasi front-end dan back-end. Terakhir adalah tahap pengujian untuk menguji fungsi aplikasi agar tidak menemukan celah error program dan aplikasi dapat berjalan dengan baik.

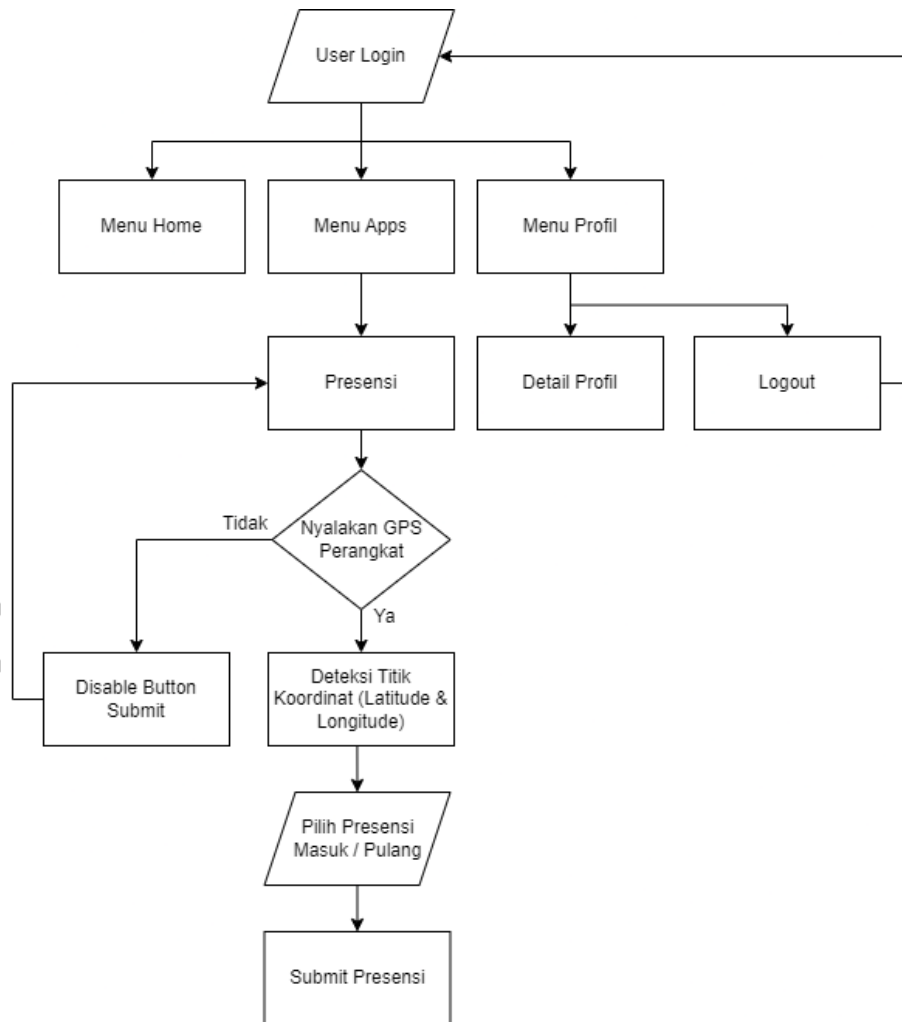


Gambar 3. 1 Tahap Pelaksanaan Kerja

Gambar tahap pelaksanaan kerja di atas menurut praktikan adalah tahap pelaksanaan yang sistematis dan realistis dalam pelaksanaan yang ada, dari semua tahap pelaksanaan tersebut praktikan diawasi dan diberikan arahan oleh pembimbing eksternal yang ada di PT Bank Rakyat Indonesia. Untuk lebih rinci akan dijelaskan lebih lanjut sebagai berikut. Timeline dari pelaksanaan kegiatan MBKM untuk proyek aplikasi MySAU berbasis mobile ini dapat dilihat pada table di bawah ini.

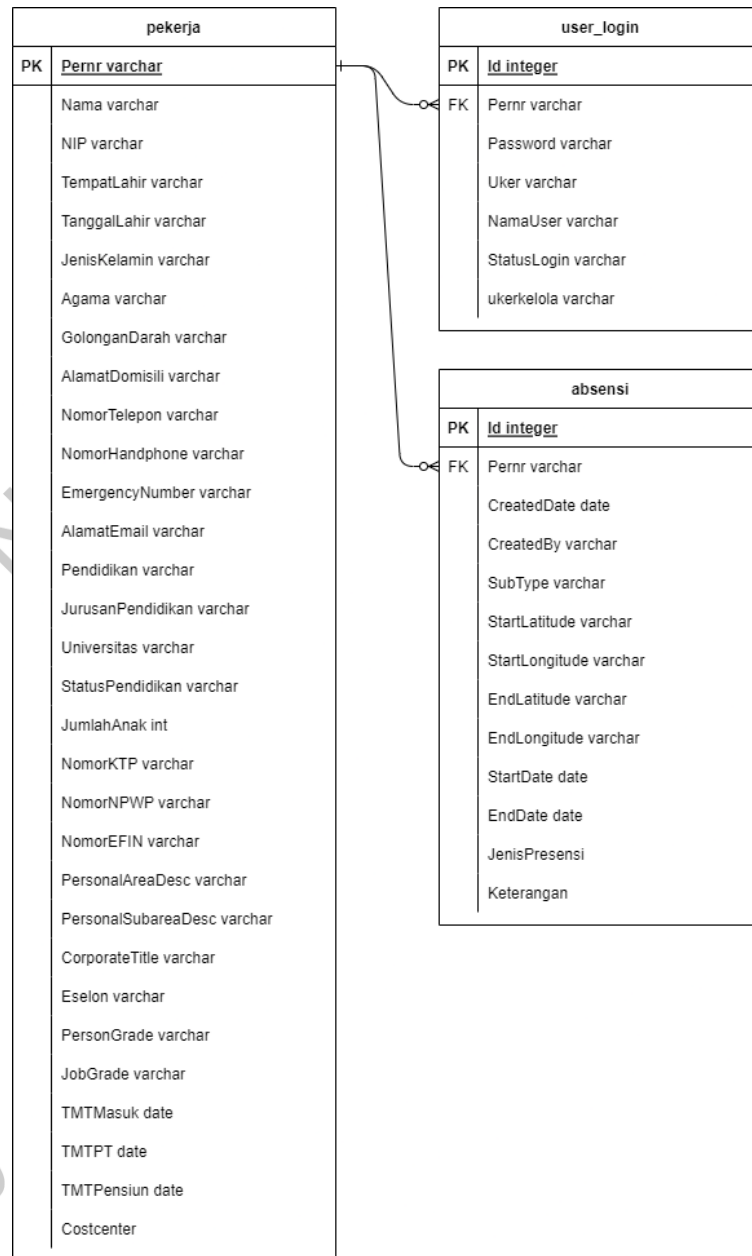
Tabel 3. 1 Linimasa Kegiatan

Kegiatan	Agustus				September				Oktober		
	M1	M2	M3	M4	M1	M2	M3	M4	M1	M2	M3
Perancangan Alur Program	■	■									
Perancangan API			■	■							
Perancangan User Interface				■	■						
Implementasi Front-End					■	■	■	■	■		
Implementasi Back-end							■	■	■	■	■



Gambar 3. 2 Alur Program Aplikasi MySAU

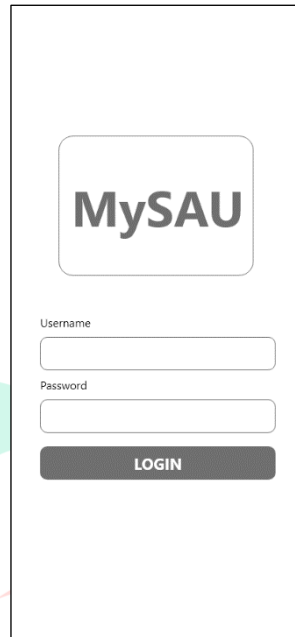
Setelah membuat alur program, praktikan mulai merancang database atau rancangan *Entity Relationship Diagram (ERD)*. ERD adalah model atau rancangan untuk membuat database, supaya lebih mudah dalam menggambarkan data yang memiliki hubungan atau relasi dalam bentuk sebuah desain. Dengan adanya ERD, maka sistem database yang terbentuk dapat digambarkan dengan lebih terstruktur dan terlihat rapi (Adani, 2021). Adapun Entity Relationship Diagram yang dibuat oleh praktikan berikut ini.



