

BAB II

KAJIAN PUSTAKA

2.1 Teori Dasar

Dalam penelitian ini dibutuhkan beberapa teori dasar yang akan mendukung dalam tahap pembuatan aplikasi seperti *Use Case*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram* yang meliputi yaitu *Android*, *UML*, dan *black box testing*.

2.1.1 Android

Android merupakan perangkat keras yang sangat banyak di gunakan oleh kalangan masyarakat saat ini, bisa di kategorikan kurang lebih dari tiga miliar pengguna aktif *android* di seluruh dunia saat ini, hal itu langsung di sampaikan oleh Vice Presiden produk pengelola google yaitu Sameer Samat dalam konferensi *developer google I/O 2021*, sistem operasi ini sudah tersedia di beberapa perangkat antara lain *smartphone* hingga *wearable*.

Biaya yang rendah yang di miliki *android* memiliki sifat system operasi agar teknologi yang tinggi dapat di Kembangan dengan mudah, karena sifat *android* yang *open source* sehingga muncul komunitas pembuatan aplikasi menggunakan kode sumber terbuka (Ramayani, 2021). *Android* artinya sebuah sistem operasi perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* serta *software*. *Android* menyediakan *platform* terbuka bagi para pengembang buat membangun perangkat lunak mereka agar bisa di gunakan di piranti bergerak. *Android* biasa di gunakan di *smartphone* dan juga tablet PC (Tjioetama et al., 2018).



Gambar 2.1 Logo *Android*

Sumber : (Data Penelitian, 2021)

2.1.2 Unified Modeling Language (UML)

Membuat suatu aplikasi tidak akan cukup hanya dengan menuangkan suatu ide saja, namun dalam pembuatan aplikasi sangat dibutuhkan desain terlebih dahulu dengan menggunakan alat *modeling* tertentu.

Unified Modeling Language (UML) adalah sesuatu *software* untuk memvisualisasikan dan mendokumentasikan hasil Analisa dan desain yang berisikan sintak untuk memodelkan system secara visual (Haerulah & Ismiyati, 2017).

UML merupakan program berorientasi objek yang memiliki kemampuan dalam menganalisa dan menjelaskan system secara detail, selain itu UML juga mempunyai kemampuan yang baik untuk mempersentasikan rancangan dari system (Herlina, 2021)

Pada umumnya ada beberapa konsep pendekatan UML dengan menggunakan *Use case Diagram*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram*.

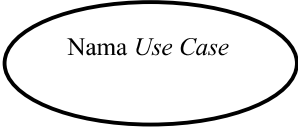
1. *Use Case Diagram*

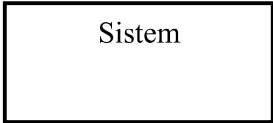

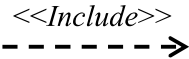
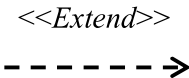
Use Case Diagram berguna untuk menggambarkan apa saja fungsi dari aplikasi yang dimana nanti nya akan memudahkan kita mengingat fitur-fitur yang ada dalam aplikasi tersebut.



Diagram Use Case terdiri dari:

- a. *Use Case*
- b. *Actor*
- c. *Relationship*
- d. *System boundari* / batas sitem (opsional)
- e. *Include*
- f. *Extend*

Tabel 2.1 Simbol-simbol *Use Case Diagram*

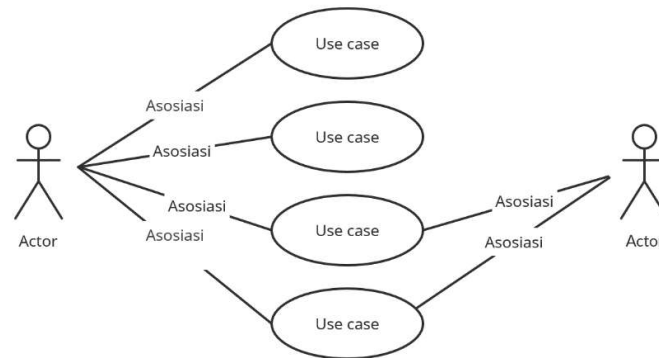
Simbol	Keterangan
	<p><i>Use Case</i></p> <ul style="list-style-type: none"> • Fungsional yang disediakan sistem untuk saling bertukar pesan antara aktor. • Batas sistem (system boundary) ditempatkan didalam. • Pemberian nama didalam use case biasanya label atau kata kerja diikuti dengan kata benda.

