

BAB II

KAJIAN PUSTAKA

2.1 Teori Dasar

Teori dasar berisi menyangkut teori-teori yang dipakai dari peneliti pada penelitian yang dilakukan. Teori dasar berfungsi sebagai pemberian gambaran tentang penelitian yang akan dilakukan dan mengembangkan ide-ide seorang peneliti. Teori dasar bisa diperoleh dari jurnal maupun buku-buku. Berikut teori-teori yang mendasar untuk penelitian ini:

2.1.1 Pengertian Aplikasi

Aplikasi adalah sejenis atau bentuk program yang dirancang khusus untuk dapat digunakan oleh *user*/pengguna sehingga bisa bermanfaat pada tujuan tertentu yang hendak dijangkau untuk kebutuhan *user*. Aplikasi juga dipakai untuk dapat dikembangkan dengan cara tertentu, sehingga menjadi subjek atau inti dari masalah yang akan diselesaikan melalui program tersebut dan mampu memudahkan segala bentuk pekerjaan pengguna (Wisnumurti et al., 2022). Klasifikasi berikut merupakan klasifikasi yang biasanya digunakan oleh perancang aplikasi:

- a. Perangkat lunak atau aplikasi khusus, adalah suatu bentuk program yang berfungsi sebagai dokumentasi dengan tujuan mengarahkan fungsi spesifik atau terpilih.

- b. Perangkat lunak maupun aplikasi pada wujud paket, adalah suatu program berfungsi sebagai dokumentasi dan berfungsi sebagai penyelesaian masalah tertentu.

2.1.2 Pengertian *Software development*

Software Development merupakan suatu perangkat lunak yang bertujuan untuk melakukan tugas secara sistematis. Fungsi utama *software development* adalah untuk dapat membangun dan mengembangkan sebuah *system* dengan berbagai informasi tentang perancangan yang sedang dikerjakan atau sebagai target yang akan dicapai (Khoir et al., 2020a). Berikut beberapa tim yang dapat digunakan untuk memudahkan seorang pekerja *software development*:

1. *UI/UX Designer* adalah orang-orang yang berkerja dan bertugas sebagai perancang desain mulai mulai semula sampai akhir produksi aplikasi.
2. *Front End Developer* adalah pekerja yang bertugas untuk mengubah gambar atau desain tim UI/UX ke dalam bentuk *code* program sehingga menghasilkan *output* pada aplikasi sesuai keinginan *client*.
3. *Back End Developer* adalah pekerja yang membuat database dengan menyediakan server bagi tim *front end* sehingga bisa dipakai oleh konsumen.
4. *Full Stack Developer* adalah pekerja yang melaksanakan kegiatan pada sisi *Front end* serta *Back End*.

2.1.3 Pengembangan Perangkat Lunak

Perkembangan suatu piranti lunak tidak dapat dilepaskan dari berbagai metode yang digunakan untuk menyelesaikan permasalahan pada pembuatan aplikasi. Berikut beberapa metode yang bermanfaat dan dapat digunakan oleh seorang programmer dalam mengembangkan perangkat lunak:

2.1.3.1 Metode *Waterfall*

Metode waterfall adalah metode awal dipakai pada pembuatan aplikasi dan kini sudah dianggap sebagai metode tradisional namun masih digunakan hingga sekarang. Metode waterfall memiliki beberapa tahap penyelesaian mulai dari tahap perancangan, analisis, desain dan implementasi. Dengan demikian, metode ini banyak menawarkan pengetahuan mulai oleh analisis, desain coding, pengujian/verifikasi, perbaikan pada pembuatan aplikasi (Qadafi & Wahyudi, 2020). Berikut ini tahapan-tahapan dalam menggunakan metode waterfall:

1. *Requirement Analysis*, merupakan langkah seorang pengembang software dimana informasi yang dibutuhkan harus dipahami terlebih dahulu dan kemudian membuat pengumpulan data melalui metode mengadakan observasi, diskusi, wawancara dan lainnya serta setelah semua keperluan data terpenuhi maka langkah yang selanjutnya dilakukan adalah analisis kebutuhan software yang nanti diperluas.
2. *Design System and Software*, merupakan langkah dimana sesudah semua informasi diperoleh sehingga tahap berikutnya seperti membuat perancangan desain yang tujuannya untuk bisa mempermudah dalam gambaran lengkap menyangkut hal apa yang akan diselenggarakan terlebih dahulu. Tahap ini

juga menolong pembuat saat menyiapkan perangkat keras pada proses pembuatan perangkat lunak dengan semuanya.