Gambar 3. 3 Entity Relationship Diagram

Gambar 3.3 merupakan rancangan ERD yang dibuat oleh praktikan. Tabel *pekerja* berfungsi untuk menyimpan data identitas diri, data pekerjaan, dan data pendidikan dari setiap pengguna. Kemudian tabel *user_login* tabel untuk menampung user login dan password dari setiap pengguna. Terakhir, tabel *absensi* menyimpan data kehadiran pekerja setiap harinya seperti data tanggal absensi, titik koordinat lokasi, dan jenis presensi baik presensi masuk maupun pulang.

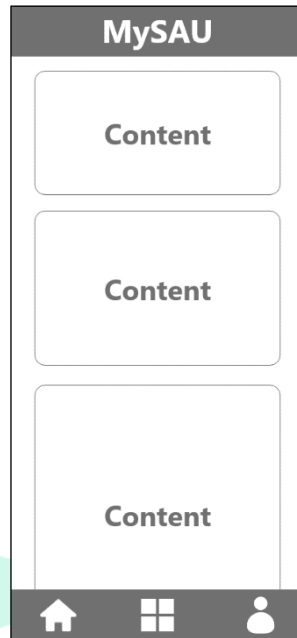
Dalam membangun sebuah aplikasi tentunya diperlukan *user interface* agar pengguna dapat menggunakan aplikasi dengan mudah. Praktikan membuat *mockup front-end* aplikasi sebelum diimplementasikan ke dalam program.



The image shows a vertical rectangular mockup of a login page. At the top center, there is a rounded square containing the text "MySAU". Below this, the word "Username" is followed by a white input field with a light gray border. Underneath, the word "Password" is followed by another white input field with a light gray border. At the bottom of the form area, there is a dark gray button with the word "LOGIN" in white capital letters.

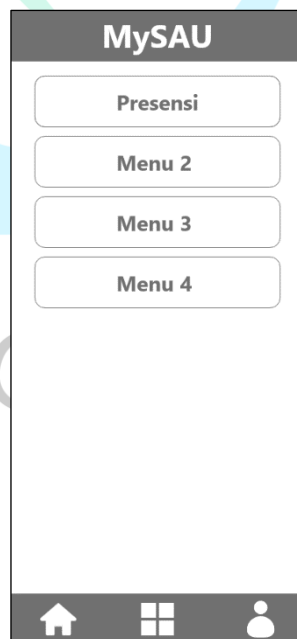
Gambar 3. 4 Mockup Halaman Login

Gambar 3.4 merupakan desain mockup halaman login. Halaman pertama pada saat membuka aplikasi adalah halaman login yang berguna untuk mengakses fitur pada aplikasi tersebut. Pengguna nantinya akan diminta memasukkan *username* dan *password*.



Gambar 3. 5 Mockup Halaman Home

Pada gambar 3.5 ditunjukkan mockup halaman home. Halaman Home nantinya akan menampilkan berbagai konten informasi untuk para pekerja SAU. Terlihat di bagian bawah terdapat tiga menu untuk berpindah ke Halaman Apps dan Profil. Menu home ini bisa diakses setelah pengguna melakukan login pada aplikasi tersebut.



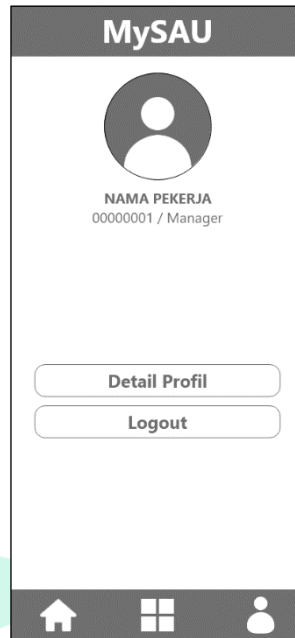
Gambar 3. 6 Mockup Halaman Apps

Gambar 3.6 merupakan mockup Halaman Apps. Halaman Apps terdiri dari beberapa fitur aplikasi pendukung untuk bekerja yang akan memudahkan para pekerja. Untuk pengembangan saat ini hanya ada fitur presensi, namun untuk pengembangan kedepannya tidak menutup kemungkinan akan ada fitur-fitur lainnya.



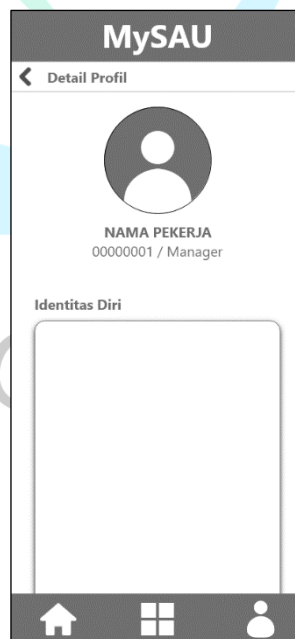
Gambar 3. 7 Mockup Halaman Presensi

Pada Gambar 3.7 menampilkan mockup Halaman Presensi. Halaman Presensi terdiri dari informasi waktu dan titik koordinat serta tombol untuk melakukan presensi harian. Pada pilihan jenis presensi terdapat 2 jenis presensi yaitu presensi masuk dan presensi pulang. Nantinya pekerja dapat melakukan presensi harian dengan mudah pada fitur ini.



Gambar 3. 8 Mockup Halaman Profil

Gambar 3.8 adalah mockup tampilan Halaman Profil. Tampilan Halaman Profil akan menampilkan informasi pengguna seperti nama lengkap, nomor pekerja, dan jabatan. Pada halaman ini juga terdapat dua menu yaitu menu Detail Profil untuk melihat profil detail pengguna yang sedang login tersebut. Kemudian menu kedua adalah menu Logout untuk keluar dari akun pada aplikasi tersebut.



Gambar 3. 9 Mockup Halaman Detail Profil

Gambar 3.9 merupakan mockup tampilan halaman Detail Profil. Seperti yang sudah dijelaskan sebelumnya, bahwa Halaman Detail Profil memuat informasi pengguna meliputi identitas diri, profil pekerja utama, riwayat pendidikan dan data keluarga.

3.2.2 Implementasi

Dalam implementasi praktikan untuk membuat sebuah aplikasi berbasis mobile, berikut adalah spesifikasi perangkat lunak yang digunakan oleh praktikan.

Tabel 3. 2 Spesifikasi Perangkat Lunak

No.	Nama Perangkat Lunak	Spesifikasi Perangkat Lunak
1.	Operating System	Windows 11
2.	Local Web Server	Apache 2.4
3.	Frontend Framework	React Native 0.70
4.	API Framework	Codelgniter 4
5.	Database	MySQL 5.7
6.	IDE	Visual Studio Code

Adapun spesifikasi perangkat keras yang digunakan praktikan dalam menunjang pengerjaan aplikasi ini. Berikut adalah tabel spesifikasi perangkat keras untuk pembuatan aplikasi.