	<p>System boundary</p> <ul style="list-style-type: none"> • Bagian atas dari sistem <i>boundary</i> terdapat pada nama sistem. • Menggambarkan tentang ruang lingkup sistem
	<p>Association</p> <ul style="list-style-type: none"> • Abstraksi penghubung aktor dengan use case yang berinteraksi.
	<p>Include</p> <ul style="list-style-type: none"> • Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang telah di tambahkan memerlukan <i>Use Case</i> ini untuk menjalankan fungsinya sebagai syarat di jalankan <i>use case</i> ini. • <i>Include</i> berarti <i>Use Case</i> yang di tambahkan akan selalu di panggil saat <i>Use Case</i> tambahan di jalankan. • <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan sudah di jalankan.
	<p>Extend</p> <ul style="list-style-type: none"> • <i>Case</i> tambahan memiliki nama depan yang sama dengan <i>Use Case</i> yang di tambahkan.

	<ul style="list-style-type: none"> • Arah panah mengarah pada <i>use case</i> yang telah di tambahkan.
	<p>Generalisasi / <i>Generalization</i></p> <ul style="list-style-type: none"> • Menghubungkan <i>use case</i> yang umum dengan yang khusus
	<p>Aktor</p> <ul style="list-style-type: none"> • Seseorang yang berinteraksi dengan sistem informasi. • Walaupun simbol dari aktor orang, akan tetapi aktor belum tentu sebagai gambar orang, akan tetapi biasanya digunakan dengan kata benda dari awal frase nama aktor.

Sumber: (Maharani, 2018)

Dibawah ini adalah gambar alur dari sebuah diagram Use case dalam contoh kasus, sebagai berikut:



Gambar 2.2 Contoh *Use Case Diagram*



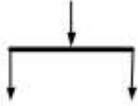
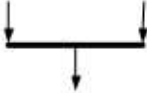
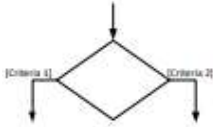
Sumber : (Data Penelitian, 2021)


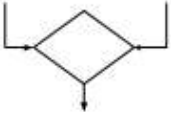
2. *Activity Diagram*

Sesuai dengan nama nya aktifitas diagram ini akan menggambarkan tentang bagaimana proses aktifitas yang terjadi pada sistem dari awal hingga akhir. Dalam aktifitas diagram ini akan di gambarkan bagaimana cara kerja atau aktiiftas kerja dari sistem tersebut.

Tabel 2.2 Simbol-simbol *activity Diagram*

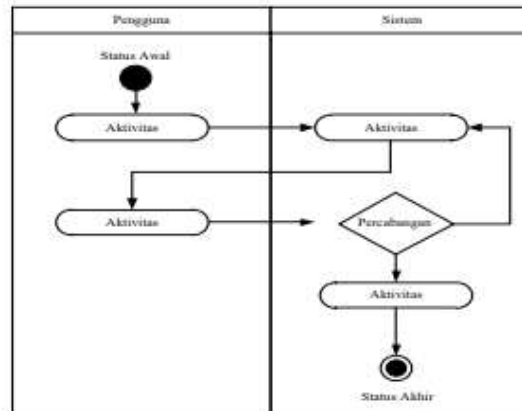
Simbol	Keterangan
●	<p><i>Start point</i></p> <ul style="list-style-type: none"> • Awal mula penelusuran. • <i>Start point</i> yang selalu akan memulai sebuah aktivitas. • Hanya boleh digunakan 1 simbol <i>start poin</i> pada sebuah aktivitas diagram.

	<p><i>End Point</i></p> <ul style="list-style-type: none"> • Akhir dari penelusuran. • <i>End Point</i> akhir dari sebuah aktifitas diagram. • >1 simbol <i>End Point</i> yang boleh digunakan pada sebuah aktifitas.
	<p><i>Actifity</i></p> <ul style="list-style-type: none"> • Akhir penelusuran. • Istila kerja yang bisa di isi buat mendeskripsikan sebuah aktivitas • satu alur masuk dan satu alur keluar yang dimiliki sebuah kegiatan.
	<p><i>Fork</i></p> <ul style="list-style-type: none"> • Percabangan. • Mengerjakan secara bersamaan dua atau lebih yang ditimbulkan satu sirkulasi.
	<p><i>Joint</i></p> <ul style="list-style-type: none"> • Penggabungan kegiatan melanjutkan disatukan asal beberapa sirkulasi.
	<p><i>Decision point</i></p> <ul style="list-style-type: none"> • Pada tengah belah ketupat tidak terdapat identifikasi (pertanyaan) <i>guards</i> wajib di miliki oleh <i>flowchart</i>.

