

BAB 2

TINJAUAN PUSTAKA

Pembangunan aplikasi presensi karyawan yang ditujukan untuk PT Motiolabs Digital Indonesia yang berguna untuk pengelolaan data presensi karyawan ini dibutuhkan tinjauan pustaka yang dapat membantu dalam proses pembuatan aplikasi beserta objek – objek yang diperlukan dalam pembuatan aplikasi ini. Berikut beberapa tinjauan pustaka yang digunakan dalam pembuatan aplikasi ini.

2.1 Profil PT Motiolabs Digital Indonesia

Dalam penelitian ini yang menjadi objek penelitian adalah PT Motiolabs Digital Indonesia di mana perusahaan ini merupakan perusahaan yang bergerak pada bidang penyediaan jasa teknologi digital. Berikut ini merupakan profil dari PT Motiolabs Digital Indonesia:

Nama	: PT Motiolabs Digital Indonesia
Alamat Kantor	: Jl. Sidomulyo No 31, Sukaluyu, Bandung, Jawa Barat
Jam Operasional	: 08:00 – 17:00 WIB
Email	: halo@motiolabs.com

2.2 Visi Misi

Berikut ini merupakan visi dan misi dari PT Motiolabs Digital Indonesia

2.2.1 Visi

Memberikan solusi teknologi untuk mendorong kemajuan transformasi digital bagi entitas bisnis di Indonesia

2.2.2 Misi

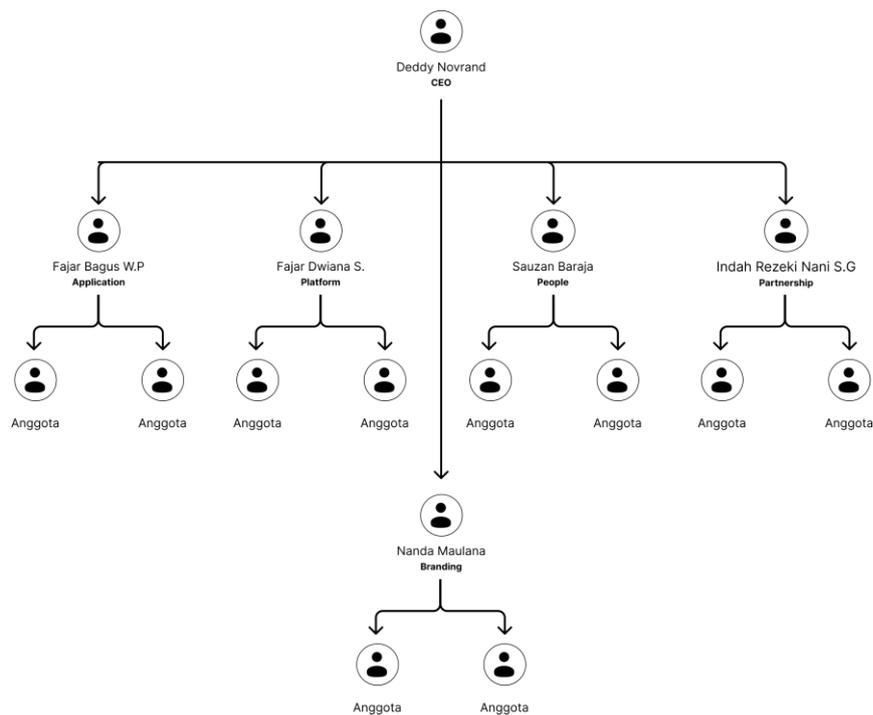
Misi yang dilakukan PT Motiolabs Digital Indonesia untuk mencapai visinya adalah sebagai berikut:

1. Menyediakan solusi digital dan layanan pendukung untuk beberapa industri utama di Indonesia.

2. Menjadi perusahaan yang disegani dan berkembang pesat dengan tetap menjaga profitabilitas.
3. Berkontribusi secara aktif untuk mengatasi tantangan klien kami dalam menerapkan transformasi digital.
4. Menjadi organisasi tempat karyawan dapat berkembang dan bekerja dengan nyaman.
5. Memberikan dampak dan kontribusi positif bagi komunitas, masyarakat, dan Indonesia.

2.3 Struktur Organisasi

Berdasarkan hasil wawancara dengan direktur dari PT. Motiolabs Digital Indonesia, Gambar 2.1 adalah struktur organisasi dari PT Motiolabs Digital Indonesia:



Gambar 2.1 Struktur Organisasi PT Motiolabs Digital Indonesia

Struktur organisasi pada gambar 2.1 diawali dengan direktur yang didalamnya memiliki beberapa divisi diantaranya divisi *application*, *platform*, *branding*, *operation*, *people* dan *partnership* yang mana masing-masing divisi tersebut memiliki anggota dan fungsinya masing – masing.

2.4 Sistem

Sistem secara umum dapat diartikan sebagai satu kesatuan dari kumpulan atau himpunan yang terdiri dari berbagai komponen atau variabel yang saling berhubungan dan saling ketergantungan untuk memudahkan aliran informasi dalam mencapai suatu tujuan tertentu. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, contohnya seperti penggunaan model matematika [9], [10].