Tabel 3. 3 Spesifikasi Perangkat Keras

No.	Nama Perangkat Keras	Spesifikasi Perangkat Keras
1.	Processor	AMD Ryzen 7 6800HS
2.	VGA	Radeon Graphics
3.	Memory	16 GB

Selain itu, adapun spesifikasi perangkat *mobile* yang digunakan praktikan untuk melakukan pengujian dalam pembuatan aplikasi yaitu sebagai berikut.

Tabel 3. 4 Spesifikasi Perangkat Pengujian

No.	Nama Komponen/Sistem	Spesifikasi Komponen/Sistem
1.	Operating System	Android Version 12
2.	API Level	API 31
3.	CPU	Qualcomm Snapdragon 855+
4.	Memory	8 GB

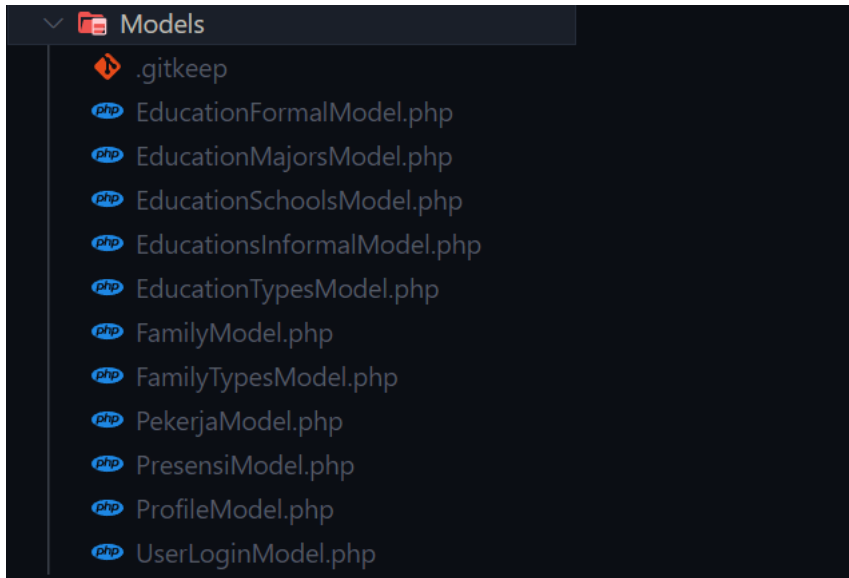
Penyimpanan data pada aplikasi ini menggunakan database MySQL, praktikan membangun database aplikasi ini pada server development PT SAU terlebih dahulu. Selain pada server development, praktikan juga membuat database cadangan database lokal yang ada pada laptop sebagai *backup* jika terjadi suatu masalah pada server.

Name	Rows	Size	Created	Updated	Engine	Comment	Type
kelurahan	80.164	4,5 MiB	2022-10-13 16:11:28		InnoDB		Table
kecamatan	7.377	352,0 KiB	2022-10-13 16:11:28		InnoDB		Table
education_sch...	4.703	448,0 KiB	2022-10-13 16:11:28		InnoDB		Table
education_maj...	2.440	176,0 KiB	2022-10-13 16:11:28		InnoDB		Table
kota/kabupaten	514	64,0 KiB	2022-10-13 16:11:29		InnoDB		Table
accessmenu	154	16,0 KiB	2022-10-13 16:11:27		InnoDB		Table
master_position	123	16,0 KiB	2022-10-13 16:11:29		InnoDB		Table
employee_sub...	71	16,0 KiB	2022-10-13 16:11:28		InnoDB	T503K	Table
menu	37	16,0 KiB	2022-10-13 16:11:29		InnoDB		Table
absence_types	35	16,0 KiB	2022-10-13 16:11:27		InnoDB	T554T	Table
provinsi	34	16,0 KiB	2022-10-13 16:11:31		InnoDB		Table
job_family	18	16,0 KiB	2022-10-13 16:11:28		InnoDB		Table
education_types	12	16,0 KiB	2022-10-13 16:11:28		InnoDB		Table
rekap_presensi	10	16,0 KiB	2022-10-13 16:11:31		InnoDB		Table
corporate_title	9	16,0 KiB	2022-10-13 16:11:27		InnoDB		Table
mapping_organ...	9	16,0 KiB	2022-10-13 16:11:29		InnoDB		Table
organization_s...	9	16,0 KiB	2022-10-13 16:11:31		InnoDB		Table
family_types	7	16,0 KiB	2022-10-13 16:11:28		InnoDB		Table
position_texts	7	16,0 KiB	2022-10-13 16:11:31		InnoDB		Table
admin	6	16,0 KiB	2022-10-13 16:11:27		InnoDB		Table
leveluser	6	16,0 KiB	2022-10-13 16:11:29		InnoDB		Table
pekerja	6	16,0 KiB	2022-10-13 16:11:31		InnoDB		Table
absensi	5	16,0 KiB	2022-10-13 16:11:27		InnoDB		Table
absensi_backup	4	16,0 KiB	2022-10-13 16:11:27		InnoDB		Table
education_inst...	4	16,0 KiB	2022-10-13 16:11:28		InnoDB		Table
family	4	16,0 KiB	2022-10-13 16:11:28		InnoDB	PA0021	Table
user_login	4	16,0 KiB	2022-10-13 16:11:31		InnoDB		Table
employee_gro...	3	16,0 KiB	2022-10-13 16:11:28		InnoDB	T501	Table
organization_a...	3	16,0 KiB	2022-10-13 16:11:31		InnoDB		Table
education...	3	16,0 KiB	2022-10-13 16:11:28		InnoDB		Table

Gambar 3. 10 Daftar Tabel Database MySAU

Gambar 3.10 menunjukkan daftar tabel yang ada di database MySQL. Seluruh tabel yang ada pada Gambar 3.10. mempunyai relasi tabel sesuai dengan konsep database relasional untuk mencegah terjadinya kolom data yang *redundant*. Namun, tabel utama pada aplikasi ini adalah tabel *user_login*, *pekerja*, dan *absensi*.

Implementasi back-end pada aplikasi ini menggunakan bahasa pemrograman PHP dengan framework CodeIgniter 4. Framework ini berfungsi sebagai API pada React Native untuk kebutuhan seperti sistem login pada aplikasi dan sebagai penghubung antara aplikasi dengan database MySQL.