	<p><i>Guard</i></p> <ul style="list-style-type: none"> • Sebuah transisi waktu dilewati sebuah syarat benar.
	<p><i>Merge</i></p> <ul style="list-style-type: none"> • Setelah melewati <i>decision point</i> dan berbeda ketika membawa kembali jalur keputusan
<p>Nama <i>swimlane</i></p>	<p><i>Swimlane</i></p> <ul style="list-style-type: none"> • Aktor yang berdasarkan serta dikelompokan sebuah cara kegiatan. • <i>Actor</i> ditulis nama <i>actor</i>. • <i>Swimlane</i> kadang-kadang digambarkan secara horizontal dan vertikal.

Sumber: (Maharani, 2018)

Dibawah ini adalah gambar contoh kasus penggunaan dari *Activity Diagram*:





Gambar 2.3 Contoh *Activity Diagram*





Sumber : (Data Penelitian, 2021)

3. *Sequence Diagram*

Sequence Diagram mendeskripsikan sistem kerja objek pada *use case* dengan mendeskripsikan ketika hidup objek dan pesan yang dikirim dan diterima antar objek. Maka dari itu untuk mendeskripsikan diagram sequen maka wajib diketahui terlebih dahulu objek-objek yang ikut serta pada sebuah *use case* beserta metode-metode nya.

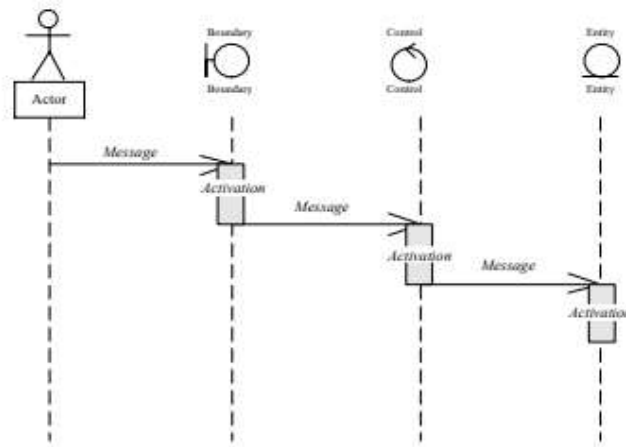
Tabel 2.3 Simbol-simbol *Sequence Diagram*

Simbol	keterangan
	<i>Actor</i> <ul style="list-style-type: none"> Orang yang akan berinteraksi dengan sistem.
	<i>Boundary</i> <ul style="list-style-type: none"> Salahsatu penghubung sistem dengan aktor..

	<p><i>Control</i></p> <ul style="list-style-type: none"> • Proses kerja sistem yang akan dikontrol dan akan mengatur jalannya suatu sistem.
	<p><i>Entity</i></p> <ul style="list-style-type: none"> • Sistem yang akan menyimpan informasi. • Data yang digambarkan oleh suatu sistem struktur data.
	<p><i>Activation</i></p> <ul style="list-style-type: none"> • Salah satu objek yang menggambarkan situasi hubungan dari suatu operasi oleh panjang simbol akan berbanding lurus dengan durasi aktif.
	<p><i>Message</i></p> <ul style="list-style-type: none"> • Uraian kegiatan yang dijelaskan pesan antar objek.

Sumber: (Maharani, 2018)

Dibawah ini adalah gambar penggunaan *Sequence Diagram* dalam contoh studi kasus sebagai berikut:



Gambar 2.4 Contoh *Sequence Diagram*

Sumber : (Data Penelitian, 2021)

4. *Class Diagram*

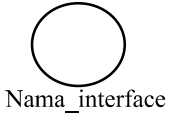



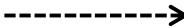

Class Diagram atau diagram kelas menggambarkan wacana struktural sistem serta pendefeniisian kelas-kelas yang akan di buat. Kelas diagram terdiri dari beberapa area yaitu:

- a. Nama
- b. Atribut (variabel yang di miliki 1 kelas).
- c. Metode (fungsi yang dimiliki suatu kelas).

Berikut simbol-simbol *Class Diagram* pada tabel di bawah ini.

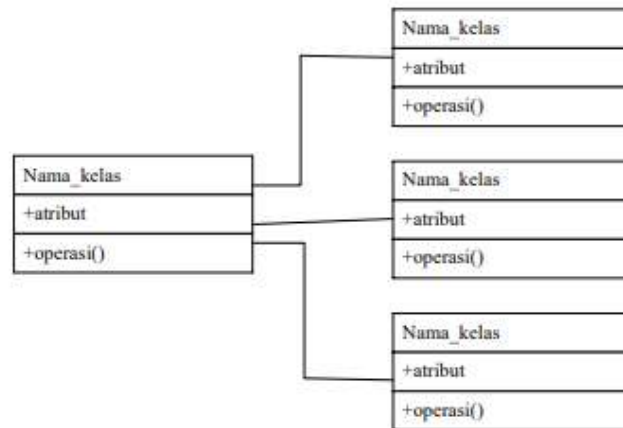
Tabel 2.4 Simbol-simbol *Class Diagram*

Simbol	Keterangan
<div style="border: 1px solid black; padding: 2px; width: fit-content;"> Nama kelas +atribut +operasi() </div>	<ul style="list-style-type: none"> • Susunan suatu sistem yang ada dalam kelas.