Umumnya sebuah sistem terdiri dari input, proses, dan output. Hal ini merupakan sebuah konsep yang mendasar pada sistem mengingat sebuah sistem dapat menerima beberapa masukan dan menghasilkan beberapa keluaran sekaligus. Sistem juga memiliki sifat – sifat tertentu, yang menjadikan sebuah identitas atau ciri yang unik untuk menandakan bahwa hal tersebut dapat dikatakan sebuah sistem [9]. Untuk mencapai itu semua terdapat beberapa karakteristik yang dapat membuat sebuah sistem dikatakan baik, yaitu:

1. Suatu sistem terdiri dari sejumlah komponen atau variabel yang saling berinteraksi antara satu dengan yang lainnya untuk membentuk suatu kesatuan.
2. Suatu sistem harus memiliki batasan untuk membatasi antara sistem dengan sistem lainya agar terhindar dari adanya penyimpangan yang disebabkan oleh sistem diluar lingkungannya.
3. Suatu sistem harus memiliki lingkungan luar yang dapat mempengaruhi sistem tersebut.
4. Suatu sistem harus memiliki penghubung antara satu subsistem dengan subsistem lainnya yang memungkinkan sumber – sumber daya mengalir dari satu sistem ke subsistem yang lainnya.
5. Suatu sistem harus memiliki sebuah masukan agar sistem tersebut dapat beroperasi sebagaimana mestinya.
6. Suatu sistem harus memiliki sebuah keluaran dari hasil masukan yang telah diproses untuk menghasilkan keluaran yang berguna atau bermanfaat.
7. Suatu sistem harus memiliki kemampuan memproses agar dapat merubah naskah menjadi keluaran.

8. Suatu sistem harus memiliki sasaran atau tujuan yang jelas agar menghasilkan informasi yang sesuai dengan tujuan yang telah ditentukan.

2.5 Presensi

Presensi adalah suatu kegiatan pencatatan daftar kehadiran individu yang tidak dapat diwakilkan pada suatu kegiatan yang berlangsung. Proses ini bertujuan untuk mengolah informasi yang telah dikumpulkan untuk menilai keberhasilan dari suatu kegiatan yang berlangsung dalam periode tertentu dan mendapatkan informasi dari individu yang telah mengikuti kegiatan tersebut. Presensi dalam kasus ini ditujukan untuk mencatat kehadiran karyawan di PT Motiolabs Digital Indonesia [11].

Dengan adanya presensi ini maka dapat mengetahui rekap kehadiran karyawan yang dapat digunakan untuk menilai kinerja dari masing – masing karyawan selain itu dapat juga digunakan untuk penghitungan gaji karyawan. Presensi yang dilakukan secara manual memungkinkan untuk terjadinya *human error* yang mengakibatkan data presensi tidak valid. Selain itu memungkinkan juga untuk terjadinya tindakan kecurangan seperti memanipulasi data presensi, serta presensi yang dilakukan secara manual memerlukan waktu pengolahan data akhir yang cukup lama untuk mendapatkan hasil yang akurat yang mana dapat mengurangi efisiensi dan efektivitas dalam pengambilan dan pengolahan data [11].

2.6 Karyawan

Karyawan adalah seseorang yang bekerja untuk suatu perusahaan, organisasi, atau individu lainnya dengan status sebagai pekerja tetap atau kontrak. Karyawan memiliki tanggung jawab untuk melaksanakan tugas sesuai dengan peran yang telah ditetapkan oleh perusahaan serta harus mematuhi setiap aturan dan kebijakan yang telah ditentukan oleh perusahaan, dengan tujuan untuk mencapai tujuan bisnis yang telah ditetapkan. Maka dari itu seorang karyawan berhak menerima upah sebagai imbalan atas pekerjaan yang telah dilakukannya [5]. Karyawan sendiri dibagi menjadi beberapa jenis, diantaranya:

1. Pegawai Negeri

Merupakan seseorang yang berkerja pada salah satu sektor pemerintahan seperti sekretariat provinsi daerah.

2. Pegawai Swasta

Merupakan seseorang yang bekerja diluar dari sektor pemerintahan yang artinya perusahaan yang didirikan oleh seseorang atau kelompok.

3. Pegawai BUMN

Merupakan seseorang yang berkerja pada sebuah perusahaan milik negara contohnya seperti Pertamina.

2.7 Aplikasi

Aplikasi atau yang biasa disingkat *apps* merupakan sebuah program siap pakai yang dibuat untuk kebutuhan tertentu sesuai dengan permasalahan yang akan diselesaikan untuk nantinya digunakan oleh end-user dalam membantu suatu pekerjaan, memberikan informasi dan edukasi, memfasilitasi aktivitas hiburan, dan masih banyak lagi yang lainnya yang dapat dimanfaatkan dengan adanya sebuah aplikasi. Contoh aplikasi yang umum digunakan oleh sebagian besar orang adalah aplikasi pengolah kata, pemutar media, penampil file, web browser, email dan editor foto maupun video [12].

Dalam era digital saati ini, kebutuhan akan sebuah aplikasi telah menjadi bagian penting dalam kehidupan sehari – hari kerana dapat mempermudah seseorang dalam melakukan banyak hal. Aplikasi juga menjadi salah satu cara bagi sebuah perusahaan atau individu untuk berinteraksi dengan pengguna secara lebih dekat dan memberi pengalaman pengguna lebih baik.

2.8 Android

Android adalah sistem operasi yang dirancang untuk perangkat *mobile* yang didasarkan pada kernel linux yang telah dimodifikasi. Android dikembangkan secara bersamaan oleh NVIDIA, Qualcomm, Motorola, T-Mobile, Intel, HTC dan Google yang tergabung dalam sebuah organisasi yang bernama OHA (*Open Handset Alliance*). Selain itu sistem operasi ini bersifat terbuka (open source) sehingga siapapun dapat menggunakannya secara bebas tanpa adanya biaya royalti, yang membuat para pegembang dapat berkreaitivitas tanpa adanya biaya

keanggotaan dan biaya lainnya, hal ini dapat menguntungkan pengembang karena aplikasi yang mereka buat dapat di distribusikan secara bebas dalam bentuk apapun [13], [14].

2.9 Visual Studio Code

Visual studio code merupakan editor kode sumber yang dikembangkan oleh Microsoft. Editor ini memiliki keunggulan salah satunya cukup ringan untuk digunakan dan bersifat *open source*, selain itu editor ini juga tersedia baik untuk Windows, macOS, dan Linux. Adapun fitur – fitur yang telah disediakan pada editor ini berguna untuk memudahkan proses pengembangan perangkat lunak, diluar dari fitur – fitur yang telah disediakan editor ini juga menyediakan *extension* atau *plugin* yang dapat di tambahkan secara mandiri untuk menambahkan fungsionalitas dari editor ini [15].