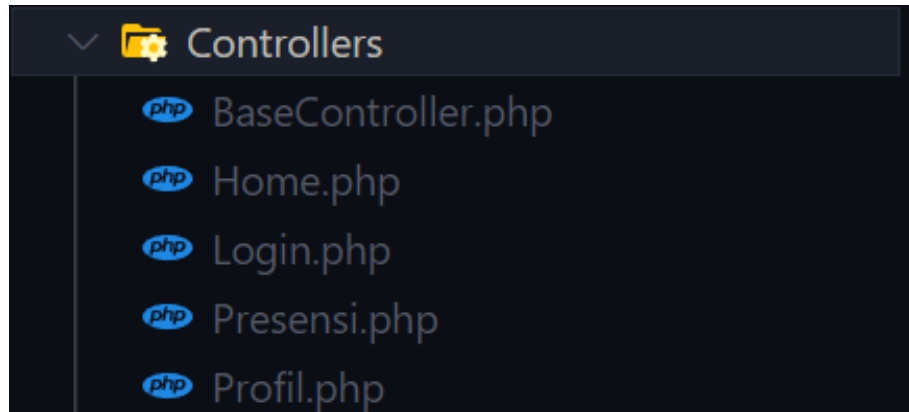
Gambar 3. 11 Models Database

Gambar 3.11 merupakan kumpulan model yang ada di back-end, model ini berfungsi untuk merepresentasikan tabel-tabel yang ada pada database MySQL untuk digunakan pada framework CodeIgniter 4. Seluruh tabel yang ada pada database dapat digunakan ketika membuat modelnya terlebih dahulu untuk dipakai pada controller.

```
Models > php UserLoginModel.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use CodeIgniter\Model;
6
7  class UserLoginModel extends Model
8  {
9      protected $table = 'user_login';
10     protected $allowedFields = ['Password'];
11 }
12
```

Gambar 3. 12 Model User Login

Gambar 3.12. adalah salah satu contoh model untuk tabel user_login yang ada di database MySQL. Model user login merepresentasikan tabel user_login pada database supaya pembuatan sistem login pada *controller* nanti dapat digunakan.



Gambar 3. 13 Controller Back-end

Gambar 3.13 merupakan kumpulan *controller* pada *controller* pada *back-end* ini berfungsi untuk mengatur segala aksi yang akan dijalankan oleh aplikasi. Pada *controller* ini ditulis fungsi-fungsi yang akan dilakukan oleh aplikasi. Contohnya pengecekan sistem *login*, pengambilan data pekerja, dan menyimpan data presensi ke dalam database MySQL.

```
public function auth()
{
    $session = session();
    // $model = new UserLoginModel();
    $PN = $this->request->getVar('username');
    $password = $this->request->getVar('password');
    // $data = $this->UserLoginModel->where('Pernr', $PN)->first();
    $data = $this->UserLoginModel->where('user_login.Pernr', $PN)->join('pekerja', 'user_login.Pernr =
pekerja.Pernr')->first();
    if ($data) {
        $pass = $data['Password'];
        $KategoriLevelUser = $this->UserLoginModel->select('admin.LevelUser')->join('admin', 'user_login.
Pernr = admin.Pernr')->findAll();
        if ($password == $pass) {
            $LevelUser = [];
        }
    }
}
```

Gambar 3. 14 Controller User Login

Gambar 3.14 merupakan potongan kode program untuk sistem login. Terlihat pada Gambar 3.14 ada sebuah *function* bernama *auth*, *function* ini berfungsi untuk *otentikasi* sistem *login* pada aplikasi tersebut. Back-end akan menerima masukan username dan password yang diisi pada kolom kemudian akan mencari username tersebut di database. Jika username tersebut ditemukan, maka password yang dimasukkan akan dicocokkan dengan yang ada pada database.

```

    $response = [
        'message' => 'Login Succesful',
        'token' => $token,
        'namaLengkap' => $data['NamaUser'],
        'jabatan' => $data['CorporateTitle']
    ];
    return $this->respond($response, 200);
} else {
    return $this->respond(['message' => 'Password anda salah.'], 200);
}
} else {
    return $this->respond(['message' => 'Email tidak terdaftar.'], 200);
}
}
}

```

Gambar 3. 15 Feedback Sistem Login

Gambar 3.15 merupakan potongan kode program untuk menampilkan respon pada tampilan aplikasi ketika melakukan *login*. Password pada masukan yang di-*submit* akan dicocokkan dengan password yang ada di database berdasarkan username. Jika password cocok maka back-end akan mengambil data dari database berdasarkan username dan aplikasi akan masuk ke dalam halaman *Home*. Jika password tidak cocok atau username tidak ditemukan, maka back-end akan mengirimkan respons ke front-end sebuah *alert*.

```

public function simpanPresensi()
{
    $jenisPresensi = $this->request->getVar('presensi');
    $PN = $this->request->getVar('username');
    $getPresensi = $this->PresensiModel->where('Pernr', $PN)->
    where('CreatedDate >=', date('Y-m-d'))->
    where('CreatedDate <=', date('Y-m-d') . " 23:59:59")->first();
    if ($jenisPresensi == 'masuk') {
        if (!$getPresensi) {
            $dataPresensi = [
                'Pernr' => $PN,
                'CreatedDate' => date("Y-m-d"),
                'StartLatitude' => $this->request->getVar('latitude'),
                'StartLongitude' => $this->request->getVar('longitude'),
                'CreatedBy' => $PN,
                'StartDate' => date('Y-m-d H:i:s'),
                'JenisPresensi' => $jenisPresensi
            ];
            $savePresensi = $this->PresensiModel->insert($dataPresensi);
        }
    }
}

```

Gambar 3. 16 Controller Presensi

Gambar 3.16 merupakan potongan kode program pada controller presensi yang berguna untuk menyimpan data presensi pekerja. Data akan di masukan ke dalam tabel presensi.

```

$response = [
    'message' => 'Presensi Masuk Berhasil Dilakukan',
    'token' => $token
];
return $this->respond($response, 200);
} else {
    $key = getenv('JWT_SECRET');
    $iat = time(); // current timestamp value
    $exp = $iat + 3600;

    $payload = array(
        "iss" => "Issuer of the JWT",
        "aud" => "Audience that the JWT",
        "sub" => "Subject of the JWT",
        "iat" => $iat, //Time the JWT issued at
        "exp" => $exp, // Expiration time of token
    );

    $token = JWT::encode($payload, $key, 'HS256');

    $response = [
        'message' => 'Anda Sudah Melakukan Presensi Masuk',
        'token' => $token
    ];
    return $this->respond($response, 200);
}

```

Gambar 3. 17 Respon Presensi

Gambar 3.17. merupakan respon yang akan dikeluarkan oleh back-end ke tampilan aplikasi untuk memberikan *alert* bahwa presensi berhasil dilakukan. Jika pekerja dalam hari yang sama sudah melakukan presensi namun melakukan kembali presensi maka aplikasi akan menampilkan respon bahwa presensi sudah dilakukan.

```

public function dataPekerja()
{
    $session = session();
    $Pernr = $this->request->getVar('username');
    $model = new UserLoginModel();
    $mainMenu = $model->select('user_login.pernr, accessmenu.IdMenu, accessmenu.LevelId, menu.title, menu.Url, menu.Deskripsi, menu.IDParent, menu.Url, menu.RoleAccess, menu.Icon')->join('admin', 'user_login.Pernr = admin.Pernr')->join('accessmenu', 'admin.LevelUser = accessmenu.LevelId')->join('menu', 'accessmenu.IdMenu = menu.IdMenu')->where('user_login.pernr', $Pernr)->where('menu.IDParent', '0')->findAll();
    $subMenu = $model->select('user_login.pernr, accessmenu.IdMenu, menu.title, menu.Url, menu.Deskripsi, menu.IDParent, menu.Url, menu.RoleAccess')->join('admin', 'user_login.Pernr = admin.Pernr')->join('accessmenu', 'admin.LevelUser = accessmenu.LevelId')->join('menu', 'accessmenu.IdMenu = menu.IdMenu')->where('user_login.pernr', $Pernr)->findAll();
    $data = [
        'mainMenu' => $mainMenu,
        'subMenu' => $subMenu,
        'levelUserLogin' => $session->get('LevelUser'),
        'dataFamilyPekerja' => $this->FamilyModel->join('pekerja', 'family.Pernr = pekerja.Pernr')->join('family_types', 'family.Type = family_types.FamType')->join('education_types', 'family.EduType = education_types.EduType')->where('family.Pernr', $Pernr)->findAll(),
        'listPekerja' => $this->PekerjaModel->findAll(),
        'dataPekerja' => $this->PekerjaModel->where('Pernr', $Pernr)->first(),
        'photoPekerja' => $this->ProfileModel->where('Pernr', $Pernr)->where('StartDate >=', date('Y-m-d'))->where('Status', 1)->first(),
        'dataEducationsPekerja' => $this->EducationFormalModel->join('pekerja', 'educations.Pernr = pekerja.Pernr')->join('education_types', 'educations.EduType = education_types.EduType')->join('education_schools', 'educations.Institution = education_schools.School')->join('education_majors', 'educations.Major1 = education_majors.Major')->orderBy('GradYear', "DESC")->where('pekerja.Pernr', $Pernr)->first();
    ];

    $response = [
        'nomorKTP' => $data['dataPekerja']['NomorKTP'],
        'tempatLahir' => $data['dataPekerja']['TempatLahir'],
        'tanggalLahir' => $data['dataPekerja']['TanggalLahir'],
        'jenisKelamin' => $data['dataPekerja']['JenisKelamin'],
        'agama' => $data['dataPekerja']['Agama'],
        'golonganDarah' => $data['dataPekerja']['GolonganDarah'],
        'statusNikah' => $data['dataPekerja']['StatusPernikahan'],
        'alamat' => $data['dataPekerja']['AlamatDomisili'],
    ];
}

