

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Teori Umum

##### 2.1.1 Aplikasi

Menurut (Rubiati & Harahap, 2019) aplikasi merupakan sebuah program yang sudah di algoritmakan agar dapat digunakan untuk sasaran kepentingan tertentu. Sedangkan menurut (Kristiadi & Supriyanti, 2017) aplikasi adalah sebuah perangkat lunak yang sudah dirancang sesuai dengan perintah kebutuhan pengguna. Sedangkan menurut (Suhimarita & Susianto, 2019) aplikasi merupakan sebuah komponen yang bisa digunakan untuk pengolahan kata atau dokumen lainnya. Defenisi lain dari aplikasi yaitu perangkat lunak yang sudah diprogram sesuai kebutuhan pengguna dengan tujuan untuk hasil yang lebih baik dari pembuat aplikasi (Novendri, Saputra, & Firman, 2019). Dalam hal ini sebuah perangkat lunak yang sudah terinstal pada sebuah *hardware* untuk dapat dioperasikan sesuai kebutuhan.

Berdasarkan dari beberapa pendapat di atas, maka penulis menyimpulkan aplikasi adalah program yang siap digunakan *user* (pengguna) yang dirancang dan dibuat dengan menggunakan algoritma bahasa pemograman.

### **2.1.2 Absensi (sistem informasi)**

Absen adalah tidak hadirnya pegawai, karyawan pada saat kerja dalam sebuah instansi atau perusahaan. Sedangkan, absensi merupakan sebuah *list* kehadiran yang isinya memuat jam masuk, jam pulang pegawai atau karyawan pada sebuah instansi atau perusahaan (Akbar & Prabowo, 2015).

Menurut (Adikara, 2013) sistem adalah interaksi antara beberapa unit yang saling berhubungan satu sama lain yang tidak bisa dipisahkan dalam satu kesatuan untuk mencapai satu tujuan yang ditetapkan.

### **2.1.3 Sistem Operasi *Android***

Menurut (Simanjuntak & Alfisyahri, 2020) *android* merupakan sistem operasi terbuka yang dipopulerkan oleh perusahaan google. Sedangkan menurut (Hakky et al., 2018) *android* adalah sebuah sistem operasi berbasis *linux* yang bisa diaplikasikan pada telepon canggih atau *tablet*. Berdasarkan pendapat dari beberapa peneliti diatas, maka penulis menyimpulkan bahwa sebuah *platform* terbuka yang dapat digunakan untuk membuat dan mengembangkan aplikasi perangkat lunak.

Perkembangan *android* karena bersifat terbuka, tidak menutup kemungkinan untuk seorang programmer memanfaatkan dalam menciptakan sebuah perangkat lunak atau *software* sesuai kebutuhan pengguna.

### **2.1.4 Cuti**

Mengacu pada UU Ketenagakerjaan No. 13 Tahun 2003 jo Pasal 79 ayat (3), UU Cipta Kerja 11 Tahun 2020, pekerja berhak atas cuti tahunan 12

hari kerja setelah pekerja/buruh yang bersangkutan bekerja selama 1 tahun/12 bulan secara terus menerus (Sunarya, Abas, et al., 2019). Ada beberapa jenis cuti pada umumnya sesuai UU Ketenagakerjaan menurut (Hadiyanto, 2018) yaitu :

1. Cuti Tahunan, untuk pelaksanaan cuti tahunan memiliki syarat salah satunya ialah pegawai atau karyawan telah bekerja selama 12 (dua belas) bulan bekerja disuatu instansi atau perusahaan. Dan pegawai atau karyawan wajib memperoleh gaji selama proses cuti.
2. Cuti Besar, syarat untuk mendapatkan cuti besar bagi pegawai atau karyawan ialah telah bekerja minimal 5 (lima) tahun secara terus menerus pada sebuah instansi atau perusahaan. Dan cuti besar diberikan paling lama 3 (tiga) bulan.
3. Cuti sakit, untuk pelaksanaan cuti sakit pegawai atau karyawan khusus yang sudah mengalami kecelakaan, penderita penyakit lain dan harus membutuhkan perawatan. Maka diberikan selama 14 (empat belas) hari kerja dengan melampirkan surat keterangan sakit dari rumah sakit. Dan pegawai atau karyawan tetap memperoleh gaji.
4. Cuti karena alasan penting, misalnya ada suami, istri, anak yang sakit atau meninggal dunia. Maka diberikan cuti paling lama 1 (satu) bulan.
5. Cuti Bersama, merupakan hak yang ditetapkan oleh Presiden, dan tidak tentu berapa lama. Maka pegawai atau karyawan tetap memperoleh gaji.
6. Cuti diluar tanggungan negara. Ini merupakan cuti yang memiliki alasan penting untuk sebuah instansi.

7. Cuti melahirkan, khusus perempuan yang melahirkan, diberikan selama 3 (tiga) bulan, dan berhak mendapatkan gaji.

Berdasarkan teori diatas penulis menyimpulkan bahwa cuti adalah keadaan atau kondisi pegawai atau karyawan pada sebuah instansi tidak masuk kerja di hari kerja yang ditentukan.