Dengan banyak atau melimpahnya fitur yang tersedia pada editor ini tentu dapat mendukung berbagai macam bahasa pemrograman seperti javascript, typescript, python, PHP, dan masih banyak lagi yang membuat editor ini menjadi fleksibel karena dapat digunakan oleh berbagai macam bahasa pemrograman. Hingga saat ini editor ini menjadi editor yang populer digunakan oleh pengembang karena memiliki antarmuka yang mudah digunakan dan memiliki fitur yang lengkap [15].

2.10 Flutter

Flutter merupakan sebuah kerangka kerja dalam pembuatan aplikasi android yang dikembangkan oleh google untuk membuat sebuah antarmuka yang cantik untuk aplikasi *mobile*. Flutter dapat berjalan baik di android maupun ios karena flutter merupakan sebuah kerangka kerja yang *hybrid* sehingga dalam satu kali pengkodean dapat di gunakan untuk sistem operasi yang berbeda yang dapat mempercepat proses pengembangan aplikasi [16] [17].

Pengkodean antar muka dengan menggunakan flutter sering disebut dengan *widget*. *Widget* merupakan kompoenen visual yang disediakan flutter untuk membangun antar muka yang cantik. *Widget* sendiri terbagi menjadi dua yaitu *stateless widget* dan *statefull widget*, *stateless widget* merupakan komponen *widget*

yang statis dimana seluruh konfigurasi yang dimuat telah di inisialisasi sejak awal sedangkan *statefull widget* merupakan kompoenen *widget* yang dinamis dimana *widget* ini dapat diperbaharui berdasarkan aksi yang pengguna lakukan atau ketika terjadinya perubahan data [16] [17].

Flutter mempunyai dua macam widget untuk pengembang aplikasi, yaitu Material Design dan Cupertino. Material Design adalah bahasa design yang dibuat oleh Google, design ini sama dengan design yang dipakai pada Android, sedangkan cupertino atau dengan sebutan lain gaya iOS adalah bahasa design yang dipakai oleh iOS. Flutter mempunyai lebih banyak widget Material Design daripada widget Cupertino, tetapi tidak usah khawatir, pada OS perangkat yang berbeda, widget bisa dipakai secara cross-platform [17]. Pada Flutter sendiri terdapat 2 cara yang dapat dilakukan untuk menjalankan aplikasi, yaitu:

1. Debug

Flutter akan berjalan secara JIT (Just in Time), aplikasi akan dicompile pada saat aplikasi berjalan. Dengan menggunakan debug, terdapat beberapa fitur yang sangat diperlukan oleh pengembang aplikasi yaitu, assert dinyalakan, observatory dinyalakan berguna untuk debug. Service extension dinyalakan, kompilasi dioptimisasi untuk pengembangan (sehingga tidak dioptimisasi untuk kecepatan, ukuran aplikasi). Pada debug Flutter juga mempunyai fitur hot reload dan hot restart. Hot reload adalah melakukan penerapan ulang pada aplikasi tanpa membuang state. Hot restart adalah seperti menutup aplikasi dan membuka lagi, tetapi hanya bagian Flutternya saja, yang pastinya akan mereset state [17].

2. Release

Flutter akan berjalan secara AOT (Ahead of Time), sehingga aplikasi perlu dicompile terlebih dahulu. Dengan menggunakan release, Flutter akan mematikan fungsi assert, informasi debug dihapus, debug dimatikan, kompilasi dioptimisasi untuk kecepatan dan ukuran aplikasi, service extension dimatikan [17].

2.11 Dart

Dart merupakan bahasa pemrograman terstruktur *open source* untuk membuat aplikasi web berbasis browser yang kompleks. Bahasa pemrograman dart dikembangkan oleh google untuk kebutuhan umum. bahasa pemrograman dart dapat dijalankan baik dengan menggunakan browser yang secara langsung mendukung kode dart atau dengan mengkompilasi kode dart ke dalam Javascript. Pada umumnya penggunaan dart pada web memakai dart2js yang artinya dart diubah ke javascript agar bisa dimengerti oleh browser, Pada perangkat mobile dart memakai flutter, dan pada perangkat desktop dart memakai dart2aot yang mengubah dart menjadi bahasa mesin [16]–[18].

Dart memiliki sintaks yang familiar karena menerapkan konsep pemrograman berorientasi objek (OOP), yang pada umumnya hampir semua bahasa pemrograman menerapkan konsep ini dimana struktur kode berada dalam class yang didalamnya berisi method maupun variabel. Dart juga menerapkan yang namanya single thread, yang mana ini memungkinkan mengeksekusi sebuah kode program secara paralel karena memiliki model konkurensi yang disebut dengan isolat. Selain itu dart dapat melakukan pemrograman asynchronous yaitu fungsi yang non-blocking yang memungkinkan program untuk menjalankan kode lain sambil menunggu fungsi asynchronous selesai. Dart juga bisa melakukan pemrograman Reactive, yaitu pemrograman asynchronous dengan stream data. Stream data artinya datanya bisa di terima lebih dari sekali, kebalikan dengan asynchronous yang datanya hanya sekali terima saja. Pada Dart asynchronous menggunakan kata Future, sedangkan reactive menggunakan Stream. Sintak yang dimiliki dart sendiri memiliki mekanisme yang umumnya hampir sama dengan bahasa pemrograman javascript, java, C++ dan C# [16]–[18].

2.12 Javascript

Pada awalnya yaitu pada tahun 1995 perusahaan yang bernama Netscape menginginkan adanya perilaku dinamis yang dapat ditambahkan pada halaman web di browsernya. Berawal dari itu semua bahasa pemrograman javascript sering disingkat JS mulai dikembangkan oleh Brendan Eich. Javascript merupakan bahasa pemrograman web yang bersifat *client side programming language*, dimana

pemrosesannya dilakukan disisi client bukan di sisi server seperti halnya bahasa pemograman PHP [19], [20].