3. *Implementation and Unit Testing*, merupakan tahap dimana seorang programmer melakukan pembagian terkait bagian kecil serta nanti disatukan kembali bersama komponen berikutnya serta kemudian dilakukan pengujian terhadap komponen tersebut untuk memastikan apakah berfungsi dan mengikuti yang diharapkan.
4. *Integration and System Testing*, merupakan langkah melakukan integrasi pada *system* secara menyeluruh sehingga dapat melihat apakah kemungkinan kesalahan terjadi atau mungkin *system* mengalami kegagalan.
5. *Operasi dan perbaikan*, merupakan tahap akhir dimana programmer menjalankan perangkat lunak yang telah diperbaiki kesalahan.

2.1.3.2 Metode RAD (*Rapid Application Development*)

RAD (*Rapid Application Development*) termasuk suatu metode saat pengembangan aplikasi yang sangat sesuai untuk pengembangan aplikasi digital pada saat ini dengan menekankan siklus pengembangan dengan waktu yang singkat dan menggunakan pendekatan berorientasi objek dalam *system* pengembangannya. Tujuan utama metode RAD adalah untuk dapat melakukan perkembangan aplikasi dengan cepat sehingga aplikasi tersebut dapat dengan mudah teriterasi tanpa harus mengulangi dari awal kembali. Metode RAD memungkinkan pengembang perangkat lunak untuk bersifat proaktif dan merespon umpan balik sehingga pengguna dapat memastikan hasil yang diperoleh semakin berfokus untuk mutu

serta mengikuti kebutuhan pemakai (Arya et al., 2020b). Berikut ini merupakan tahapan-tahapan yang ada pada metode RAD:

1. *Bussiness Modeling*, dalam pengembangan perangkat lunak tahap ini digunakan sebagai tahapan dalam mencari aliran informasi, pengendalian informasi, mengidentifikasi, serta memilih jenis informasi apa yang nanti diperoleh.
2. *Data Modeling*, untuk tahap ini semakin banyak memberi perhatian untuk pemodelan data yang akan diperlukan saat membangun proyek dan objek yang dibutuhkan berupa atribut maupun karakteristik dari semua objek tersebut dan kemudian kaitan antar objek tersebut pula didefeneisikan.
3. *Process Modeling*, tahap ini mengikutsertakan proses pembuatan lagi aliran data yang diciptakan oleh tahap *Modelling*. Aliran ini merupakan aliran yang telah memenuhi keperluan dalam implementasi fungsi bisnis yang sudah dipilih. Pada tahap ini juga mengikutsertakan proses penambahan, mutasi dan penghapusan data khusus.
4. *Application Generation*, untuk tahap ini pengguna dapat memakai bagian oleh program sebelumnya untuk membuat komponen baru yang akan digunakan kembali. Selain itu, kerangka kerja yang digunakan untuk mempercepat proses pembuatan perangkat lunak.
5. *Testing and Turnover*, tahap ini termasuk tahapan yang paling ujung digunakan karena untuk tahap ini dilakukan pengujian terhadap bagian baru yang sudah dibuat atau diperluas secara menyeluruh untuk setiap antar muka.

Semua bagian yang ada mesti diadakan pengujian dengan semuanya sebelum dipakai dengan produktif.

2.1.3.3 Metode *Scrum*

Metode *scrum* termasuk metode pendekatan yang dipakai untuk rekayasa perangkat lunak. Metode ini juga termasuk metode pengembangan dari metode agile. Metode ini dimanfaatkan sebagai metode penyelesaian permasalahan dan dapat memberikan hasil kerja yang efektif dengan waktu seminimal mungkin dan pada akhir tujuan yang besar (Ritonga et al., 2021). Langkah yang digunakan untuk metode tersebut termasuk dengan melakukan tahapan seperti:

1. Langkah awal yang mesti dibuat termasuk dengan menentukan tim.
2. Membuat target waktu dalam penyelesaian projek.
3. Membagi peran-peran pada tim tersebut.
4. Kumpulkan berbagai permasalahan.
5. Membuat *sprint*.