<p>Antarmuka / <i>interface</i></p>  <p>Nama_interface</p>	<ul style="list-style-type: none"> • Pemrograman berorientasi objek pada <i>interface</i>.
<p>Asosiasi / <i>Association</i></p> 	<ul style="list-style-type: none"> • Makna umum relasi antar kelas, asosiasi ini juga disertai dengan <i>multiplicity</i>.
<p>Asosiasi berarah</p> 	<ul style="list-style-type: none"> • Relasi antar kelas yang dimana makna kelas yang satu digunakan oleh kelas yang lain, asosiasi berarah ini juga biasanya disertai dengan <i>multiplicity</i>.
<p>Generalisasi</p> 	<ul style="list-style-type: none"> • Relasi antar kelas generalisasi-spesialisasi (umum-khusus)
<p>Kebergantungan</p> 	<ul style="list-style-type: none"> • Antar kelas kebergantungan pada makna dengan relasi antar kelas.
<p>Agregasi / <i>Aggregation</i></p> 	<ul style="list-style-type: none"> • Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>).

Sumber: (Data Penelitian, 2021)

Berikut adalah contoh penggunaan dari *Class Diagram* dalam contoh kasus sebagai berikut:



Gambar 2.5 Contoh *Class Diagram*

Sumber : (Data Penelitian, 2021)

2.1.3 Pengujian Aplikasi

Pengujian aplikasi adalah suatu rencana yang sistematis di lakukan untuk mendapat atau mengevaluasi kebenaran dari suatu perangkat lunak. Dalam penelitian ini, peneliti melakukan pengujian dengan menggunakan *Black Box Testing*.

Black Box Testing adalah alat penguji perangkat lunak yang dari segi spesifikasi *functional* tanpa penguji desain dan kode program untuk mengetahui fungsi dari masukan dan keluaran perangkat tersebut.

Maetode *Black Box Testing* adalah suatu metode yang cukup efesien di gunakan karena hanya memerlukan batas bawah dan atas dari data yang di harapkan, entimasi banyak nya data yang di hitung melalui banyak nya *field* data entri yang akan di uji, atau data entri yang harus di penuhi serta kasus batas atas dan bawah yang memenuhi (Cholifah et al., 2018).

Menurut (Mustaqbal et al., 2015) *Black Box Testing* bisa juga diartikan sebagai tester pengujian pada spesifikasi yang fungsional yang mengarah untuk menjumpai hal-hal seperti berikut ini:

1. Fungsi yang tidak benar atau salah dan tidak ada.
2. Kesalahan pada antar muka (*interface errors*).
3. Kesalahan pada susunan dan akses basis data.
4. Kesalahan dari kinerja (*performance errors*).
5. Kesalahan inisialisasi dan pengalihan.

(Mustaqbal et al., 2015) menyimpulkan bahwa *Black Box Testing* adalah alat pengujian perangkat lunak yang memiliki sifat sumber terbuka (*open source*) dan memiliki spesifikasi yang efisien dan mudah digunakan.

2.2 Teori Khusus

2.2.1 QR Code

QR Code (Quick Response Code) merupakan gambar kode dua dimensi yang bisa menyimpan data. *QR Code* bisa digunakan menyimpan berbagai macam data seperti teks, seperti kode biner, numerik, dan alfanumerik.

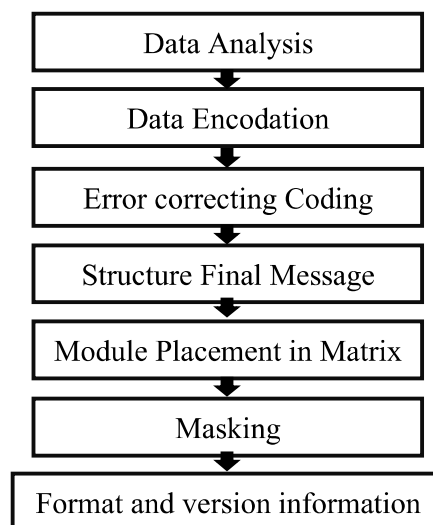
QR Code adalah gambar dua dimensi yang menampilkan suatu data, terutama data berbentuk teks. *QR Code* merupakan pengembangan dari *Barcode* yang dimana dari satu dimensi menjadi dua dimensi (Nugraha, 2011).