Dengan pemrosesan yang hanya dapat dilakukan di sisi *client* ini lah, maka bahasa pemograman javascript hanya dapat dijalankan pada browser karena bahasa pemograman javascript ditanam didalam halaman website. Selain itu javascript merupakan bahasa script yang mampu membuat sebuah halaman website menjadi lebih interaktif, untuk melakukan hal itu perlu yang namanya DOM (*Document Object Model*). DOM merupakan interface yang memungkinkan untuk manipulasi halaman web baik dari segi struktur, tampilan dan kontennya [19].

2.13 JSON

Javascript Object Notation (JSON) adalah format pertukaran data ringan dalam bentuk text yang dapat dengan mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat [21]. JSON pertama kali dipopulerkan oleh Douglas Crockford. Pada tahun 2006 JSON secara resmi menjadi tata bahasa formal dibawah RFC 4627 melalui sebuah organisasi yang bernama *Internet Engineering Task Force* (IETF). Pada bulan desember 2009, JSON secara resmi diakui sebagai bahasa standar ECMA. JSON secara garis besar berbentuk sebuah objek yang didalamnya dapat di isi oleh berbagai jenis struktur data [20], [21].

Objek pada JSON memiliki struktur data yang universal, meliputi kumpulan pasangan atribut-nilai dan daftar nilai terurutan atau larik. Ini merupakan format data yang sangat umum dan digunakan sebagai pengganti XML. JSON merupakan jenis format data yang diturunkan dari javascript, tetapi banyak dari bahasa pemograman modern yang sudah dapat mem-parsing format data JSON, ini menjadikan JSON ideal digunakan sebagai media pertukaran data antar bahasa pemograman yang berbeda [20], [21]. Untuk lebih jelasnya mengenai struktur JSON dapat dilihat pada Gambar 2.2 Contoh Struktur JSON.



Sumber Gambar: <https://www.petanikode.com/json-pemula/>

Gambar 2.2 Contoh Struktur JSON

2.14 Node JS

Node JS merupakan *runtime environment* dari javascript yang memperluas javascript sebagai bahasa pemrograman yang tidak hanya bisa digunakan di sisi *client* namun dapat digunakan sebagai *server-side*. Node JS bersifat *open source* yang artinya dapat digunakan oleh siapapun secara gratis. Node JS menjalankan V8 javascript *engine*, sehingga dapat mengeksekusi javascript diluar *environment* browser yang memungkinkan memiliki performa yang tinggi [22].

2.15 Express JS

Express JS merupakan salah satu framework yang berasal dari bahasa pemrograman javascript yang dirancang secara fleksibel dan minimalis. Express JS sendiri merupakan sebuah library dasar dari framework Node JS. Dengan menggunakan Express JS pengembang tidak perlu lagi menggunakan module http bawaan Node JS. Selain itu Express JS tidak memakan resource yang cukup banyak sehingga cocok untuk diterapkan dalam pembuatan Rest API karena dapat meningkatkan performa pada saat pemanggilan API [15].

2.16 PostgreSQL

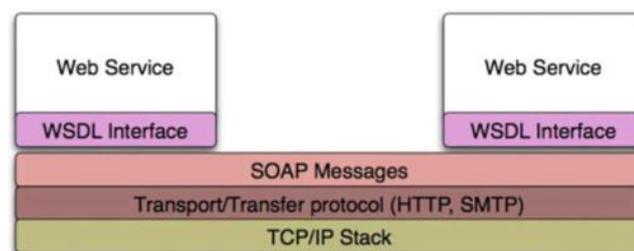
PostgreSQL merupakan salah satu jenis *database management system* atau manajemen berbasis data yang bersifat *open-source*. PostgreSQL dikembangkan oleh University of California di Berkeley Computer Science Departement.

PostgreSQL merupakan database relational yang mendukung perintah perintah SQL maupun PL/pgSQL (Procedural Language). Selain itu PostgreSQL juga memiliki dokumentasi yang lengkap, komunitas dalam jumlah yang cukup besar, serta didukung oleh berbagai bahasa pemrograman seperti PHP, Python, Java, C++, Perl, dan Tcl [23].

2.17 Web Service

Web service adalah sebuah teknologi berbasis web yang memungkinkan sebuah aplikasi dapat saling berkomunikasi dan berbagi data satu sama lain melalui internet dengan menggunakan jaringan protokol HTTP atau HTTPS untuk menyelesaikan masalah domain tertentu. Layanan ini dapat digunakan dan dimanfaatkan oleh pengguna bahasa pemrograman, sistem operasi dan arsitektur yang berbeda.

Berkaitan dengan arsitektur, ada beberapa arsitektur yang dapat digunakan dalam pembuatan sebuah *web service* yaitu dengan menggunakan *simple object access protocol* (SOAP) dan *representational state transfer* (REST), sementara service tersebut direpresentasikan dalam bentuk format text, JSON, atau XML agar lebih mudah dibaca dan dipahami.



Gambar 2.3 Struktur logis tumpukan komunikasi layanan web

2.18 Application Programming Interface (API)

Application Programming Interface adalah mekanisme yang dijalankan agar dua komponen perangkat lunak dapat saling berkomunikasi dengan serangkaian definisi dan protokol tanpa harus mengetahui bagaimana sistem atau layanan tersebut bekerja di dalamnya [22]. *Application Programming Interface* sangat erat kaitannya dengan *client* dan *server*, dimana *client* merupakan yang mengirim

permintaan ke *server* sedangkan *server* yang mengirim response hasil dari *request client*. *Application Programming Interface* memiliki empat cara kerja diantaranya:

1. API SOAP

Merupakan salah satu cara kerja api yang populer di masa lalu dimana API ini menggunakan *Simple Object Access Protocol* dimana pertukaran data antara *client* dengan *server* menggunakan XML.