```

Gambar 3. 18 Controller Profile

Gambar 3.18 merupakan controller yang didalamnya terdapat fungsi untuk mengambil data pengguna seperti data identitas diri, data pekerja utama, data pendidikan, dan data keluarga dari database MySQL. Data tersebut kemudian disimpan dan akan ditampilkan ke aplikasi menggunakan *respon* dari *JSON*.

Implementasi pada front-end aplikasi ini menggunakan framework *React Native*. Framework *React Native* merupakan framework yang digunakan untuk membuat aplikasi mobile, framework ini bersifat cross-platform sehingga dapat digunakan untuk membuat aplikasi Android dan iOS sekaligus memudahkan praktikan dalam mengembangkan MySAU Mobile versi iOS di kemudian hari jika diperlukan.

```
const HomeStack = createStackNavigator();
function HomeStackScreen() {
  return (
    <HomeStack.Navigator>
      <HomeStack.Screen
        name="HomeMenu"
        component={Home}
        options={{headerShown: false}}
      />
    </HomeStack.Navigator>
  );
}

const AppsStack = createStackNavigator();
function AppsStackScreen() {
  return (
    <AppsStack.Navigator>
      <AppsStack.Screen
        name="AppsMenu"
        component={Apps}
        options={{headerShown: false}}
      />
      <AppsStack.Screen name="Presensi" component={Attendance} />
    </AppsStack.Navigator>
  );
}
```

Gambar 3. 19 Route Halaman pada Front-end

Gambar 3.19 adalah potongan program pada *App.js*, *App.js* merupakan perutean untuk seluruh halaman yang ada pada aplikasi MySAU. Seluruh halaman dimasukkan sesuai dengan stacknya seperti Home dan Apps seperti menu di bagian bawah yang sudah dirancang sebelumnya pada mockup.


```
login = () => {
  Axios.post(
  )
  .then(responseJson => {
    if (responseJson.data.message == 'Login Succesful') {
      alert(responseJson.data.message);
      AsyncStorage.setItem('username', this.state.username);
      AsyncStorage.setItem('password', this.state.password);
      AsyncStorage.setItem('namaLengkap', responseJson.data.namaLengkap);
      AsyncStorage.setItem('jabatan', responseJson.data.jabatan);
      this.setState({isLogin: true});
      this.props.navigation.push('Navigation');
    } else {
      alert(responseJson.data.message);
      this.setState({isLogin: false});
    }
  })
  .catch(error => {
    console.error(error);
  });
};

render() {
  return (
    <View style={{flex: 1, backgroundColor: '#fff'}}>
      <StatusBar backgroundColor="#fff" barStyle="dark-content" />
      <View
        style={{justifyContent: 'center', alignItems: 'center', flex: 0.5}}>
        <Image
          source={require('../images/logo.png')}
          style={{
            width: 300,
            height: 300,
            marginTop: 20,
            marginBottom: 5
          }}
        />
      </View>
    </View>
  );
}
```

Gambar 3. 20 Front-end Halaman Login

Gambar 3.20 merupakan potongan program front-end menampilkan halaman *login*. Selain itu, program ini berfungsi untuk mengirimkan data yang di-*input* pada *form username* dan *password* ke API framework CodeIgniter 4. Data login akan disimpan pada AsyncStorage yang berfungsi untuk menyimpan data ke sebuah penyimpanan lokal pada android.



Personal Number

Password

LOGIN



Gambar 3. 21 Halaman Login

Gambar 3.21 merupakan halaman login yang dihasilkan dari Login.js pada framework React Native. *Button* pada halaman ini akan mengirimkan data *username* dan *password* ke API back-end. Kemudian API akan memberikan respon ke front-end berupa alert bahwa login sukses atau username/password salah.

```
render() {
  return (
    <ScrollView style={{flex: 1}}>
      <StatusBar backgroundColor="#fff" barStyle="dark-content" />
      <View style={{padding: 15}}>
        <View
          style={{
            backgroundColor: '#fff',
            borderRadius: 10,
            elevation: 10,
            justifyContent: 'center',
            alignItems: 'center',
          }}>
          <Image
            source={require('../images/submarinewhite.png')}
            style={{
              width: 355,
              height: 355,
            }}
            resizeMode="contain"
          />
        </View>
      </View>
    </ScrollView>
  )
}
```

Gambar 3. 22 Front-end Halaman Home

Gambar 3.22 merupakan potongan program front-end untuk menampilkan halaman *Home*. Pada front-end halaman *Home* digunakan elemen *ScrollView* agar konten pada halaman menyesuaikan berbagai resolusi perangkat *mobile*, jika resolusi perangkat *mobile* rendah maka akan muncul *scrollbar* sehingga konten tidak terpotong dan tetap dapat dilihat seluruhnya. *ScrollView* berguna agar menjadikan tampilan lebih dinamis. Pada halaman ini juga memuat sebuah gambar dengan opsi *contain*, ukuran gambar akan menyesuaikan sesuai dengan ukuran layar perangkat.



Gambar 3. 23 Halaman Home

Gambar 3.23 merupakan tampilan *Home*, halaman ini menampilkan profil perusahaan, visi, dan misi PT Siaga Abdi Utama. Seperti yang sudah dijelaskan sebelumnya, karena seluruh konten tidak bisa tercetak dalam 1 tampilan maka aplikasi akan menyesuaikan dengan fitur scrollbar agar bisa melihat seluruh konten. Pada halaman home ini juga memuat gambar seperti yang sudah dijelaskan pada potongan kode program gambar 3.22.

```

render() {
  return (
    <View style={{flex: 1}}>
      <StatusBar backgroundColor="#fff" barStyle="dark-content" />
      <View style={{flex: 0.3, padding: 15}}>
        <View style={{marginBottom: 5}}>
          <TouchableOpacity
            style={{
              justifyContent: 'center',
              alignItems: 'center',
              borderRadius: 10,
              backgroundColor: '#fff',
              elevation: 5,
              height: 100,
            }}
            onPress={() => {
              this.props.navigation.push('Presensi');
            }}
          >
            <FontAwesomeIcon
              size={50}
              icon={faLocationDot}
              color={'#0461a8'}
            />
            <Text style={{color: 'black', fontWeight: 'bold'}}>Presensi</Text>
          </TouchableOpacity>
        </View>
      </View>

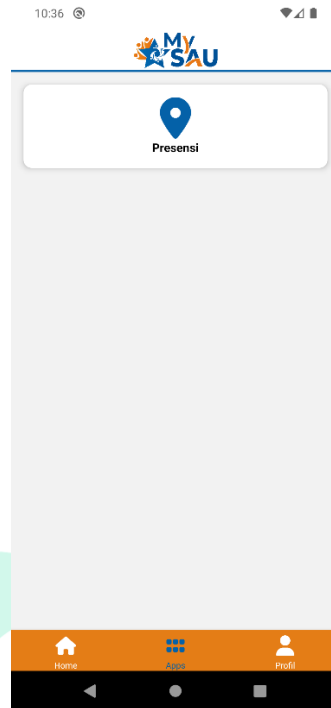
      <View style={{flex: 0.4, padding: 15}}></View>

      <View style={{flex: 0.1, padding: 15}}></View>
    </View>
  );
}