### **2.1.5 *Global Positioning System (GPS)***

Menurut (Budiwati, 2016), pengertian *GPS (Global Positioning System)* adalah penentuan posisi yang menggunakan satelit dengan sistem radio navigasi. Awalnya *GPS* digunakan untuk aplikasi militer oleh Departemen Pertahanan Amerika Serikat yang terdiri dari jaringan 24 satelit, dan pada tahun 1980-an pemerintah menyediakan untuk penggunaan komersil. Untuk cara kerja *GPS* pada dasarnya yaitu mengukur jarak antara penerima dengan satelit yang melibatkan 3 (tiga) bagian penting didalamnya adalah satelit, segmen pengguna dan segmen kontrol. Berikut adalah penjelasan singkat dari 3 (tiga) bagian dari *GPS* yaitu :

1. Satelit, salah satu alat yang sudah terpasang dari luar angkasa untuk mengirimkan sinyal ke bumi, dan sudah dilengkapi pengaturan waktu tujuannya agar mengetahui berapa lama waktu kerja sinyal satelit yang dikirimkan dan diterima pengguna.
2. Segmen pengguna, sebuah tindakan yang diberikan pengguna untuk menggunakan dan menerima sinyal untuk akses *GPS*.

3. Segmen kontrol, pada segmen ini tujuannya untuk memastikan apakah satelit dapat berfungsi dengan baik, dan alat yang digunakan sudah dipasang dibumi.

Masalah navigasi dan posisi letak sangatlah penting dalam berbagai aktifitas misalnya posisi perjalanan, lokasi, dan beberapa instansi lainnya. *Global Positioning System (GPS)* merupakan solusi yang tepat untuk mendapatkan lokasi yang akurat tinggi karena sudah dianggap memenuhi syarat E911 (Zandbergen & Barbeau, 2011).

Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang diberi nama *GPS reciever* yang berfungsi untuk menerima sinyal yang dikirim dari satelit GPS. Posisi di ubah menjadi titik yang dikenal dengan nama *way-point*. *Way-point* tersebut berupa titik-titik koordinat garis lintang dan garis, pada saat ini *GPS* sudah banyak digunakan dari berbagai aplikasi untuk mendapatkan lokasi.

## **2.2 Teori Khusus**

### **2.2.1 *QR (Quick Response) Code***

Menurut (Singh, 2016), *QR Code* atau kode respon cepat merupakan sebuah gambar berbentuk segi empat dua dimensi yang memuat informasi bisa gambar, video, teks dan lainnya. Penelitian selanjutnya, mendefenisikan kode *QR* yaitu sebuah kode batang dua dimensi yang dikembangkan oleh Denso Wave dan dipublikasikan pada tahun 1994. Sedangkan menurut (Wijaya & Gunawan, 2016), kode *QR (Quick Response)* adalah sebuah kode

dua dimensi yang memiliki pola piksel warna hitam dan putih yang mampu menyimpan 2089 digit atau 4289 karakter, termasuk tanda baca ataupun karakter spesial di dalamnya. Dengan keunggulan tersebut, maka *QR code* mampu menampilkan berbagai teks, membuka *URL*, menyimpan kontak pada buku telepon, dll.

Pada umumnya, untuk membuka kode *QR* ini memerlukan perangkat scan atau pemindaian lewat telepon canggih (*smartphone*). Ada 2 (dua) jenis *QR Code* yaitu :

1. *QR Code* statis, merupakan sebuah kode yang dibuat dan tidak bisa dirubah, termasuk apa saja informasi yang terdapat didalam kode tersebut, atau dengan istilah lain kode *QR* statis ini cocok untuk penggunaan privat.
2. *QR Code* dinamis, defenisinya kebalikan dari *QR code* statis yaitu sebuah kode yang bisa dirubah kembali sesuai keperluan.

Berikut adalah gambar beberapa tipe kode *QR Code* yang dilansir dari (Denso Wave, 2016) :



**Gambar 2. 1** Model 1 & 2  
Sumber : *qr code.com*



**Gambar 2. 2** Kode *QR* Mikro  
Sumber : *qr code.com*



**Gambar 2. 4** *SQRC*  
Sumber : *qr code.com*



**Gambar 2. 3** Kode *IQR*  
Sumber : *qr code.com*



**Gambar 2. 5** *Frame QR*  
**Sumber :** *qr code.com*

Dari gambar diatas memiliki penjelasan sebagai berikut :

1. Model 1 dan 2, pada model 1 versi terbesar adalah 14 (73x73 modul) yang mampu menyimpan angka hingga 1.167. Model 2 merupakan pengembangan dari model 1 dengan versi terbesar 40 (177x177 modul) yang mampu menampung angka 7.089.
2. Kode *QR* Mikro, adalah fitur satu pola untuk mendeteksi orientasi. Versi terbesar dari kode ini adalah M4 (17x17 modul) yang dapat menampung hingga 35 angka.
3. Kode *IQR*, adalah fitur yang dibuat dalam bentuk persegi Panjang dengan versi maksimum 61 (422x422 modul) yang dapat menampung 40.000 angka.
4. *SQRC*, adalah fitur yang dapat digunakan untuk menyimpan data pribadi dan fungsinya hampir sama dengan kode *QR* biasa.
5. Bingkai *QR*, adalah sebuah fitur yang memuat gambar pada area canvas.