Gambar 2.6 Contoh *QR Code*

Sumber : (Data Penelitian, 2021)

QR Code adalah suatu gambar kode batang yang yang cara kerjanya dapat dilakukan dengan menggunakan pembaca kode (*Scanner*) oleh karena itu sangat berguna jika di terapkan dalam proses kerja dalam suatu perusahaan. *QR Code* dikembangkan oleh salah satu perusahaan Jepang yang bergerak di bidang otomotif. *QRCode* ini di publikasikan pada tahun 1994 dengan tujuan untuk melacak kendaraan di bagian manufaktur dengan cepat dan mendapatkan respon yang cepat (Nugraha, 2011). Langkah-langkah pembangkitan *QR Code* dari sebuah teks dapat dilihat dari diagram alir berikut ini:



Gambar 2.7 Contoh Diagram alir *QR Code*

Sumber : (Pasca, Rinaldi, 2011)

2.2.2 Software Pendukung

1. *Android Studio*

Android studio ialah suatu perangkat lunak yang dikembangkan oleh *Google*. *Android studio* adalah lingkungan pengembangan perangkat lunak terpadu yang merupakan IDE pemrograman *android* yang dimana sebelumnya adalah *Eclipse*. Dimana *Google* mengahiri pengembangannya dari *Eclipse* dan fokus hanya pengembangan *Android studio* saja karena *Android Studio* memiliki fitur-fitur yang cukup mudah sehingga lebih sangat mendukung pembuat program level dasar hingga ahli.



Gambar 2.8 Logo *Android Studio*

Sumber : (Data Penelitian, 2021)

Integrasi resmi *Android Studio* untuk sistem operasi *Android* pengembangan *Android* tertanam Perangkat lunak Jet Brains'Inttellij IDE (Pengembangan Environment Terpadu) sebagai platform untuk aplikasi *Android*. *Android Studio* yang dirilis oleh *Google* termasuk dalam lisensi pengembangan berkelanjutan *Apache 2.0* memungkinkan *Android Studio* berjalan di sistem operasi *Linux*, *Windows*, dan *MacOS*. Fitur bawaan di versi terbaru *Android Studio* Terbaru, tetapkan tata letak, *string* UI, dan gambar *bitmap* (Ramayani, 2021).

Diluncurkan pada tanggal 16 Mei 2013 dalam konferens *Google I/O* yang pada waktu itu dimana masih dalam tahap pratinjau akses versi 0.1 hingga ahir nya versi stabil 3.0 yang dimna di rilis pada pertengahan bulan Oktober 2017 dan ini menjadi salah satu *software* terlaris di kalangan muda pada saat ini, aplikasi ini dapat dapat beroperasi di berbagai sistem separate *Windows, macOS, dan Linux*(Helma, Rizka, n.d.)

Dapat disimpulkan bahwa sejatinya aplikasi *Android Studio* ini tdak memerlukan tempat penyimpanan yang cukup besar, bisa di katakan tidak akan menghabiskan kapasitas memori. Dan *Android Stusio* ini juga sangat mempermudah dan mendukung pengembangan aplikasi (Yudho, 2019).

2. *My SQL*

My SQL adalah aplikasi perangkat lunak yang di dalam nya terdapat sistem *management* basis data *SQL (Database Management System)* biasa di sebut DBMS. Aplikasi memiliki fungsi untuk mengatur, membuat dan mengelolah *database*. (Raharjo, 2011) *My SQL* adalah salah satu berhubungan dengan *database* pengelolah sistem (RDBPS) yang dimana pada saat ini sangat banyak diminati oleh para *depelover* aplikasi *database*, baik untuk aplikasi web maupun berbasis dekstop untuk menyimpan, mengelolah, dan mengatur data pada aplikasi tersebut. *My SQL* memiliki beberapa kelebihan dari RDBPS lain adalah mudah, gratis, simpel, portabel, dan *efesiensi*.