2.1.3.4 Metode *Prototype*

Metode ini termasuk metode yang dipakai pada *system* kemajuan perangkat lunak yang memungkinkan program dirancang dan dibuat dengan secara bertahap dalam waktu yang cepat. Pengembangan *system* dengan menggunakan metode ini dilakukan perubahan mengikuti permintaan pemakai. Metode ini juga disebut metode teknik pengembangan *system* yang dapat menunjukkan pada *client* atau

pemilik sendiri tentang *system* yang sedang didirikan dari tim pengembang (Yoko et al., 2019). Tahapan untuk metode ini terbagi atas beberapa bagian berikut:

1. *Requirements Gathering and Analysis* (Analisis Kebutuhan), pada tahap ini tim pengembang mengumpulkan bahan-bahan yang digunakan untuk membahas *system* secara menyeluruh dengan memepertanyakan apa saja yang diinginkan oleh pemakai nantinya.
2. *Quick design*, tahap ini adalah tahap yang paling sederhana karena proses yang digunakan hanya membagikan deskripsi maupun gambaran tentang *system* yang akan didirikan.
3. Membangun *Prototype*, tahap ini merupakan tahap yang ditunjukkan kepada seorang programmer yang akan membangun sebuah *system* dan di setujui secara bersama.
4. Mengevaluasi Pengguna Awal, merupakan tahap yang melakukan presentasi *system* kepada *client* dan melakukan evaluasi terhadap komentar dan saran yang diterima.
5. *Refining Prototype* (perbaikan *Prototype*), tahap ini adalah tahap yang digunakan oleh pengembang untuk melakukan perbaikan secara berulang hingga *client* menyetujui pengembangan *system* tersebut.
6. *Implement Product and Maintain*, tahap ini termasuk tahap akhir yang digunakan oleh tim pengembang dan pada tahap ini dilakukan pengujian *system* serta diserahkan kepada *client* yang bersangkutan dengan keadaan sudah aman dalam penggunaan.

2.2 *Android*

Android merupakan *system* operasi berbasis linux yang bisa dipakai pada beragam perangkat *mobile*, selain itu fitur-fitur pada *android* juga sudah diuji dan mampu malakukan aktivitas dengan baik. *Android* bersifat *open source* makanya bisa dipakai dari siapapun dalam mengembangkan suatu aplikasi (Situmorang & Sirait, 2020). Karena *system* pada *android* selalu terus menerus mengalami perkembangan maka *system* operasi pada *android* sudah dianggap sebagai *system* yang tidak bisa dianggap sebagai biasa aja dan sekarang *system android* sudah digunakan pada perusahaan untuk meningkatkan kinerja dan menumbuhkan kedisiplinan ditempat kerja. Salah satau *system android* yang dimanfaatkan ditempat adalah *system* absensi (Dayumi & Femy Mulya, 2018). Berikut dibawah ini versi-versi android yang telah diluncurkan:

1. *Android Versi 1.0 – 1.1*, android versi ini awal mulanya diluncurkan ketika tanggal 23 sepetember 2008 dari google yang disebut dengan versi alpa beta dengan dilengkapi google talk dan youtube.



Gambar 2. 1 *Versi Alpa Beta 1.0-1.1*

Sumber: (Kasema et al., 2018)

2. *Android Versi 1.5*, android versi ini juga disebut android versi cupcake yang diluncurkan untuk 30 april 2009 dengan kelebihan mendukung *keyboard virtual*.



Gambar 2. 2 *Versi 1.5 Cupcake*
Sumber: (Kasema et al., 2018)

3. *Android Versi 1.6*, android versi ini pertama kali diluncurkan pada 15 september 2009 dengan kelebihan pada user friendly yang lebih fleksibel.



Gambar 2. 3 *Versi 1.6*
Sumber: (Pasinggil & Palelleng, 2018)

4. *Android Versi 2.0 – 2.1* , anroid versi ini awalnya diluncurkan untuk 26 oktober 2009 dengan kelebihan ditemukan untuk tampilan navigasi belokan yang sekarang disebut dnegan google maps.



Gambar 2. 4 *Versi 2.0-2.1 Eclair*
Sumber: (Pasinggil & Palelleng, 2018)

5. *Android Versi 2.2*, android versi ini awal mulanya diluncurkan untuk 10 mei 2010 dengan nama lain froyo dan memiliki kelebihan pada *hotspot*, *wifi* dan *fitur USB thetring*.



Gambar 2. 5 *Versi 2.0 Froyo*
Sumber: (Pasinggil & Palelleng, 2018)

6. *Android Versi 2.3*, android ini diluncurkan ketika 6 desember 2010 dengan nama lain Gingerbread dan memiliki kelebihan pada tampilan yang lebih sederhana.



Gambar 2. 6 *Versi 2.3 Gingerbread*
Sumber: (Pasinggil & Palelleng, 2018)

7. *Android Versi 3.0-3.2*, versi ini di luncurkan pada 22 february 2011 dengan kelebihan pada tampilan multi-core.