2. API RPC

Merupakan cara kerja API yang memanfaatkan *Remote Procedure Calls* atau panggilan prosedur jarak jauh, dimana *client* menjalankan fungsi pada server, dan server mengirimkan output kembali ke *client*.

3. API Websocket

Merupakan cara kerja API yang memanfaatkan objek JSON untuk meneruskan data. API Websocket mendukung komunikasi dua arah antara *client* dan *server*.

4. API REST

Merupakan cara kerja API yang saat ini sangat populer karena fleksibel. API REST memanfaatkan protokol http dalam pengiriman informasinya, dimana *client* mengirim *request* ke *server* kemudian *server* memberikan response kepada *client* baik dalam bentuk JSON, XML, atau teks.

2.18.1 Google Maps API

Google Maps API merupakan sebuah layanan yang dikembangkan oleh google dengan tujuan untuk mempermudah pengembang dalam mendapatkan layanan peta digital. Didalam API ini selain dapat menampilkan layanan peta digital menyediakan juga layanan lainnya seperti pencarian lokasi, rute, pemberian marker, serta dapat mengubah tampilan peta yang dapat disesuaikan dengan aplikasi yang dibuat [7]. Dengan adanya layanan ini tentu dapat lebih mempermudah serta dapat mengurangi waktu proses pengembangan yang membutuhkan visualisasi berupa peta digital. Untuk dapat menggunakan layanan google maps api terdapat beberapa hal yang harus dilakukan, diantaranya:

1. Melakukan pendaftaran akun Google Cloud Platform.

2. Buat proyek baru dengan nama proyek yang dapat disesuaikan.
3. Aktifkan Google Maps JavaScript API pada konsol pengembangan Google Cloud Platform.
4. Dapatkan kunci API yang diperlukan untuk mengakses Google Maps API.
5. Setelah mendapatkan kunci API, selanjutnya dapat dilakukan pengintegrasian dengan proyek aplikasi yang sedang di buat.

2.18.2 Google Machine Learning Kit

Dalam proses pengembangan aplikasi, saat ini para developer banyak diberikan kemudahan untuk mempercepat pekerjaan, salah satunya dengan adanya layanan Google yaitu *Google Machine Learning Kit*. *Google Machine Learning Kit* merupakan salah satu produk Google yang menghadirkan keahlian dari *Machine Learning* untuk aplikasi android dan iOS. *Google Machine Learning Kit* memiliki kemampuan yang baik karena dibuat oleh para ahli *Machine Learning* dari Google. *Google Machine Learning Kit* dapat digunakan oleh semua kalangan secara gratis dan mudah untuk diintegrasikan karena memiliki dokumentasi yang cukup lengkap dan mudah untuk dimengerti. *Google Machine Learning Kit* memiliki beberapa model yang dapat digunakan diantaranya *text recognition, face detection, pose detection, selfie segmentation, barcode scanning, image labeling, object detection and tracking* dan *digital ink recognition* [16].

2.19 Face Recognition

Face recognition adalah salah satu teknologi yang memanfaatkan teknik biometrik yang dipergunakan untuk mengidentifikasi identitas seseorang melalui ciri – ciri unik yang terdapat pada citra wajah seseorang. Saat ini penggunaan sistem *face recognition* telah banyak digunakan untuk menangani berbagai permasalahan salah satunya adalah untuk melakukan proses presensi agar tidak terjadi dan mencegah kecurangan dalam proses presensi itu sendiri [24]. Sebagai sistem biometrik, sistem *face recognition* memiliki beberapa tahapan yang harus dilalui agar hasil dari pengenalan dari citra wajah yang dilakukan akurat, diantaranya:

1. *Face recognition* (pengenalan wajah)

Merupakan sebuah teknik atau proses yang dilakukan untuk membandingkan sebuah citra wajah dengan database wajah hingga menemukan wajah yang paling cocok dengan citra masukan tersebut.

2. *Face authentication* (autentikasi wajah)

Merupakan sebuah proses yang dilakukan untuk menguji keaslian atau kesamaan wajah dengan inputan sebelumnya.

3. *Face localization* (lokalisasi wajah)

Merupakan sebuah proses pendeteksian wajah dengan asumsi hanya terdapat satu wajah yang terdapat didalam citra tersebut.

4. *Face tracking* (penjajakan wajah)

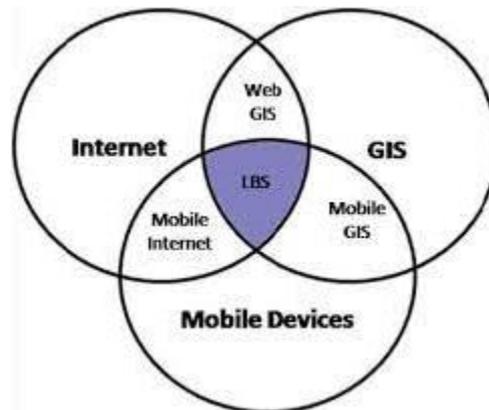
Merupakan sebuah proses untuk memperkirakan suatu wajah secara *real time*.

5. *Facial expression recognition* (pengenalan ekspresi wajah)

Merupakan sebuah proses yang dilakukan untuk mengetahui ekspresi dari wajah seseorang.

2.20 Location Based Service

Location Based Services merupakan suatu layanan informasi yang berfokus bagaimana menentukan posisi dari sebuah objek dengan melihat setiap perubahan secara aktif sehingga mampu mendeteksi letak dari objek tersebut atau bisa disebut dengan metode *positioning*. Terkait dengan metode *positioning*, ada beberapa sistem yang dapat digunakan, diantaranya, secara manual, melalui *Cellular Based Station*, dan melalui *Global Positioning System*, ketiga jenis sistem tersebut dapat digunakan untuk mendapatkan letak posisi perangkat, hanya saja yang membedakan dari ketiganya adalah cara kerja sistem tersebut untuk mendapatkan letak posisi dari suatu perangkat.