Gambar 2.9 Logo *My SQL*

Sumber : (Data Penelitian, 2021)

SQL (Structured Query Language) adalah bahasa yang tersusun untuk mengakses atau menggunakan data-data pada database dan entitas-entitas yang ada pada database tersebut. SQL juga memiliki arti sebagai bahasa baku yang digunakan dalam aneka macam database karena itu akan relatif mudah dalam menggunakan meskipun berpindah dari *database* yang satu ke *database* yang lain. Index di *SQL-Server* diartikan sebagai objek database yang dibuat dengan tabel serta kolom. Index di database sangat bermanfaat untuk mencari nilai kolom pada tabel tertentu dengan cepat karena jika tidak menggunakan index maka database harus melakukan pencarian dari tabel baris pertama hingga tabel baris paling akhir dan akan menggunakan waktu yang cukup lama. Selain itu juga, tanpa index akan membutuhkan kapasitas memori yang besar apabila dilakukan pencarian pada tabel yang mempunyai baris yang banyak. Maka bisa dikatakan bahwa index ini mirip dengan daftar isi di sebuah buku (Pamungkas, 2018).

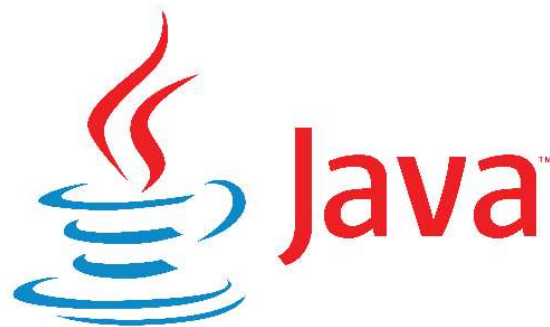
3. *Java*

Java adalah *software* yang dapat di jalan kan di bermacam-macam komputer dan termasuk juga Hp pintar (*Smartphone*). *Software* ini juga termasuk dalam bahasa pemrograman yang bersifat *General Purpose* dan di buat secara kusus untuk memanfaatkan implementasi se minimal mungkin.

Mengadopsi *E-Book* (Haqi, 2019) *Java* adalah sebuah aplikasi yang di perkenalkan oleh Sun Microsystem Inc yaitu sekitar pertengahan tahun 1990. Menurut Sun, java adalah nama untuk sekumpulan *tecnologi* yang membuat dan menjalankan *software* pada komputer ataupun pada lingkungan jaringan. Java dirilis pada Agustus 1991 dari perusahaan Sun Microsystem Inc yang di pimpin oleh James Gosling. Dengan tanda “*Write Once, Run Anywhere*” (WORA) untuk melambangkan kelebihan yaitu *java* merupakan bahasa pemrograman yang bersifat *multiplatform*. *Java* diberi lisensi GPL (*General Public License*) yang menjadikan mengedit dan menulis aplikasi secara *open source* dan juga di bekali dengan dua kemampuan handal yaitu *Java Virtual Machine* dan gaya bahasa C/C++. Sebagai Bahasa pemrograman berorientasi objek , java menggunakn kelas untuk membentuk suatu objek (Rahmawati et al., 2015).

Menurut (Sallaby et al., 2015) *Java* adalah salah satu Bahasa pemrograman *scripting* yang sering di gunakan untuk pembuatan aplikasi *Android*, *java* bisa berfungsi menjadi penambah tingkah laku supaya *widged* bisa tampil lebih menarik. *Java* sangat diminati karena java bisa dijalankan diberbagai *platform* sistem operasi dan dikenal dengan istilah *Write One, Run Anywhere* sebab kompatibilitasnya.

Penulis (Ramayani, 2021) mengambil kesimpulan bahwa *java* bisa mengedit dan menulis aplikasi secara *open source* serta memiliki dua kemampuan handal yaitu *Java Virtual Machine* dan *C/C++* dan juga *Java* memberikan pilihan berbagai jenis model dalam membuat aplikasi *Mobile* dan *Client-Server*.



Gambar 2.10 Logo *Java*

Sumber : (Data Penelitian, 2021)

2.2.3 Objek Penelitian

Penelitian ini dilakukan di PT. SP Manufacturing yang berlokasi di Jln. Citra Buana Central Park lot.3, 5, Batam Centre, Batam, Kepulauan Riau.

1. Sejarah Perusahaan

PT. SP Manufacturing Batam adalah perusahaan yang bergerak di bidang Manufactur, dimana perusahaan ini membuat barang-barang *electronic* seperti PCBA, *Wire Harnes* dan lain sebagainya. Perusahaan ini berdiri sejak 2 Maret 2016, yang dimulai dari 10 orang karyawan dan dimana sekarang sudah mencapai kurang lebih dari 750an karyawan. PT SP Manufacturing

Bata mini tergolong kepada perusahaan yang memproduksi dan memasarkan produk utama yaitu seperti PCBA, dan *Wire Harnes*.

2. Visi dan Misi

a. Visi

Perusahaan yang menjadikan salah satu perusahaan yang terbaik di industry manufactur Indonesia yang sangat menjunjung nilai keselamatan dan Kesehatan kerja.

b. Misi

- Menjadikan perusahaan manufactur yang paling sukses dan paling menguntungkan dalam bidang industry *electronic*
- Menghasilkan produk yang baik dan memberikan kesejahteraan dan hak-hak seluruh karyawan.