Gambar 2. 7 *Versi 3.0-3.2 Honeycomb*
Sumber: (Pasinggil & Palelleng, 2018)

8. *Android Versi 4.0*, android versi ini pertama kali diluncurkan pada 19 oktober 2011 dengan kelebihan mendukung user interface pada tablet dan memudahkan dalam pengiriman data karena adanya NFC.



Gambar 2. 8 *Versi 4.0*

Sumber: (Pasinggil & Palelleng, 2018)

9. *Android Versi 4.1*, android versi ini pertama kali diluncurkan pada 9 juni 2012 dengan nama lain jellybean dan memiliki kelbihan pada batrai yang lebih awet.



Gambar 2. 9 *Versi 4.1*

Sumber: (Pasinggil & Palelleng, 2018)

10. *Android Versi 4.4*, android versi ini diluncurkan pada 31 oktober 2013 dengan nama lain kitkat dan memiliki kelebihan pad model *interface update*.



Gambar 2. 10 *Versi 4.4 Kitkat*

Sumber: (Pasinggil & Palelleng, 2018)

11. *Android Versi 5.0*, android versi ini awalnya diluncurkan ketikaa 17 oktober 2014 serta disebut dengan Lollipop dengan kelebihan pada *dual sim card*.



Gambar 2. 11 *Versi 5.0 lolipop*
Sumber: (Pasinggil & Palelleng, 2018)

12. *Android Versi 6.0*, android versi ini awal mulanya diluncurkan pada 28 mei 2015 dengan kelebihan pada *fingerpint* dan disebut dengan versi *Marshmallow*.



Gambar 2. 12 *Versi 6.0 Marshallow*
Sumber: (Pasinggil & Palelleng, 2018)

13. *Android Versi 7.0-7.1*, android versi ini pertama kali diluncurkan pada 22 agustus 2016 dengan nama lain Nougat dan memiliki kelebihan pada keahlian agar membuka lebih dari sebuah windows dengan bersamaan.



Gambar 2. 13 *Versi 7.0-7.1 Nougat*
Sumber: (Pasinggil & Palelleng, 2018)

14. *Android Versi 8.0-8.1*, android versi ini pertama kali diluncurkan pada 21 agustus 2017 dengan nama lain Oreo dan memiliki kelebihan pada batrai yang bertahan lama dan kecepatan yang semakin baik.



Gambar 2. 14 *Versi .0 Oreo*
Sumber: (Pasinggil & Palelleng, 2018)

15. *Android Versi 9*, android versi ini pertama kali diluncurkan pada 6 agustus 2016 dengan kelebihan pada pola daya kegiatan aplikasi.



Gambar 2. 15 *Versi 9*
Sumber: (Pasinggil & Palelleng, 2018)

16. *Android Versi 10*, android versi ini awalnya diluncurkan untuk 3 september 2019 melalui kelebihan pada fitur otomatis menampilkan teks yang muncul untuk video.



Gambar 2. 16 *Versi 10*
Sumber: (Pasinggil & Palelleng, 2018)

17. *Android Versi 11*, android versi ini pertama kali diluncurkan pada 8 september 2020 dengan nama lain *Red Velved* dan memiliki kelebihan pada akses aplikasi lain.



Gambar 2. 17 *Versi 11 Red Velvet*
Sumber: (Pasinggil & Palelleng, 2018)

18. *Android Versi 12*, android versi ini pertama kali diluncurkan pada 4 oktober 2021 dengan nama lain *Snowcone* dengan kelebihan pada fitur kepintaran buatan yang membuat tema warna pada menu sistme aplikasi.



Gambar 2. 18 *Versi 12 Snow Cone*
Sumber: (Pasinggil & Palelleng, 2018)

19. *Android Versi 13*, Android versi ini pertama kali di luncurkan untuk 10 februari 2022 melalui kemampuan pada peningkatan privasi, keamanan serta otimasi UI.



Gambar 2. 19 *Versi 13*
Sumber: (Pasinggil & Palelleng, 2018)

2.3 Teori Khusus

2.3.1 Absensi

Pencatatan absensi merujuk pada proses dokumentasi kehadiran seseorang atau pegawai, yang merupakan bagian integral dari pelaporan dalam suatu institusi. Data kehadiran disusun dengan rapi dan mudah dicari, dan dapat digunakan saat dibutuhkan oleh pihak terkait (Mir et al., 2018).

Menurut (Arya et al., 2020b) ditemukan beragam jenis absensi yang dipisahkan dari metode pemakaian serta tingkat kegunaannya. Biasanya, absensi dapat dipisahkan atas dua kategori:

1. Absensi secara manual, yang melibatkan pengentrian kehadiran maupun kedatangan memakai pena, tanda tangan, serta kertas.
2. Absensi dengan non-manual, yang melibatkan pengentrian kehadiran maupun kedatangan memakai alat atau sistem terkomputerisasi, misalnya kartu pada kode batang (barcode) atau identifikasi sidik jari (*fingerprint*).