Gambar 2.4 Teknologi *Location Based Services*

Berdasarkan Gambar 2.4 Teknologi *Location based service* dapat digambarkan sebagai suatu layanan yang berada pada pertemuan diantara tiga teknologi, yaitu, *Geographic Information System*, *Internet Service*, dan *Mobile Device*. Selain itu didalam teknologi *location based service* ini terdapat dua unsur utama yang harus terpenuhi, diatanya:

1. *Location Manager*

Merupakan sebuah *tools* yang berguna untuk menampilkan sebuah *interface* berupa tampilan maps yang didalamnya menyediakan berbagai fitur yang berguna untuk melakukan manipulasi tampilannya, Salah satu *tools* yang biasanya sering digunakan adalah Google Maps API.

2. *Location Provider*

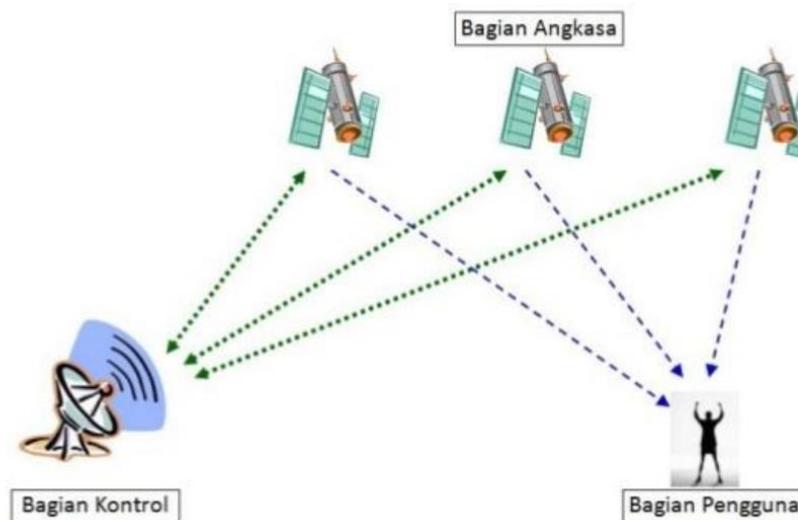
Merupakan sebuah *service* atau teknologi yang berguna untuk mencari titik lokasi dari sebuah perangkat secara *real time* berdasarkan pergerakan atau perpindahan dari satu tempat ke tempat lainnya, teknologi *location provider* ini umumnya sudah terdapat di seluruh perangkat saat ini yang diberinama GPS.

Adapaun komponen – komponen yang harus tersedia pada saat penggunaan teknologi *location based service* ini, diantaranya, *Mobile Device*, *Communication Network*, *Positioning Component*, *Service and Application Provider*, dan *Data and Content Provider* [2], [6], [25].

2.21 *Global Positioning System (GPS)*

Global position system atau yang sering disingkat GPS merupakan sebuah sistem navigasi yang memanfaatkan informasi melalui satelit. Informasi yang diterima tersebut diterima melalui antena – antena yang terdapat pada perangkat melalui sinyal – sinyal gelombang yang dikirim dari satelit yang berguna untuk dalam memperhitungkan letak posisi dari keberadaan suatu perangkat, kecepatan, maupun waktu.

Cara kerja dari GPS itu sendiri bukan mentransmisikan informasi dari suatu posisi, melainkan mentransmisikan posisi dari satelit dengan jarak penerima GPS dari satelit. GPS saat ini sangat berguna dalam kehidupan sehari – hari, sebagai contoh penggunaan navigasi untuk menentukan arah, untuk proses indentifikasi dan verifikasi pada saat proses presensi untuk mengetahui letak posisi seseorang pada saat melakukan presensi dan masih banyak lagi yang dapat dilakukan dengan memanfaatkan teknologi GPS ini [7]. Untuk lebih jelasnya mengenai bagaimana cara kerja GPS dapat dilihat pada Gambar 2.5 Cara Kerja GPS.



Gambar 2.5 Cara Kerja GPS

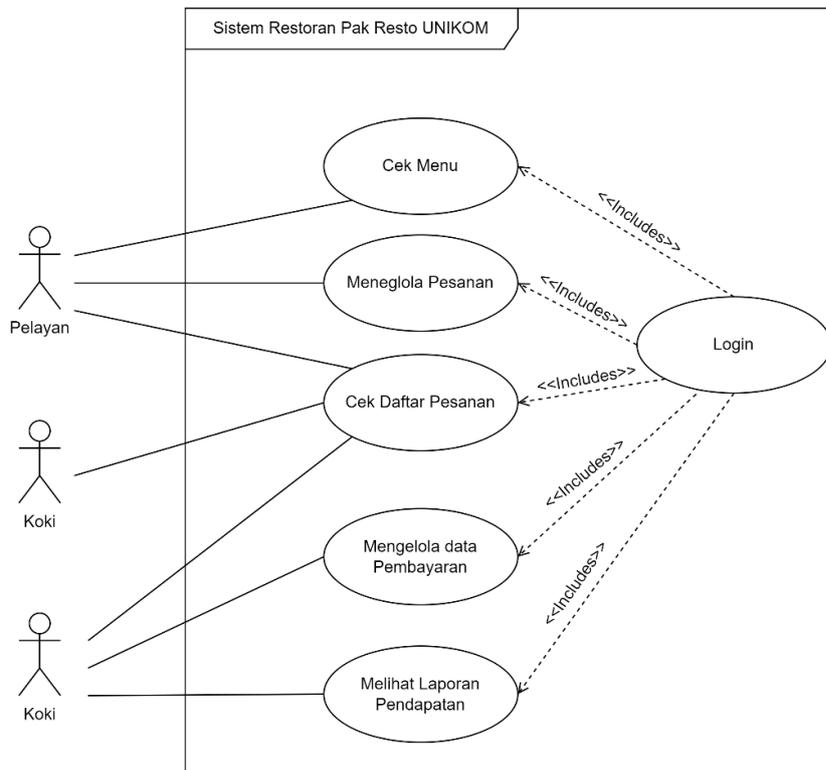
2.22 *Unified Modelling Language (UML)*

Unified Modelling Language (UML) adalah bahasa pemodelan yang dibuat dengan tujuan untuk memberikan cara standar dalam menggambar, membangun dan mendokumentasikan suatu sistem perangkat lunak. Di dalam pemodelan UML

terdapat 4 standar model yang digunakan yaitu *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram* dan *Class Diagram* [26].