### **2.2.2** *Lock GPS (Global Positioning System)*

*Lock GPS* yaitu keadaan sebuah *GPS* penerima (*receiver*) yang sudah terkoneksi dengan sinyal satelit dan menangkap kembali informasi dari satelit. Tujuannya untuk mengetahui lokasi dengan akurasi yang tinggi.

Sebuah *GPS receiver* setidaknya harus mengunci sinyal minimal tiga buah satelit yang dipergunakan untuk menghitung posisi 2D (*latitude* dan *longitude*) dan jalur pergerakan (Akbar & Prabowo, 2015). Salah satu fungsi *GPS locker* yaitu salah satu alat bantu untuk mendapatkan sinyal *GPS* ditempat yang susah.

Pada saat ini, sudah banyak *GPS* dipasang pada *smartphone* untuk mendeteksi keberadaan lokasi, teknologi *GPS* dirancang dengan memanfaatkan satelit yang ada di luar angkasa. Satelit ini akan mengirimkan sinyal untuk mendeteksi keberadaan seseorang atau lokasi suatu tempat tertentu.

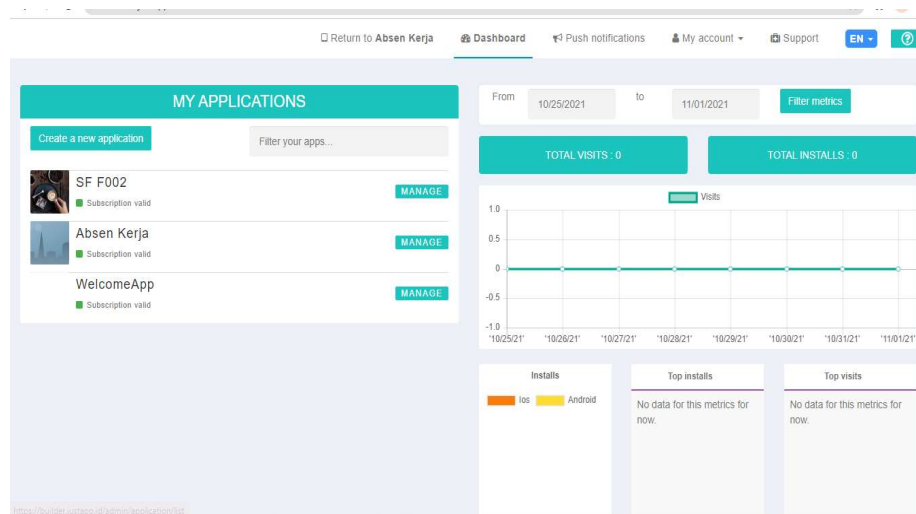
Menurut (Alfeno & Devi, 2017) ada beberapa kategori terkait penggunaan *GPS* yaitu :

1. Waktu, memberikan akurasi waktu yang tinggi dari *GPS* penerima.
2. Lokasi, *GPS* memberikan informasi lokasi, *latitude*, *longitude* dan *altitude*.
3. Kecepatan, dengan adanya perpindahan lokasi atau tempat maka *GPS* akan dapat memberikan informasi dari kecepatan perpindahan tersebut.
4. Rute, *GPS* dapat menunjukkan tujuan perjalanan ke lokasi tujuan.
5. Komulasi data, sebuah informasi yang tersimpan ke *GPS receiver*, baik itu perjalanan yang pernah dilalui, kecepatan rata-rata dan waktu yang sampai ketempat tujuan.
6. *Tracking*, digunakan untuk memantau objek.



### 2.2.3 JustApp

*Justapp* adalah sebuah *platform* yang berbasis web yang dapat digunakan untuk membuat aplikasi seluler *android/ios* tanpa coding. *Justapp* ini salah satu *platform* buatan anak bangsa dari Indonesia yang berguna untuk membantu UMKM dalam meningkatkan penjualan produk, sehingga menambah nilai untuk pendapatan (Team, 2020). Jadi, peneliti memanfaatkan *platform* ini sebagai alat untuk membuat aplikasi absensi yang digunakan pada toko cahaya elektronik. Pada *platform* ini menyediakan beberapa fitur untuk digunakan sesuai kebutuhan, dan menyediakan database yang mengikuti sistem *justapp* itu sendiri. Untuk pembuatan aplikasi dari *platform justapp* yaitu dengan melakukan *drag and drop*. Berikut adalah tampilan atau halaman utama dari *justapp*.