```

Gambar 3. 24 Front-end Halaman Apps

Gambar 3.34 menampilkan kode program front-end untuk halaman *Apps*. Halaman *Apps* berisi menu yang terbuat dari *button* menggunakan elemen *TouchableOpacity* untuk melakukan aksi ke halaman selanjutnya yang sudah ditentukan pada elemen tersebut.



Gambar 3. 25 Halaman Apps

Pada Gambar 3.35 menampilkan fitur-fitur pada aplikasi MySAU Mobile, untuk saat ini fitur yang tersedia hanya presensi online. Fitur presensi berguna

```

componentDidMount() {
  this.checkLogin();
  setInterval(() => {
    this.setState({
      currentTime: moment().format('ddd, Do MMMM YYYY, HH:mm:ss'),
    });
  }, 1000);
  Geolocation.getCurrentPosition(
    position => {
      this.setState({
        latitude: JSON.stringify(position.coords.latitude),
        longitude: JSON.stringify(position.coords.longitude),
      });
    },
    error => {
      Alert.alert(error.message.toString());
    },
    {
      showLocationDialog: true,
      enableHighAccuracy: true,
      timeout: 20000,
      maximumAge: 0,
    },
  );
}

```

Gambar 3. 26 Menampilkan Waktu dan Geo Location

Pada gambar 3.26 ditunjukkan potongan kode program untuk menampilkan waktu dan lokasi pengguna. Praktikan menggunakan *library moment*

untuk mengambil waktu terkini pada zona waktu dalam format Indonesia, sedangkan untuk pengambilan data lokasi titik koordinat menggunakan *library Geolocation*.

```
Presensi = () => {
  let username = this.state.username;
  let password = this.state.password;
  let presensi = this.state.presensi;
  let latitude = this.state.latitude;
  let longitude = this.state.longitude;
  Axios.post(
    [REDACTED]
  )
  .then(responseJson => {
    if (responseJson.data.message == 'Presensi Masuk Berhasil Dilakukan') {
      Alert.alert('Presensi', responseJson.data.message);
      this.props.navigation.push('Presensi', {
        username,
        password,
        presensi,
        latitude,
        longitude,
      });
    } else {
      Alert.alert('Presensi', responseJson.data.message);
    }
  })
  .catch(error => {
    console.error(error);
  });
};
```

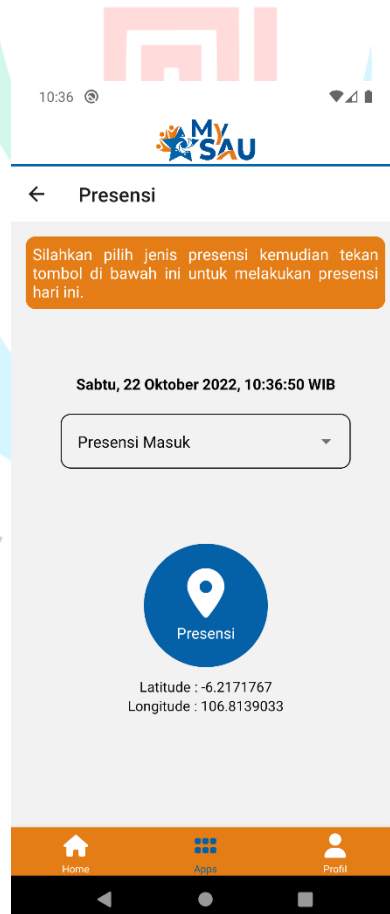
Gambar 3. 27 Koneksi API untuk Data Presensi

Gambar 3.27 merupakan potongan kode program koneksi API untuk data presensi. *Library Axios* pada *framework React Native* berfungsi untuk menghubungkan React Native dengan API melalui metode *get* dan *post* bahasa pemrograman PHP. Data presensi akan dikirimkan ke back-end aplikasi melalui koneksi *Axios* dengan menuliskan *URL* controller pada back-end beserta dengan *value* dan metodenya baik itu *post* maupun *get*. Pada front-end halaman Presensi juga diberikan kondisi jika akses lokasi pada perangkat dimatikan atau ditolak maka tombol untuk presensi akan dinonaktifkan dan tidak bisa ditekan.

```
render() {
  const colorScheme = Appearance.getColorScheme();
  let dropdown;
  if (colorScheme === 'dark') {
    dropdown = (
      <View
        style={{
          borderRadius: 25,
          marginTop: 20,
          backgroundColor: '#191A19',
        }}
      >
        <RNPickerSelect
          placeholder={{
            label: 'Pilih jenis presensi...',
            value: null,
            color: '#fff',
          }}
          items={[
            {label: 'Presensi Masuk', value: 'masuk', color: '#fff'},
            {label: 'Presensi Pulang', value: 'pulang', color: '#fff'},
          ]}
          onChange={value => this.setState({presensi: value})}
        />
      </View>
    );
  }
}
```

Gambar 3. 28 Front-end Halaman Presensi

Gambar 3.28 merupakan potongan kode program front-end untuk membuat dropdown pilihan jenis presensi yang akan ditampilkan pada halaman Presensi.



Gambar 3. 29 Halaman Presensi

Gambar 3.29 merupakan halaman Presensi yang menampilkan data waktu, geo lokasi pengguna, jenis presensi, dan tombol dinamis untuk melakukan presensi.

```
getDataPekerja = () => {
  Axios.post(
    'http://192.168.1.10:3000/api/pekerja',
  ).then(responseJson => {
    AsyncStorage.setItem('nomorKTP', responseJson.data.nomorKTP);
    AsyncStorage.setItem('tempatLahir', responseJson.data.tempatLahir);
    AsyncStorage.setItem('tanggalLahir', responseJson.data.tanggalLahir);
    AsyncStorage.setItem('jenisKelamin', responseJson.data.jenisKelamin);
    AsyncStorage.setItem('agama', responseJson.data.agama);
    AsyncStorage.setItem('golonganDarah', responseJson.data.golonganDarah);
    AsyncStorage.setItem('statusNikah', responseJson.data.statusNikah);
    AsyncStorage.setItem('alamat', responseJson.data.alamat);
    AsyncStorage.setItem('noTelp', responseJson.data.noTelp);
    AsyncStorage.setItem('noHP', responseJson.data.noHP);
    AsyncStorage.setItem(
      'emergencyNumber',
      responseJson.data.emergencyNumber,
    );
    AsyncStorage.setItem('email', responseJson.data.email);
    AsyncStorage.setItem('area', responseJson.data.area);
    AsyncStorage.setItem('subarea', responseJson.data.subarea);
    AsyncStorage.setItem('TMTmasuk', responseJson.data.TMTmasuk);
    AsyncStorage.setItem('TMTpt', responseJson.data.TMTpt);
    AsyncStorage.setItem('TMTpensiun', responseJson.data.TMTpensiun);
    AsyncStorage.setItem(
      'tingkatPendidikan',
      responseJson.data.tingkatPendidikan,
    );
    AsyncStorage.setItem('universitas', responseJson.data.universitas);
    AsyncStorage.setItem(
      'jurusanPendidikan',
      responseJson.data.jurusanPendidikan,
    );
    AsyncStorage.setItem('nilaiIPK', responseJson.data.nilaiIPK);
    AsyncStorage.setItem('tahunLulus', responseJson.data.tahunLulus);
    this.setState({isLogin: true});
    this.props.navigation.push('Detail Profil');
  });
};
```