Menurut (Henny Leidiyana & Yusuf, 2021) Absensi atau jam kehadiran karyawan adalah suatu bentuk pencatatan waktu kerja pada setiap kehadiran karyawan di suatu perusahaan atau tempat kerja lainnya. Pencatatan kehadiran karyawan ini bisa berupa kehadiran biasa yang tertulis atau dalam bentuk pencatatan mesin waktu. Catatan waktu kehadiran biasanya terbagi atas dua komponen seperti:

1. Pencatatan waktu hadir (*attendance time keeping*)

Jika seseorang pegawai atau pegawai tidak ingat maupun tidak mencatat waktu datangnya untuk waktu jam hadir sehingga nanti berdampak pada bagian gaji bersih tiap bulan yang akan diterima oleh pegawai tersebut terutama yang berkaitan dengan tunjangan baik tunjangan transportasi, makan maupun lainnya. Catatan waktu kehadiran dikumpulkan dan dihitung berdasarkan data jumlah kehadiran karyawan tersebut dan dibayarkan pada periode atau waktu tertentu.

2. Pencatatan waktu kerja

Pencatatan waktu kerja dimaksud adalah untuk dapat melakukan pencatatan terhadap waktu yang digunakan dalam bekerja yang dimanfaatkan untuk diberlakukannya pengecekan terhadap waktu kehadiran kerja dan menghasilkan data produksi yang dibutuhkan sesuai dengan perhitungan gaji.

2.3.2 Location Based Service (LBS)

Location Based Services (LBS) tau layanan berbasis lokasi merujuk pada layanan data yang bisa diakses melalui perangkat berjalan lewat jaringan, memungkinkan tampilan geografis posisi perangkatnya. Dua elemen utama oleh *Location Based Services* meliputi:

1. *Location Manager (API Maps)*: Memberikan perangkat untuk sumber maupun source untuk LBS, *Application Programming Interface (API Maps)* memfasilitasi tampilan peta.
2. *Location Providers (API Location)*: Menyiapkan teknologi penemuan lokasi yang dipakai dari perangkat. *API Location* berkaitan pada data GPS (*Global Positioning System*) serta data lokasi *real-time*.



Gambar 2. 20 *Location Based Service (LBS)*
Sumber: (Arya et al., 2020b)

2.3.3 *Global Positioning System (GPS)*

GPS termasuk istilah dari NAVSTRAR GPS (*Navigation Satelit Timming and Ranging Global Positioning System*) yang terdiri dari tiga bagian penting, seperti satelit, pengendali, serta pemakai. GPS termasuk sebuah sistem navigasi serta pemilihan posisi melalui satelit. Sistem ini dirancang agar menyediakan informasi posisi serta kecepatan tiga dimensi, serta data waktu dengan terus-menerus pada semua dunia, tanpa terpengaruh oleh waktu serta cuaca, untuk banyak pengguna dengan bersamaan. GPS mampu membagikan data posisi pada tingkat ketelitian bervariasi, diawali oleh beberapa milimeter hingga puluhan meter



Gambar 2. 21 *Global Positioning System (GPS)*
Sumber: (Henny Leidiyana & Yusuf, 2021)

2.3.4 *Google Maps*

Google Maps termasuk layanan peta gratis dari Google yang memungkinkan pengguna mengakses peta dunia untuk melihat suatu daerah. *Google Maps API* termasuk suatu perpustakaan berbasis JavaScript yang disiapkan dari Google agar mendirikan aplikasi peta digital yang handal, memungkinkan fokus untuk data yang ingin ditampilkan. Melalui layanan peta, *Google Maps* menampilkan gambar lokasi, informasi lalu lintas, dan bahkan memberikan visualisasi bentuk atau tampilan bumi. Akses ke *Google Maps* dapat dilakukan melalui berbagai perangkat, termasuk komputer, tablet, atau ponsel selama terhubung dengan internet atau jaringan.