**Gambar 2. 6** Halaman utama *justapp*  
Sumber : *justapp.id*

#### 2.2.4 *Cascading Style Sheet (CSS)*

*Cascading Style Sheet (CSS)* adalah sebuah bahasa pemrograman yang digunakan untuk merubah tampilan web serta memisahkan antara konten dan desain (Rahmad Basuki, 2013). Sedangkan, menurut (Novendri et al., 2019) *CSS* merupakan suatu atribut yang mengontrol tampilan dokumen secara bersamaan.

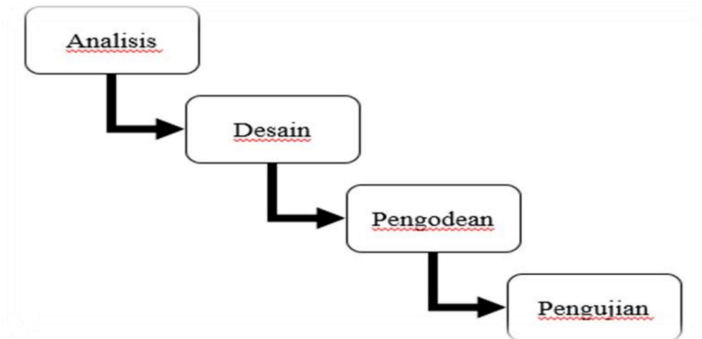
Berdasarkan pendapat diatas maka penulis menyimpulkan bahwa *css* ini bisa dimanfaatkan untuk merubah tampilan web menjadi lebih baik, karena bisa membentuk tampilan sesuai keinginan pengguna. Dan *CSS* juga bisa diartikan sebagai bahasa desain terhadap suatu tampilan web dengan tujuan untuk merubah tampilan web agar lebih menarik. Ada beberapa jenis *CSS* yaitu :

1. *Inline style sheet*, untuk penulisan *cssnya* ditulis langsung di tag *HTML (HyperText Markup Language)*.
2. *Internal style sheet*, untuk penulisannya ditulis antara tag *head* dengan diikuti tag *style*.
3. *External style sheet*, untuk penulisan ditulis secara terpisah dengan format *.css* pada nama file.

#### 2.2.5 Metode Waterfall

Metode penelitian yang digunakan peneliti pada pembuatan perangkat lunak sistem aplikasi ini yaitu model *SLDC* air terjun (*waterfall*), pada model

aliran seperti air terjun ini menyediakan pendekatan secara bertahap dan terurut (S. & Shalahuddin, 2011). Berikut adalah gambar model *waterfall* :



**Gambar 2. 7** Model *Waterfall*

**Sumber :** S. & Shalahuddin, 2011

Berdasarkan gambar di atas, dapat dijelaskan bahawa tahapan dari perancangan sistem dengan menggunakan metode air terjun atau *waterfall* adalah :

1. Analisis

Tahap awal untuk proses pengumpulan data untuk menspesifikasikan kebutuhan aplikasi yang dipahami pengguna, identifikasi masalah dengan tujuan untuk mendapatkan solusi dari permasalahan yang ada. Pada tahap ini menguraikan desain yang akan diimplementasikan secara teknis.

2. Desain perangkat lunak

Pada tahap kedua ini, menyangkut tentang proses yang berfokus pada pembuatan program perangkat lunak termasuk arsitektur perangkat lunak, representasi antar muka, dan prosedur pengkodean. Sehingga, pada untuk tahap proses selanjutnya akan menjadi acuan untuk pembuatan kode aplikasi.

3. Pembuatan kode program

Tahap ini memuat *coding* yang digunakan pada komputer untuk membuat program perangkat lunak sesuai desain pada tahap kedua. Peneliti menggunakan *coding CSS* dari *platform justapp* dalam pembuatan aplikasi agar tampilan lebih menarik.

#### 4. Pengujian

Tahap pengujian ini berfokus pada perangkat lunak aplikasi. Peneliti menggunakan *blackbox* untuk meminimalisir kesalahan pada *coding*, tujuan pengujian untuk memperbaiki kesalahan (*error*).

#### 5. Pemeliharaan (*maintenance*)

Pada tahap ini, tidak menutup kemungkinan ketika terjadi perubahan pada perangkat lunak yang dibagikan ke pengguna, bisa dari kesalahan yang tidak ditemukan pada proses pengujian atau perangkat lunak yang dibagikan harus menyesuaikan dengan lingkungan. Maka tahap inilah perangkat lunak tersebut diperbaiki kembali.

### 2.2.6 Analisis perancangan




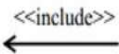
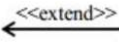

Menurut (Sri Mulyani NS, 2016) kepanjangan dari *UML* yaitu *Unified Modeling Language*, merupakan suatu metode atau teknik pemodelan secara visual untuk perancangan sebuah sistem berorientasi objek.