2.22.1 Use Case Diagram

Use case diagram merupakan representasi dari interaksi yang terjadi antara pengguna dengan sistem yang di ilustrasikan dalam bentuk diagram. Selain itu *use case diagram* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem yang dibangun serta siapa saja yang berhak mengakses atau menggunakan *module* dari fungsi tersebut [26]. Pada Gambar 2.6 Contoh Use Case Diagram menggambarkan bagaimana aktor terlibat kedalam berbagai macam kejadian di dalam sebuah sistem absensi:



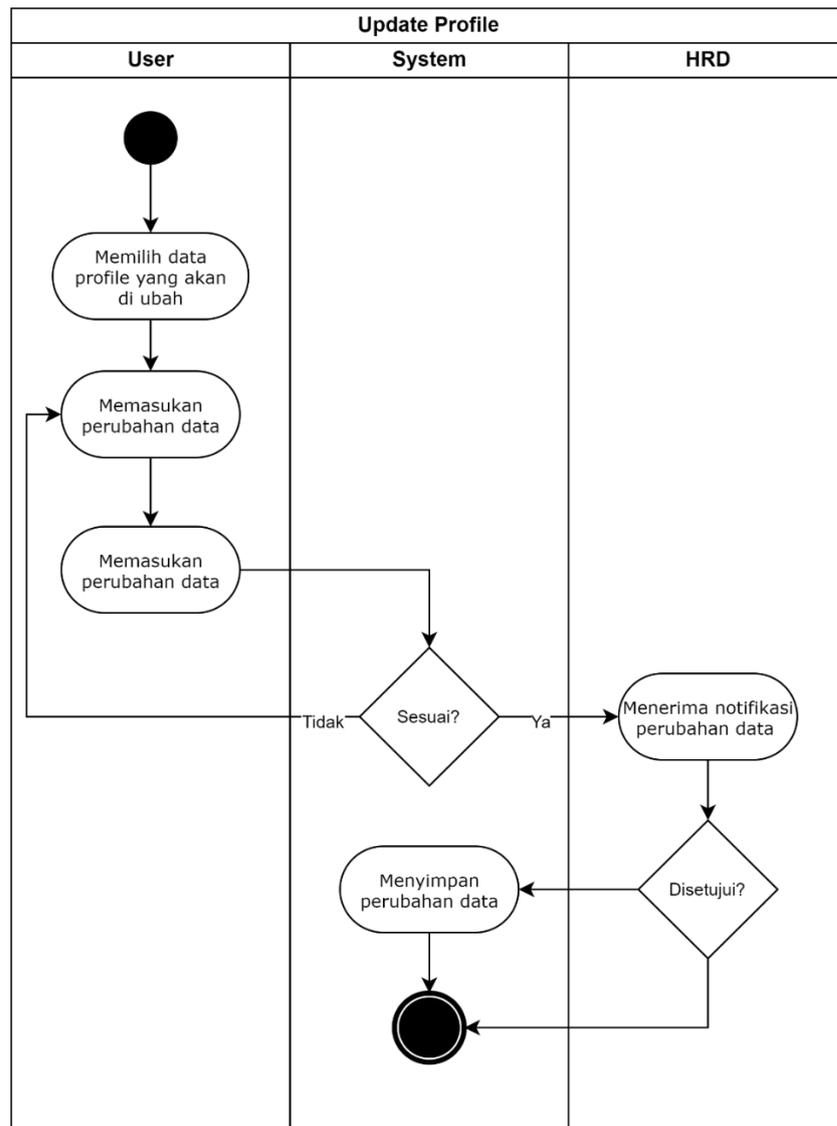
Gambar 2.6 Contoh Use Case Diagram

2.22.2 Activity Diagram

Activity Diagram merupakan representasi dari sebuah kegiatan dan Tindakan dengan dukungan pilihan, iterasi, dan konkurensi yang terjadi pada

sebuah sistem dari awal berjalannya suatu proses hingga berakhirnya suatu proses untuk mencapai suatu tujuan yang telah ditentukan [26].