Gambar 2.11 PT. SP Manufacturing Batam

Sumber: (Data Penelitian, 2021)

2.3 Penelitian Terdahulu

Dalam sebuah penelitian harus memiliki bahan perbandingan dan referensi dari beberapa jurnal yang telah diteliti oleh para peneliti sebelumnya, disini ada beberapa jurnal sebagai bahan perbandingan sebagai berikut:

1. (Bambang, Fuad, 2015) **Pengembangan QR Code Scanner Berbasis Android Untuk Sistem Informasi Museum Sonobudoyo Yogyakarta**, ISSN: 1829-667X. Dalam penelitian ini sistem manajemen koleksi Museum hanya orang-orang tertentu yang dapat mengakses museum tersebut maka dari itu dirancanglah aplikasi untuk menampilkan informasi detail tentang benda-benda koleksi pameran di Museum Yogyakarta ini, sehingga pemandu dan pengunjung bisa mendapatkan informasi lebih detail tentang benda-benda koleksi pameran.
2. (Afizi et al., 2017) **Designing Intelligent Secure Android Application for Effective Chemical Inventory**, Conference Series 226 (2017) 012086 doi: 10.1088/1757-899X/226/1/012086. Penelitian ini bertujuan untuk mempermudah pengelolaan stok bahan kimia, karena pada hari-hari sebelumnya sistem pengelolaan *inventory* bahan kimia dalam penulisan bahan stok dan dalam hal pembelian, sangat banyak unit yang hilang dan kadaluarsa. Dengan adanya aplikasi ini pelacakan bahan *inventory* kimia menjadi lebih mudah, dan membuat karyawan laboratorium lebih mudah memantau dalam penggunaan bahan kimia, dengan aplikasi *android* ini untuk sistem manajemen

inventaris bahan kimia, dapat menjaga kepatuhan keselamatan laboratorium lebih efisiensi.

3. (Ismasari, Adi, 2018) **Aplikasi *Stock Opname* Berbasis Android Dengan Teknologi *QR Code* Menggunakan Metode *Prototype*** (Studi Kasus: PT. APM Armada Autoparts), ISSN: 2407-3903. Penelitian ini di rancang untuk melakukan perhitungan ulang stok barang yang bertujuan untuk mengetahui jumlah stok yang ada. Aplikasi *Stock Opname* ini dibuat karena ketika sedang tutup atau tidak dalam kondisi yang ramai, hal ini mejadi penyebab perusahaan tidak teratur dalam melakukan *Stock Opname*, dengan aplikasi ini dapat menampilkan informasi terperinci dan pencatatan stok barang dapat langsung di simpan ke *database*, sehingga dapat menggantikan sistem basis data yang telah ada, dan makin mempercepat pengelolaan dan dapat mengurangi tingkat kesalahan pada pencatatan *Stock Opname*. kimia, dapat menjaga kepatuhan keselamatan laboratorium lebih efisiensi.
4. (Irmawati et al., 2019) **Rancang Bangun Aplikasi *Stock Opname* Pada PT. Arie Muti Berbasis *Android***, ISSN: 0216-6933. Aplikasi ini dirancang karena perusahaan tidak disiplin dalam melaksanakan *Stock Opname*, maksud dilakukan nya Perancangan Aplikasi *Stock Opname* berbasis *Android* ini, agar dapat mempermudah melakukan pengelolaan *Stock Opname* dan diharapkan dapat melakuakan pengelolaan data *Stock Opname* lebih elastis karena computer tidak akan digunakan lagi dalam melakukan pengelolaan data, akan tetapi hanya dengan menggunakan *Smartphone* berbasis *android*.