Perancangan sistem aplikasi absensi berbasis *qr code* dirancang dengan menggunakan diagram *UML (Unified Modeling Language)* yang terdiri dari :

Berikut merupakan bagian-bagian dari diagram *UML* untuk menunjukkan simbol elemen aspek sistem yang dibuat,

1. *Use Case* Diagram merupakan sebuah gambaran interaksi antara satu aktor dengan sistem yang dibuat. Dalam hal ini adanya keterkaitan antara pengguna untuk mendapatkan informasi yang ada didalam sistem tersebut. Berikut adalah notasi atau simbol dari *use case*, yaitu :





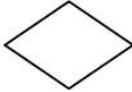
**Tabel 2. 1** Notasi *usecase*

Notasi	Nama	Fungsi
	Aktor	Menggambarkan apa yang harus dikerjakan pada sistem
	<i>Use case</i>	Menjelaskan apa saja kegiatan yang dilakukan oleh aktor pada sistem
	Relasi asosiasi	Menggambarkan hubungan antar aktor dengan <i>use case</i> dengan tanda panah
	<i>Include</i>	Sebuah pilihan yang digunakan satu <i>use case</i> dengan <i>use case</i> lainnya untuk fungsionalitas
	<i>Extend</i>	Menunjukkan tambahan sebuah <i>use case</i> dengan <i>use case</i> lainnya jika kondisi terpenuhi
	Generalisasi	Menggambarkan <i>use case</i> yang mempunyai persamaan

Sumber: *Star uml*

2. *Activity* Diagram adalah suatu yang menggambarkan aktivitas dari proses pelaksanaan aliran kerja dalam sebuah sistem. Berikut adalah tabel notasi *activity* diagram, yaitu :

**Tabel 2. 2** Notasi *activity* diagram

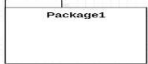


Notasi	Nama	Fungsi
	<i>Start state</i>	Awal mulai kegiatan aliran kerja.
	<i>End state</i>	Menggambarkan aliran kerja berakhir.
	<i>State</i>	Menunjukkan kegiatan alur kerja pada sistem.
	<i>Transisi</i>	Menggambarkan aliran kerja kegiatan.
	<i>Decision</i>	Menggambarkan keputusan yang dibuat dalam aliran kerja

Sumber : *Star UML*

3. *Package* Diagram adalah diagram yang mengelompokkan paket kelas.

Berikut adalah tabel *package* diagram, yaitu :



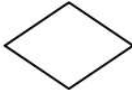




**Tabel 2. 3** Notasi *package* diagram

Notasi	Nama	Fungsi
	<i>Package</i>	Sekumpulan elemen model
	<i>Import</i>	Menunjukkan indikasi pada yang ditambahkan pada sumber paket
	<i>Acces</i>	Menunjukkan indikasi pada nama sumber paket

Sumber : *Diagram UML*

4. *State Diagram* adalah menggambarkan semua kondisi antara suatu objek dari suatu *class* dan keadaan yang menyebabkan *state* berubah. Berikut adalah tabel *state diagram*, yaitu :


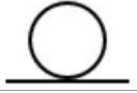




**Tabel 2. 4** Notasi *state diagram*

Notasi	Nama	Fungsi
	<i>Simple State</i>	Menunjukkan kegiatan alur kerja pada sistem.
	<i>Transisi</i>	Menggambarkan aliran kerja kegiatan.
	<i>Choice</i>	Menggambarkan keputusan yang dibuat dalam aliran kerja .
	<i>Start state</i>	Awal mulai kegiatan aliran kerja.
	<i>End state</i>	Menggambarkan aliran kerja berakhir.
	<i>Join</i>	Untuk menggabungkan aktivitas yang berjalan bersamaan
	<i>Fork</i>	Untuk memisahkan aktivitas yang berjalan bersamaan

Sumber : *Start UML*

5. *Sequence Diagram* menjelaskan interaksi objek yang disusun berdasarkan waktu, serta bisa memberikan gambaran baik secara fisik dan logis antara objek.

Tabel 2. 5 Notasi *sequence diagram*

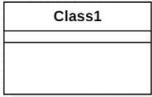
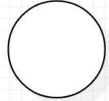
Notasi	Nama	Fungsi
	<i>Actor</i>	Menunjukkan seseorang yang berinteraksi dengan sistem
	<i>Entity Class</i>	Menunjukkan suatu hubungan yang diterapkan
	<i>Boundary Class</i>	Menggambarkan form
	<i>Control Class</i>	Menunjukkan penghubung antara <i>boundary</i> dengan tabel
	<i>A focus of Control and A life line</i>	Menunjukkan awal mulai dan berakhir pesan
	<i>Message</i>	Menunjukkan pengiriman komunikasi

Sumber : badoystudio.com

6. *Class Diagram* merupakan deskripsi kelompok objek dengan *property*, perilaku dan relasi yang sama, sehingga dapat membantu memvisualisasikan struktur kelas dari suatu sistem.






**Tabel 2. 6** Notasi *class diagram*

Notasi	Nama	Fungsi
	<i>Class</i>	Sekumpulan dari objek yang berbagi atribut serta operasi yang sama
	<i>Interface</i>	Menggambarkan antar muka sistem

**Sumber :** *Star UML*

7. *Communication Diagram* adalah diagram yang menggambarkan sebuah tahapan terjadinya aktivitas dengan tertuju pada objek masing-masing.