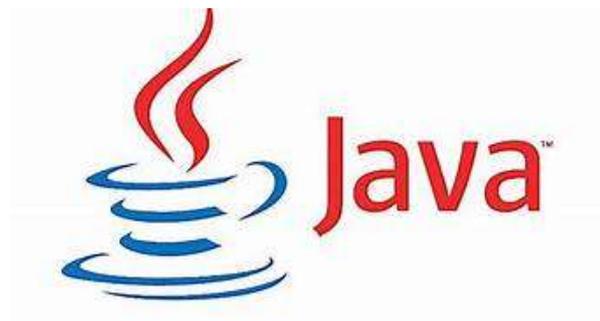
Gambar 2. 22 *Google Maps*
Sumber: (Dayumi & Femy Mulya, 2018)

2.4 *Software Pendukung*

Termasuk sebuah perangkat lunak atau aplikasi yang dipakai dari pengguna agar menjadi perangkat tambahan yang mendukung proses penyelesaian penelitian atau proses pembuatan aplikasi. Berikut ini adalah perangkat lunak yang dibutuhkan:

2.4.1 Bahasa Pemrograman java

Java termasuk bahasa pemrograman berorientasi objek yang bisa beroperasi pada beragam jenis komputer, termasuk telepon genggam. Bahasa pemrograman ini pertama kali dikembangkan dari Sun Microsystems, yang sekarang menjadi komponen oleh Oracle, dan diperkenalkan untuk tahun 1995.



Gambar 2. 23 Bahasa Pemrograman Java
Sumber: (Henny Leidiyana & Yusuf, 2021)

2.4.2 JSON Data (*JavaScript Object Notation*)

JSON Data termasuk format pertukaran data yang ringan, gampang dimengerti, dibaca, serta ditulis dari manusia. Format ini juga gampang dipahami serta diurai dari mesin. JSON Data diandasi untuk subset Bahasa Pemrograman JavaScript, yaitu Standar ECMA-262 3rd Edition - Desember 1999. Meskipun termasuk format teks yang sepenuhnya independen, JSON memakai konvensi yang akrab bagi semua pengembang bahasa C, misalnya C, C++, C#, Java, JavaScript, Perl, Python, serta lain-lain (Dayumi & Femy Mulya, 2018).

2.5 *Start* (UML)

Unified Modeling Language (UML) termasuk sebuah perangkat lunak standar *modelling* yang sering dipakai oleh seorang programmer agar membuat

pemodelan gambar desain berorientasi objek tentang perancangan yang akan dibuat (Fitriana et al., 2020). Berikut di bawah ini merupakan tujuan dari penggunaan UML bagi para perancang aplikasi:

1. Pemodelan yang akan dibuat dalam bentuk berorientasi objek dan digunakan untuk pemodelan visual.
2. UML merupakan perangkat lunak yang memperoleh bahasa yang gampang untuk dimengerti manusia pada saat membuat pemodelan.



Gambar 2. 24 *Star UML*

Sumber: (Henny Leidiyana & Yusuf, 2021)

Seperti pemodelan yang biasa dipakai seperti:

1. *Usecase Diagram*

Usecase diagram termasuk sebuah diagram yang membantu dalam memberikan gambaran atau deskripsi tentang actor yang berperan pada suatu system makanya pemakai semakin memahami langkah kerja yang nanti dibuat dari system itu serta membagikan perintah untuk system melalui beragam macam pertanyaan. Berikut symbol-simbol yang ditemukan untuk *usecase diagram*:

Tabel 2. 1 Usecase Diagram

Simbol	Keterangan
	Simbol ini dipakai agar mengaitkan antara usecase yang satu bersama yang lainnya.
	Merupakan pengguna yang memiliki peran pada system yang sedang berjalan atau manusia itu pribadi
Asosiasi / association 	Merupakan koneksi antara usecase serta actor yang bekerja secara langsung.
Ekstensi/extend <<extend>> 	Memiliki arti sebagai kondisi tertentu yang sedang berjalan pada sistem
Generalisasi / generalization 	Kondisi dimana dua kasus yang saling terhubung baik secara khusus maupun umum.
Menggunakan/include/uses <<include>> 	Merupakan koneksi yang berhubungan dengan pengoperasian usecase dalam system penambahan.

Sumber: (Henny Leidiyana & Yusuf, 2021)

2. Activity Diagram

Activity diagram termasuk suatu kegiatan maupun kegiatan yang dibuat dari sebuah system yang menggambarkan aliran kerja pada system tersebut (Henny Leidiyana & Yusuf, 2021). Berikut symbol-simbol yang ditemukan untuk activity diagram:

Tabel 2. 2 Activity Diagram

Simbol	Keterangan
Start 	Suatu kondisi yang dilakukan diawal aktifitas
Aktivitas 	Suatu aktivitas yang dimulai melalui kata kerja
Percabangan / decision 	Merupakan percabangan memiliki lebih oleh dua kegiatan
Pengabungan / join 	Kegiatan yang digabungkan sebab memperoleh lebih dari dua aktivitas.
Finish 	Situasi selesaia dari tahap eksekusi system pada aktivitas diagram.
Swimle  Atau	Merupakan tanggung jawab dari suatu system terhadap aktivitas yang terjadi.