Gambar 3. 30 Menyimpan Data dari Axios ke Async Storage

Gambar 3.30 merupakan kode program untuk menyimpan data dari axios ke Async Storage. Pada kode program front-end halaman Profil terdapat function untuk mengambil seluruh data pekerja yang sedang login dari API back-end dengan memanfaatkan *Axios* dan disimpan ke penyimpanan lokal perangkat tersebut. Function ini baru berjalan Ketika pengguna menekan tombol Detail Profil dan nantinya data ini akan ditampilkan pada halaman Detail Profil.

```
konfirLogout = () =>
  Alert.alert('Konfirmasi Logout', 'Apakah anda yakin ingin Logout ?', [
    {
      text: 'Cancel',
      style: 'cancel',
    },
    {text: 'OK', onPress: () => this.Logout()},
  ]);

Logout = async () => {
  AsyncStorage.clear();
  this.props.navigation.push('Login');
};
```

Gambar 3. 31 Fungsi Logout

Gambar 3.31 merupakan fungsi pada halaman *Profil* untuk melakukan *logout* dari akun pengguna yang sedang aktif. Fungsi ini akan berjalan Ketika pengguna menekan tombol Logout dan aplikasi akan berpindah ke halaman *Login* kembali.

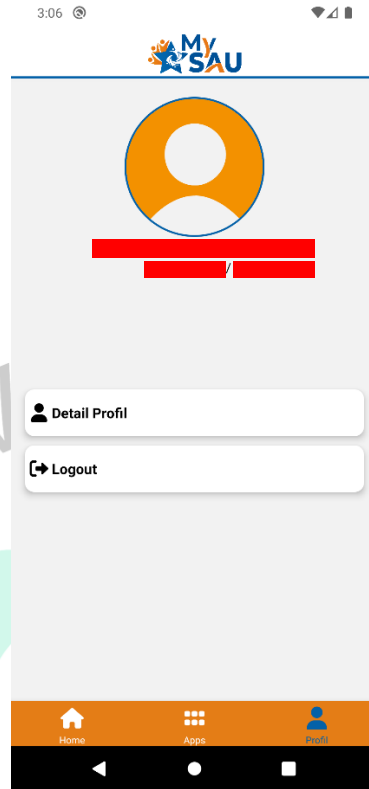
```

<View style={{flex: 0.5, padding: 15}}>
  <View style={{marginBottom: 5}}>
    <TouchableOpacity
      style={{
        borderRadius: 10,
        backgroundColor: '#fff',
        elevation: 5,
        height: 50,
        padding: 5,
        flexDirection: 'row',
        alignItems: 'center',
      }}
      onPress={this.getDataPekerja}>
      <FontAwesomeIcon size={20} icon={faUser} color={'#000'} />
      <Text style={{color: 'black', fontWeight: 'bold', fontSize: 15}}>
        {' '}
        Detail Profil
      </Text>
    </TouchableOpacity>
  </View>
  <View style={{marginVertical: 5}}>
    <TouchableOpacity
      style={{
        alignItems: 'center',
        flexDirection: 'row',
        borderRadius: 10,
        backgroundColor: '#fff',
        elevation: 5,
        height: 50,
        padding: 5,
      }}
      onPress={this.konfirLogout}>
      <FontAwesomeIcon
        size={20}
        icon={faRightFromBracket}
        color={'#000'}
      />
      <Text
        style={{
          color: 'black',
          fontWeight: 'bold',
          fontSize: 15,
        }}>
        {' '}
        Logout
      </Text>
    </TouchableOpacity>
  </View>
</View>

```

Gambar 3. 32 Front-end Halaman Profil

Gambar 3.32 merupakan potongan kode program untuk menampilkan halaman Profil. Pada gambar tersebut terdapat dua tombol untuk masuk ke menu Detail Profil dan Logout.



Gambar 3. 33 Halaman Profil

Gambar 3.33 merupakan tampilan Halaman Profil. Halaman Profil menampilkan nama lengkap pengguna yang sedang login, nomor pekerja dan jabatan pekerja. Di bawahnya terdapat tombol untuk melihat lebih lengkap data pekerja dan tombol untuk logout dari akun pengguna tersebut.

```

getDetailPekerja = async () => {
  try {
    let findNamaLengkap = await AsyncStorage.getItem('namaLengkap');
    this.setState({
      namaLengkap: findNamaLengkap,
      isLogin: true,
    });
    let findUsername = await AsyncStorage.getItem('username');
    this.setState({
      username: findUsername,
      isLogin: true,
    });
    let findNIK = await AsyncStorage.getItem('nomorKTP');
    this.setState({
      nik: findNIK,
      isLogin: true,
    });
    let findTempatLahir = await AsyncStorage.getItem('tempatLahir');
    this.setState({
      tempatLahir: findTempatLahir,
      isLogin: true,
    });
    let findTanggallahir = await AsyncStorage.getItem('tanggallahir');
    this.setState({
      tanggallahir: findTanggallahir,
      isLogin: true,
    });
    let findJenisKelamin = await AsyncStorage.getItem('jenisKelamin');
    this.setState({
      jenisKelamin: findJenisKelamin,
      isLogin: true,
    });
  }
}

```

Gambar 3. 34 Menampilkan data dari AsyncStorage

Gambar 3.34 menunjukkan kode program untuk memanggil data dari penyimpanan lokal perangkat yang sudah disimpan menggunakan AsyncStorage pada halaman Profil sebelumnya. Data tersebut kemudian dimasukkan ke *State*.

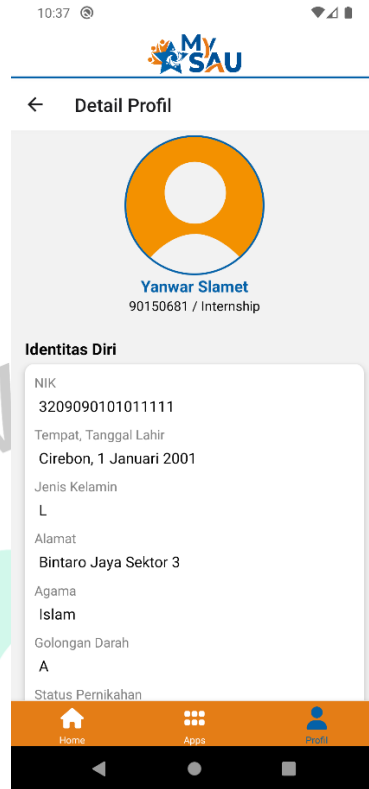
```

return (
  <ScrollView>
    <View>
      <View
        style={{
          justifyContent: 'center',
          alignItems: 'center',
          paddingTop: 20,
        }}>
        <View
          style={{
            justifyContent: 'center',
            alignItems: 'center',
            width: 150,
            height: 150,
            borderRadius: 150 / 2,
            overflow: 'hidden',
            borderWidth: 2,
            borderColor: '#0461a8',
          }}>
          <Image
            source={require('../images/user.png')}
            style={{width: 200, height: 200, borderRadius: 200 / 2}}
            resizeMode="contain"
          />
        </View>
        <Text style={{color: '#0461a8', fontSize: 17, fontWeight: 'bold'}}>
          {this.state.namaLengkap}
        </Text>
        <Text style={{color: 'black'}}>
          {this.state.username} / {this.state.jabatan}
        </Text>
      </View>
    </View>
  </ScrollView>
)

```

Gambar 3. 35 Front-end Detail Profil

Gambar 3.35 merupakan potongan kode program pada front-end Detail Profil. Data *AsyncStorage* yang sudah dipanggil dan dimasukkan ke dalam *State* selanjutnya ditampilkan pada Halaman *Detail Profil*.