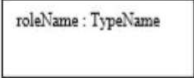

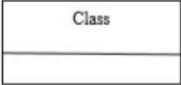
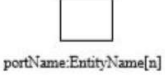
**Tabel 2. 7** Notasi *Communication diagram*

Notasi	Nama	Fungsi
	<i>Lifeline</i>	Objek dari sebuah <i>class</i> yang tersusun secara horizontal
	<i>Actor</i>	Menunjukkan seseorang yang berinteraksi dengan sistem
	<i>Message</i>	Pengiriman pesan

**Sumber :** *Star UML*

8. *Composite Structure Diagram* adalah diagram yang menggambarkan struktur internal dari klasifikasi *class*, *component* atau *usecase*.

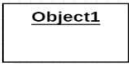
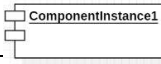
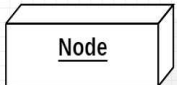

**Tabel 2. 8** Notasi *Composite Structure* diagram

Notasi	Nama	Fungsi
	<i>Property</i>	Satu set dari <i>instance</i>
	<i>Connector</i>	Menggambarkan komunikasi dari dua buah <i>instance</i>
	<i>Class</i>	Menunjukkan penjabaran sebuah struktur jika ada.
	<i>Port</i>	Menggambarkan <i>composite structure</i> diagram tanpa memunculkan detail internal dari sistem

Sumber : Ansoriweb.com

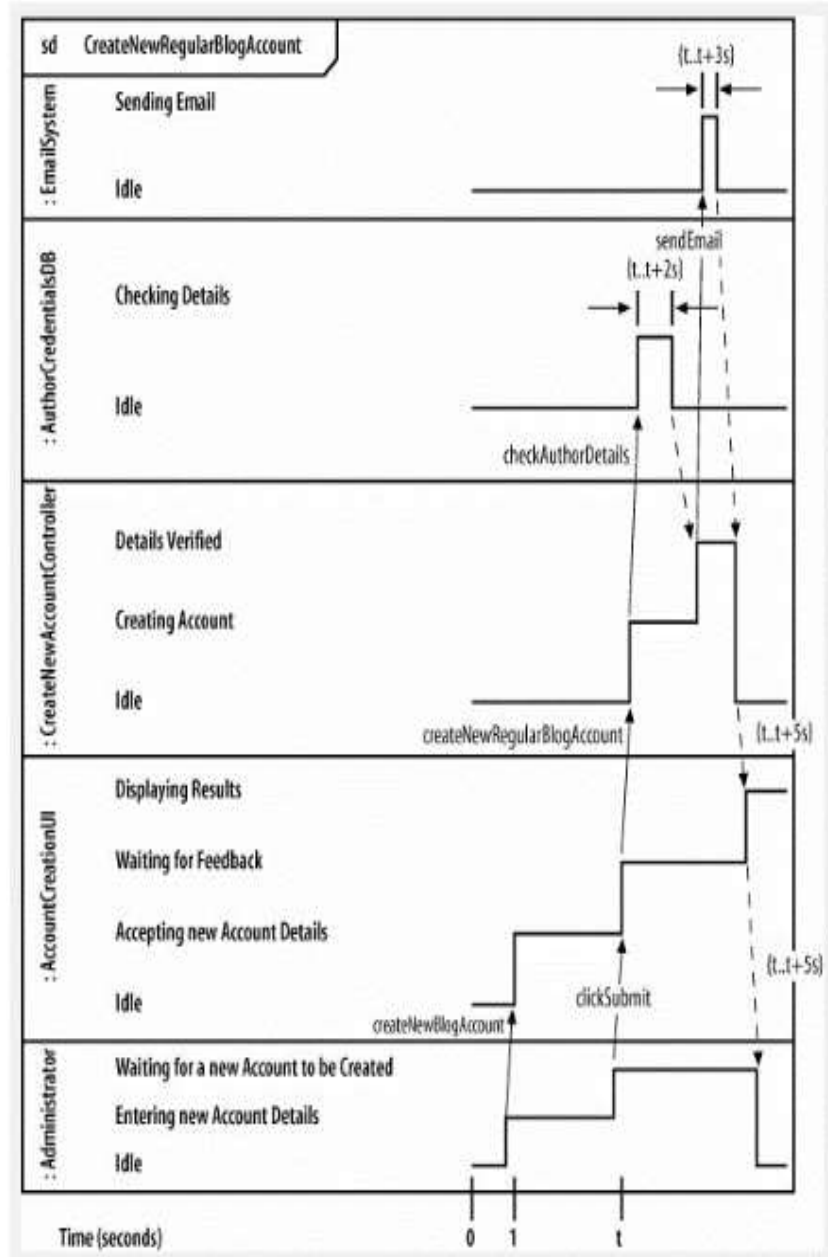
9. *Object* Diagram adalah diagram yang menggambarkan hubungan sistem pada objek.