Sumber: (Henny Leidiyana & Yusuf, 2021)

3. Class Diagram

Class Diagram termasuk suatu kegiatan berkaitan dengan kelas-kelas untuk system tersebut serta memiliki aturan tertentu untuk masing-masing tanggung jawab pelaku system. Berikut tampilan symbol-simbol yang ditemukan untuk *class diagram*:

Tabel 2. 3 *Class Diagram*

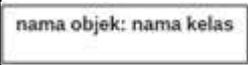
Simbol		Keterangan
<i>Class</i>	Nama_kelas	termasuk class yang ada untuk struktur sistem
	+atribut	
	+operasi()	
<i>Interface</i> 		Tampilan tatap muka pada sebuah objek yang ada di sistem.
<i>association</i> 		Hubungan antar kelas yang satu bersama kelas yang lain yang saling memiliki arti dalam banyak hal.
<i>directed association</i> 		Merupakan penerapan hubungan antar kelas dengan kategori hubungan lainnya.
Generalisasi 		Keterkaitan antara kelas umum ke kelas khusus
<i>depedency</i> 		Memiliki makna ketergantungan antar kelas yang sama.
<i>agregation</i> 		Hubungan kelas dengan makna sebagian (<i>whole part</i>)

Sumber: (Henny Leidiyana & Yusuf, 2021)

4. *Sequence Diagram*

Merupakan suatu diagram yang menggambarkan wujud oleh suatu tindakan system untuk usecase yang dipaparkan dari waktu hidup entitas dalam membagikan informasi. Seperti symbol-simbol yang ditemukan untuk *usecase diagram*:

Tabel 2. 4 *Squence Diagram*

Simbol	Keterangan
<p><i>Actor</i></p> 	Merupakan orang atau pengguna atau bahkan benda yang melakukan kerja pada sistem
<p><i>life line</i></p> 	Merupakan garis kehidupan yang dipaparkan pada sebuah objek tertentu.
<p>Objek</p> 	Merupakan interaksi pada system berwujud pesan.
<p>Waktu aktif</p> 	Merupakan keadaan aktif yang dinyatakan suatu objek dalam interaksi system.
<p>Pesan tipe create</p> <p><<create>></p> 	termasuk tampilan pemaparan sebuah objek yang membuat objek lain.

Sumber: (Henny Leidiyana & Yusuf, 2021)

2.6 Penelitian Terdahulu

Termasuk penelitian yang pernah atau dibuat dari peneliti lain nanti sebagai referensi untuk penelitian ini.

- 1 Oleh jurnal penelitian yang sudah dibuat dari (Dayumi & Femy Mulya, 2018) melalui judul “Sistem Absensi karyawan Berbassis Location Based Service (LBS) Menggunakan Platform*Android*“ Jurnal sistem komputer serta kepintaran buatan Vol 2 no 1 dengan hasil penelitian aplikasi sistem absensi

ini dapat digunakan sebagai media untuk melakukan aktifitas absensi dan dapat juga digunakan untuk melihat report (laporan) kinerja karyawan.

- 2 Dari jurnal penelitian yang sudah dibuat dari (Khoir et al., 2020b) melalui judul “Implementasi GPS (*Global Positioning System*) Pada Presensi Berbasis *Android* di BMT Insan Mandiri” Jurnal sains komputer dan Informatika vol4 no 1 ISSN 2548-9771 dengan hasil penelitian sistem aplikasi *android* dapat memperlancar proses presensi kedatangan juga kepeulangan marketing atau tenaga pemasaran dan dapat mengurangi kecurangan atau manipulasi persensi serta mampu mempermudah perusahaan dalam melakukan pengawasan teradap marketing.
- 3 Dari jurnal penelitian yang sudah dibuat dari (Dan et al., 2020) dengan judul ”Rapid Application Development Sistem Pembelajaran Daring Berbasis *Android*” Jurnal Intech vol 1 no 2 ISSN 2722-7367 dengan hasil penelitian dengan penggunaan RAD dapat diterima dengan baik oleh pengguna dengan presentase sebesar 91% yang telah diuji dengan menggunakan *User Acceptance Testing* (UAT).
- 4 Dari jurnal penelitian yang sudah dibuat dari (Arya et al., 2020b) dengan judul “Sistem Informasi Absensi Pegawai Menggunakan Metode RAD dan LBS Pada Koordinat Absensi” Dalam Jurnal Media Informatika Budidarma Volume 4 Nomor 1 ber-ISSN 2614-5278, terdapat hasil penelitian yang menyebutkan bahwa bagian pegawai administrasi dapat dengan mudah memantau dan merekap data presensi pegawai sesuai kebutuhan. Selain itu, pegawai yang sedang mengadakan dinas luar kantor kini dapat membuat

absensi tanpa mesti datang ke kantor sebelumnya, sehingga beban kerja yang diperoleh dari pegawai dapat sedikit berkurang.