Gambar 3. 36 Halaman Detail Profil

Pada gambar 3.36 ditunjukkan halaman Detail Profil. Halaman Detail Profil menampilkan seluruh data pengguna meliputi data identitas diri, data pekerja utama, kontak, dan data pendidikan.

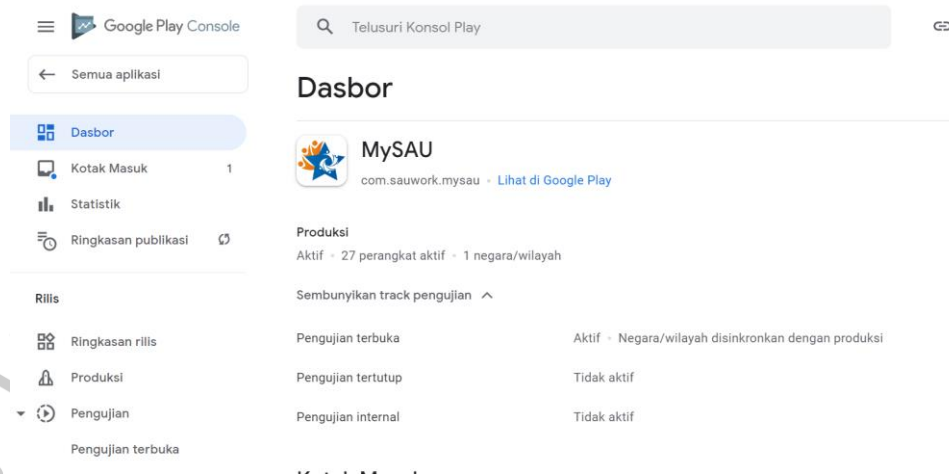
3.2.3 Pengujian

Pada tahap pengujian, praktikan menggunakan pengujian metode *black box* karena menurut praktikan dengan pengujian metode *black box* memiliki beberapa keuntungan mengingat pengujian *black box* dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambguitas atau tidak konsistensi alat dalam spesifikasi persyaratan. Pengujian *black box* merupakan teknik pengujian *software* yang berfokus pada spesifikasi fungsionalitas.

Tabel 3. 5 Pengujian *Black Box*

No.	Skenario Pengujian	Hasil Yang Diharapkan	Kesimpulan
1	Login Aplikasi	Aplikasi dapat login dan menampilkan menu Home	Berhasil
2	Menu Presensi	Aplikasi dapat mendeteksi Waktu dan Lokasi pengguna serta melakukan presensi harian	Berhasil
3	Menu Profil	Aplikasi dapat menampilkan profil pekerja yang sedang login	Berhasil
4	Logout	Aplikasi keluar dari akun pengguna dan Kembali ke halaman Login	Berhasil

Pada Tabel 3.5 merupakan tabel pengujian black box yang menggambarkan pengujian yang dilakukan pada sistem aplikasi MySAU. Pengujian tidak hanya dilakukan oleh praktikan dan Tim IT SAU, pengujian juga dilakukan oleh pihak Google pada saat publikasi Aplikasi MySAU ke *platform* Google Play Store.



Gambar 3. 37 Publikasi MySAU ke Google Play Store

Gambar 3.37 merupakan cuplikan Google Play Console untuk publikasi aplikasi MySAU ke Google Play Store. Pada gambar 3.37 ditunjukkan bahwa aplikasi tersebut sudah dilakukan pengujian terbuka dan sudah tersedia di Google Play Store untuk diunduh.

3.3 Kendala Yang Dihadapi

Selama pelaksanaan kegiatan MBKM di PT Bank Rakyat Indonesia, praktikan mendapatkan permasalahan dalam melakukan pekerjaan. Berikut adalah kendala yang dihadapi oleh praktikan:

1. Karena praktikan hanya mengenal dasar dari bahasa pemrograman React Native, maka dibutuhkan waktu untuk melakukan implementasi aplikasi terutama pada bagian API.
2. Banyak library yang terbilang baru digunakan oleh praktikan seperti Geolocation dan Axios sehingga praktikan membutuhkan waktu untuk mempelajari penerapan library tersebut.
3. Kesibukan waktu praktikan dan pembimbing dalam pekerjaannya masing-masing sehingga terkadang sedikit terhambat waktu dalam berkonsultasi yang menjadikan tidak efektif.

3.4 Cara Mengatasi Kendala

Selama melaksanakan MBKM di PT Bank Rakyat Indonesia praktikan menjadi terlatih bagaimana mencari solusi untuk menyelesaikan suatu masalah yang dihadapi. Penyelesaian masalah tersebut tidak luput dengan adanya diskusi antara berbagai pihak seperti pembimbing eksternal dan pekerja yang lain dalam tim. Berikut ini merupakan cara mengatasi kendala yang dilakukan oleh praktikan.

1. Memberanikan diri untuk berdiskusi kepada pembimbing eksternal terkait dengan kendala yang dihadapi serta meminta saran dari program sementara yang telah dihasilkan.
2. Belajar secara mandiri mengenai penggunaan library yang digunakan di internet dan giat membaca dokumentasi library yang akan digunakan.
3. Mencoba *coding* langsung secara *try and error* pada program yang akan dibuat sehingga perlahan-lahan mulai memahami letak permasalahan pada program tersebut.

3.5 Pembelajaran Yang Diperoleh dari MBKM

Setiap kegiatan yang dilakukan selama kegiatan MBKM di PT Bank Rakyat Indonesia mempunyai manfaat bagi praktikan. Pembelajaran yang diperoleh selama kegiatan MBKM di PT Bank Rakyat Indonesia yaitu sebagai berikut.

1. Mendapatkan pembelajaran mengenai pentingnya disiplin waktu terutama pada pembuatan aplikasi MySAU dan rasa tanggung jawab atas pekerjaan yang telah diberikan oleh pembimbing eksternal.
2. Mampu belajar mandiri untuk memperoleh ilmu pengetahuan tentang bahasa pemrograman dan penerapan sebuah library pada aplikasi.
3. Melatih keterampilan dan kreatifitas praktikan dalam membangun sebuah aplikasi *mobile* seperti pembuatan logika alur program, tampilan front-end aplikasi yang *user-friendly* agar pengguna tidak merasa kesulitan pada saat menggunakan aplikasi tersebut.
4. Melatih kemampuan berbicara pada saat berkomunikasi dengan rekan kerja dan pembimbing eksternal serta kemampuan mempresentasikan cara penggunaan aplikasi.

5. Mengetahui bagaimana kondisi lingkungan pekerjaan di PT Bank Rakyat Indonesia (Persero) Tbk.