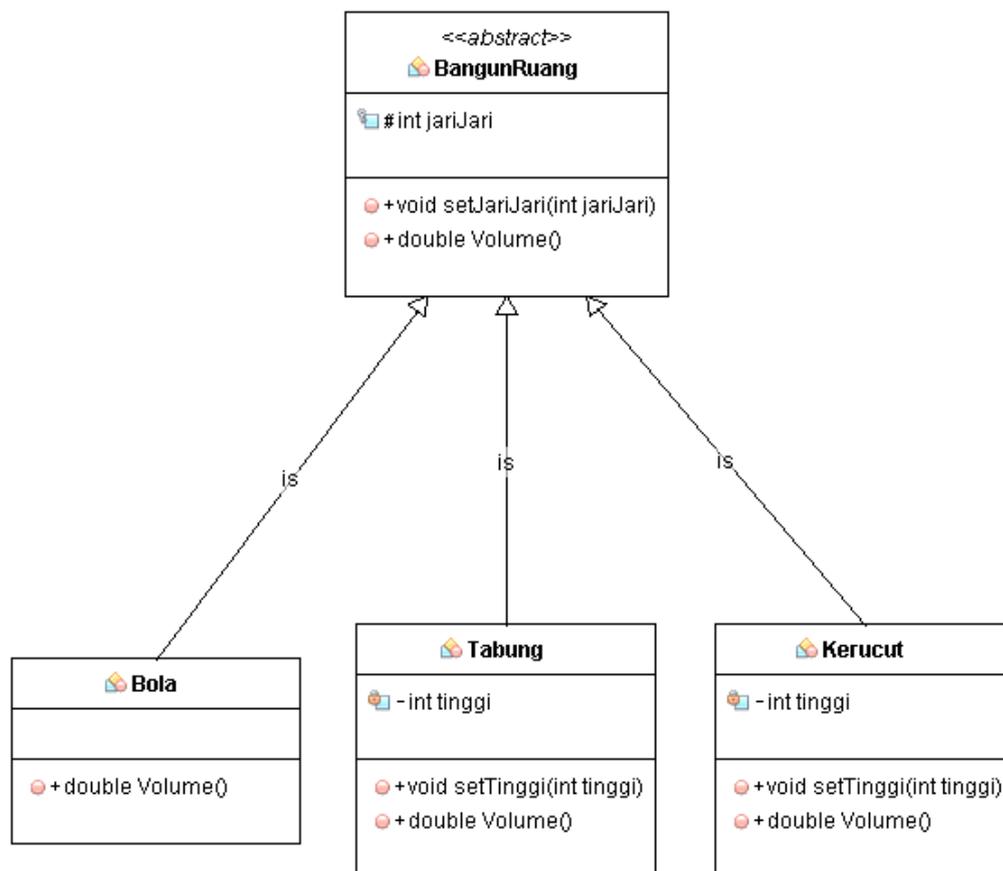
Pada Gambar 2.7 Contoh Activity Diagram menunjukkan tentang bagaimana sebuah tindakan yang dilakukan oleh pengguna dan respon dari sistem terhadap tindakan yang dilakukan pengguna secara berurutan yang disertai dengan validasi untuk menentukan tindakan apa yang selanjutnya harus dijalankan.



Gambar 2.7 Contoh Activity Diagram

2.22.3 Class Diagram

Class Diagram merepresentasikan struktur sistem yang dimulai dari pendefinisian kelas, atribut dan metode atau operasi. Selain itu *Class Diagram* merupakan pondasi utama dalam pemodelan berorientasi objek yang digunakan untuk membuat sebuah rancangan kelas – kelas yang akan diprogram yang didalamnya memiliki berbagai macam method atau fungsi yang dapat dijalankan [26]. Untuk lebih jelasnya dapat dilihat pada Gambar 2.8 Contoh Class Diagram yang merupakan salah satu contoh dari *Class Diagram*:

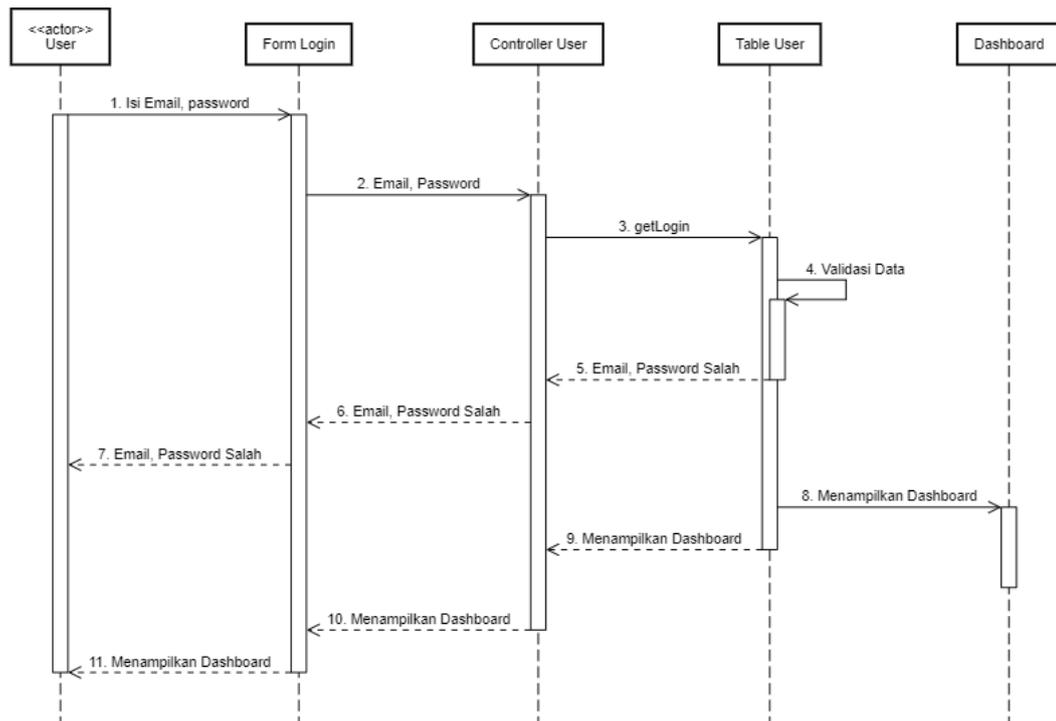


Gambar 2.8 Contoh *Class Diagram*

2.22.4 Sequence Diagram

Sequence Diagram merupakan representasi dari sebuah objek yang terlibat dalam skenario yang dijalankan secara berurutan dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Urutan interaksi

pada *Sequence Diagram* terjadi ketika *use case* tertentu dieksekusi [26]. Pada Gambar 2.9 *Sequence Diagram* dapat dilihat bahwa setiap proses yang terjadi di dalam *Sequence Diagram* dilakukan secara berurutan sesuai dengan urutannya dan setiap objek saling berinteraksi satu sama lain.



Gambar 2.9 Contoh *Sequence Diagram*