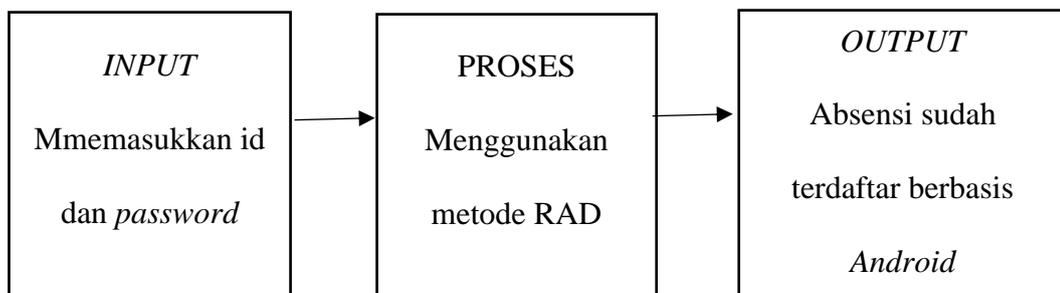
- 5 Dari jurnal penelitian yang sudah dibuat dari (Irawan et al., 2020) dengan judul “Penerapan Location based Service Untuk Pencarian Lokasi Rapat Menggunakan Metode Design Sprint” Dalam Jurnal Mahasiswa Teknik Informatika Volume 4 Nomor 2, hasil pengujian pada beberapa perangkat menunjukkan bahwa semua fitur yang telah dirancang dapat beroperasi dengan baik, kecuali untuk fitur arah lokasi. Keterbatasan ini disebabkan oleh kebutuhan pengguna untuk memiliki aplikasi Google Maps bawaan agar dapat menggunakan fitur tersebut.
- 6 Dari jurnal penelitian yang sudah dibuat dari (Informasi et al., 2023) dengan judul “Perancangan Sistem Absensi Berbasis Location Based Service (LBS) Pada PT. Hascar Interasional Motor” Jurnal Informasi dan Teknologi Vol 5 no 1 ISSN 2714-9730 dengan hasil penelitian bahwa dapat memudahkan karyawan dalam melakukan absensi, tidak perlu menulis jam masuk dan keluar secara manual tetapi dapat dilakukan dengan melakukan absensi melalui aplikasi absensi karyawan berbasis *android*.
- 7 Dari jurnal penelitian yang sudah dibuat dari (Henny Leidiyana & Yusuf, 2021) dengan judul “Aplikasi Kehadiran Karyawan Berbasis *Android* Menggunakan QR Code Scanning dan Location Based Service” Jurnal of information and information security (JIFORITY) vol 2 no 1 ISSN 2722-4058 dengan hasil penelitian dengan memanfaatkan aplikasi ini memungkinkan proses presensi membuat semakin efektif serta efisien daripada sistem

terdahulu. Aplikasi ini dilengkapi dengan fitur keamanan yang memungkinkan hanya satu perangkat pada seorang pegawai, dari nomor IMEI handphone. Sistem ini sudah diuji dalam pemakaian, serta hasilnya menunjukkan peningkatan efisiensi sebab bersifat tanpa kertas.

- 8 Dari jurnal penelitian yang sudah dibuat dari (Info & Service, 2023) dengan judul “Penggunaan Smartphone Berbasis *Android* Dalam Penerapan Location Based Service Pada Absensi Karyawan dengan Metode OOAD” vol 16 no 1 ISSN 2502-339X memberi kesimpulan bahwa sistem aplikasi absensi karyawan berbasis *android* dengan menerapkan LBS bisa otomatis mengkonversi koordinat kehadiran pegawai menjadikan suatu alamat makanya personalia menemukan area keberadaan pegawai ketika membuat absensi.

2.7 Kerangka Pemikiran

Merupakan pemikiran dari peneliti yang dimulai dari awal, proses hingga hasil *output* yang dituangkan dalam bentuk kerangka. Dalam perancangan aplikasi absensi karyawan menggunakan *location based based* (LBS) dilakukan dengan langkah berikut ini:



Gambar 2. 25 Kerangka Pemikiran