**Tabel 2. 9** Notasi *object* diagram

Notasi	Nama	Fungsi
	<i>Object</i>	Objek dari kelas yang berjalan pada sistem
	Komponen	Komponen sistem
	<i>Node</i>	Merujuk kepada perangkat keras dan perangkat lunak, jika didalam node tercantum komponen maka harus sesuai dengan komponen sebelumnya pada diagram komponen.
	<i>Link</i>	Hubungan antar node

Sumber : sis.binus.ac.id


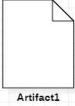

10. *Timing* Diagram adalah diagram yang menunjukkan perubahan waktu kondisi terhadap objek yang berbeda.



**Gambar 2. 8** Contoh *timing* diagram  
**Sumber :** sis.binus.ac.id

11. *Component* Diagram adalah diagram yang menghubungkan komponen dengan perangkat lunak .



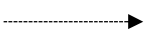
**Tabel 2. 10** Notasi *Component* Diagram

Notasi	Nama	Fungsi
	Komponen	Komponen sistem
	Artefak	Menggambarkan hubungan komponen dalam sebuah node
	<i>Interface</i>	Menggambarkan antar muka sistem

Sumber : sis.binus.ac.id

12. *Deployment* Diagram adalah diagram yang menunjukkan tata letak suatu sistem.



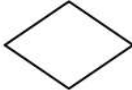


**Tabel 2. 11** Notasi *Deployment* Diagram

Notasi	Nama	Fungsi
	Komponen	Komponen sistem
	<i>Node</i>	Menggambarkan hubungan antar yang mengacu terhadap perangkat keras dan perangkat lunak
	<i>Dependency</i>	Menggambarkan arah antar node ke node lainnya

Sumber : sis.binus.ac.id

13. *Interaction Overview* Diagram adalah diagram yang menggambarkan hubungan *activity* diagram dengan *sequence* diagram.

**Tabel 2. 12** Notasi *Interaction Overview* Diagram

Notasi	Nama	Fungsi
	<i>Activity</i>	Menunjukkan yang sedang berlangsung
	<i>Control flow</i>	Menggambarkan aliran antar node
	<i>Decision</i>	Menggambarkan keputusan yang dibuat dalam aliran kerja .
	<i>Start state</i>	Awal mulai kegiatan aliran kerja.
	<i>End state</i>	Menggambarkan aliran kerja berakhir.

Sumber : Ansoriweb.com

### 2.2.7 *Black-box testing*

Menurut (Hendri et al., 2020) *black-box* merupakan sebuah perangkat lunak yang digunakan untuk mendeteksi kesalahan dari aplikasi yang dibuat, baik dari *interface*, model data serta akses ke sumber eksternal. Dari pengertian diatas penulis mendefenisikan *black-box* atau kotak hitam adalah suatu software yang dipakai untuk menguji aplikasi yang dibuat, dengan tujuan untuk mengetahui kesalahan dari sistem aplikasi tersebut.

Pengujian *black-box* memiliki beberapa cara untuk memastikan sistem mengetahui beberapa kesalahan dalam proses input, sehingga pengujian ini bisa mengatasi kesalahan sebagai berikut :

1. Fungsi yang salah atau menghilang dari perangkat lunak yang diuji.
2. Terdapat kesalahan antar muka dari perangkat lunak.
3. Permasalahan dari kinerja perangkat lunak.
4. Kesalahan dari terminasi perangkat lunak.

Berikut adalah beberapa teknik dalam pengujian *black-box*, menurut (Desember at el., 2017) yaitu : partisi kesetaraan (*equivalence partitioning*), analisis nilai batas (*boundary value analysis*), pengujian *robustness* (*robustness testing*), pengujian perilaku dan pengujian hubungan sebab-akibat (*behavior testing dan cause-effect relationship testing*).

### **2.3 Penelitian terdahulu**

Berikut adalah beberapa penelitian terdahulu yang menjadi inspirasi penulis dalam membuat penelitian tentang aplikasi absensi berbasis *QR Code*, yaitu :

1. (Akbar & Prabowo, 2015) dengan judul Aplikasi Absensi menggunakan metode *lock gps* dengan *android* di pt. pln (persero) app malang basecamp Mojokerto, diterbitkan oleh majalah *techno* (nama jurnal), vol 5, nomor 2, penelitian ini membahas masalah absensi pegawai yang memakan waktu untuk melakukan kehadiran karena jarak jauh, menggunakan metode

*usability testing* merupakan wawancara kepada pegawai, hasilnya menciptakan sebuah aplikasi yang bernama aplikasi mobile dengan memanfaatkan GPS.