5. (Lainama, Dinata, 2020) **Rancang Bangun Aplikasi Penyimpanan Stok Kacamata Dengan Menggunakan Kode QR Berbasis *Android***, ISSN: 2460-1306. Penelitian ini di rancang menggunakan aplikasi dengan *QR Code* berbasis *Android* agar mempermudah penyimpanan stok kacamata yang di miliki oleh pemilik toko karena tidak mengetahui jumlah stok kacamata dengan pasti, dengan aplikasi *QR Code* ini sangat membantu pengguna untuk melihat data barang dengan cepat.
6. (Dodi Vebri, Radius, 2020) **Sistem Manajemen Hasil Produksi Berbasis *Android* Menggunakan Teknologi *QR Code* di Pt. Pura Nusapersada, P-** ISSN: 2580-7927 | E-ISSN: 2615-2738. Penelitian ini membahas tentang permasalahan sering nya terdapat perbedaan data antara bagian produksi dengan bagian gudang saat membuat laporan bulanan, dengan sistem manajemen menggunakan *QR Code* berbasis *Android* dapat memngurangi kemungkinan kesalahan data, dan akan sangat mudah dan cepat pendataan Melalui Aplikasi Berbasis Web Ini.
7. (Muhammad et al., 2020) **Rancang Bangun Aplikasi Pemandu Wisata Museum Sumbawa Berbasis *Android* Dengan Memanfaatkan *QR Code***, ISSN: 2686-3359. Penelitian ini dirancang menggunakan aplikasi pemandu wisata berbasis *Android* dengan menggunakan kode QR dengan salah satu permasalahan nya, tidak adanya informasi-informasi yang detail di tulis pada setiap barang koleksi Museum Sumbawa, dan terbatas nya jumlah pemandu mueum yang bisa menjelaskan informasi-informasi detail dari setiap koleksi Museum, dengan rancangan aplikasi Pemandu Wisata berbasis android ini,

dapat mempermudah dan membantu kinerja pengelola museum dan mempermudah pengunjung museum untuk mendapatkan informasi.

8. (Rahmalisa et al., 2020) **Aplikasi Absensi Guru Pada Sekolah Berbasis Android Dengan Keamanan QR Code (Studi Kasus: SMP Negeri 4 Batang Gansal)** p-ISSN: 2460-0679 | e-ISSN: 2477-6890. Saat ini pengambilan kehadiran guru di Sekolah SMP Negeri 4 Batang Gansal ini masih manual yaitu menggunakan catatan dengan kertas, dengan cara ini dapat memicu terjadinya kecurangan dan kurang disiplinnya sehingga tidak terkontrol. Dengan pemanfaatan teknologi dibuatlah aplikasi absensi guru berbasis *android* sehingga dengan aplikasi ini bisa menampilkan daftar informasi absensi dalam bentuk grafik, sehingga lebih meningkatkan kedisiplinan pada kehadiran di sekolah SMP Negeri 4 Batang Gasal.
9. (Ramayani, 2021) **Aplikasi Pencatatan Kerusakan Dan Perbaikan Mesin Pendingin Berbasis Android** ISSN (online) 2715-6265. Saat ini pencatatan kerusakan mesin pendingin pada salahsatu Hypermart di Batam masih menggunakan manual, sehingga sering terjadi kesalahan seperti lupa mengisi data *Record* di dalam *form service report*, pelaporan data dalam satu bulan tidak terdata atau kertas yang di gunakan hilang. Dengan aplikasi ini dapat menghemat penggunaan media kertas, dan dapat membantu mempermudah pekerjaan *maintenance* dalam melakukan pencatatan karena sudah menggunakan *android*.

2.4 Kerangka Pemikiran

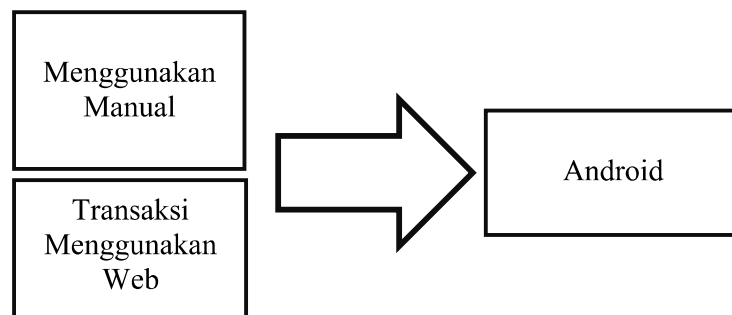
Dalam metode kerangka pemikiran ini akan dijelaskan diagram tentang alur logika dari penelitian yang di lakukan. Rancangan dalam kerangka ini akan menggambarkan hubungan antara variabel satu dengan variabel yang lain.

Dengan latarbelakang di atas, dapat diidentifikasi beberapa problem yang bisa di ambil dan di simpulkan dalam penelitian ini yaitu:

1. Pencatatan stok material masih menggunakan manual dengan *Stock Card*.
2. Dalam transaksi material masih menggunakan Web sehingga harus melakukan 2 kali transaksi.
3. Sering terjadi lupa mencatat matrial yang di ambil.

Karena belum efektif sistem pengelolah material, maka di butuhkan aplikasi yang dapat membantu dan mempermudah kerja karyawan.

Berikut ini adalah contoh gambar bagian dari kerangka pemikiran:



Gambar 2.12 Kerangka Pemikiran

Sumber : (Data Penelitian, 2021)