2. (Sunarya, Febriyanto, & Januarini, 2019) dengan judul Aplikasi Mobile Absensi Karyawan Dan Pengajuan Cuti Berbasis *GPS*, diterbitkan oleh *CCIT Journal* vol 12, nomor 2, penelitian ini membahas masalah pengembangan absensi dengan menggunakan *GPS* yang sebelumnya tidak menggunakan *GPS* dan proses pengajuan cuti yang tidak efisien karena harus meminta persetujuan atasan ketika mengajukan cuti, menggunakan metode *SWOT (Strength, Weakness, Opportunity, Threat)*, hasilnya terbuatlah sebuah aplikasi yang dapat digunakan karyawan untuk absen dan mengajukan cuti.
3. (Putra, Haeruddin, & Puspitasari, 2019) dengan judul Aplikasi Absensi Perkuliahan Berbasis *Android* Studi Kasus Absensi FKTI Universitas Mulawarman, diterbitkan oleh *Jurnal Rekayasa Teknologi Informasi (JURTI)*, vol 3, nomor 1, penelitian ini membahas masalah rekap data kehadiran masih menggunakan manual, menggunakan metode *waterfall*, hasilnya Memudahkan mahasiswa melakukan absen dan memonitoring kehadiran mahasiswa.
4. (Adikara, 2015) dengan judul Pemanfaatan *Mac Address* Hostspot Dalam Pengembangan Sistem Absensi *GPS* Dalam Rangka Meningkatkan Keakuratan Posisi Pengguna, diterbitkan oleh *Sisfo* atau sistem informasi , vol 5, nomor 4, penelitian ini membahas masalah sulitnya mendapatkan

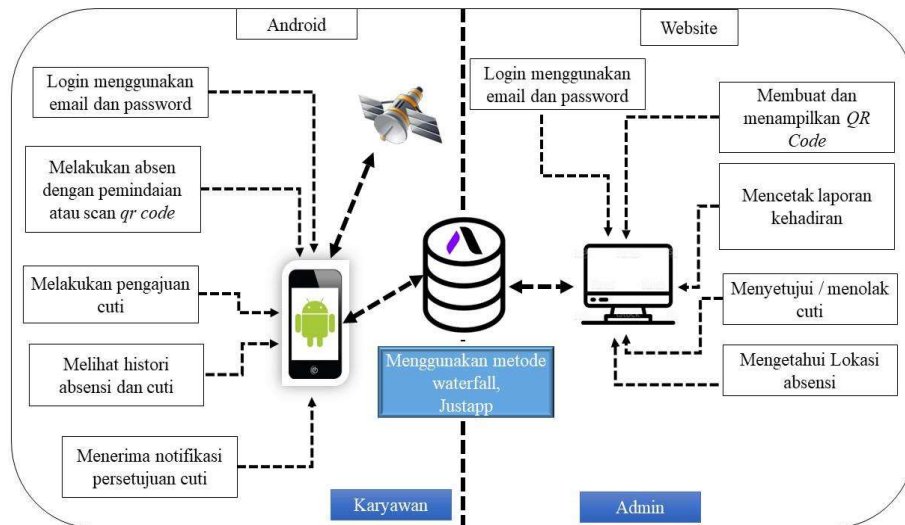
informasi lokasi keberadaan pengguna, menggunakan metode *SWOT* (*Strength, Weakness, Opportunity, Threat*), hasilnya Meningkatkan keakuratan titik lokasi pengguna ketika melakukan absensi berdasarkan *mac address* telepon canggih.

5. (Febriandirza, 2020) dengan judul Perancangan Aplikasi Absensi Online Dengan Menggunakan Bahasa Pemrograman Kotlin, diterbitkan oleh *Pseudocode* vol 7, nomor 2, penelitian ini membahas masalah timbulnya resiko kecurangan akibat penggunaan absensi manual, menggunakan metode *waterfall*, menghasilkan sebuah aplikasi absensi berbasis android dengan bahasa pemograman *kotlin*.
6. (Tambunan & Zetli, 2020) dengan judul Perancangan Aplikasi Absensi Karyawan Dengan Menggunakan Kode *QR* Berbasis *Android*, diterbitkan oleh *Comasie* vol 3, nomor 02, penelitian ini membahas masalah absen karyawan yang sering lupa, menggunakan metode *waterfall*, menghasilkan sebuah aplikasi absensi karyawan berbasis android.



## 2.4 Kerangka pemikiran

Berdasarkan penelitian yang penulis buat, maka dapat dilihat dari gambar kerangka pemikiran berikut ini :



**Gambar 2. 9** Kerangka Pemikiran  
**Sumber :** Data Peneliti, 2021

Berdasarkan gambar diatas maka proses pertama kali dilakukan oleh admin. Dimana admin dapat *login* menggunakan email dan password yang sudah terdaftar, kemudian menginput nama karyawan dan menampilkan barcode berdasarkan nama karyawan tersebut kedalam database dari platform *justapp*. Untuk proses cuti, admin menginput jenis-jenis cuti pada database *platform justapp* yang berlaku pada toko tersebut. Pengguna *android login* dengan email dan password yang sudah terdaftar, kemudian melakukan pemindaian atau scan pada *QR Code* yang telah disediakan admin untuk proses kehadiran.