BAB II

KAJIAN PUSTAKA

2.1 Teori Dasar

Dibutuhkan beberapa teori dasar yang mendukung tahapan penelitian dengan menggunakan beberapa komponen yang meliptui seperti *Android, UML, dan black box testing.*

2.1.1 Android

Android mempunyai berbiaya yang tidak mahal, sehingga menjadi pilihan bagi perusahaan teknologi, Andoid mendorong munculnya aplikasi yang dibuat berdasarkan kode sumber terbuka.

Saat ini Android merupakan sistem operasi berbasis *linux* Sistem operasi Android dikembangkan khusus untuk perangkat seluler yang mendukung sistem kerja dengan layar sentuh. Saat ini pun, pengguna ponsel telah menggunakan sistem operasi tersebut. Adapun berbagai macam *gadget* menggunakan Android sebagai perangkat platformnya, mulai dari ponsel pintar, tablet, PC, jam tangan, TV hingga kamera dan perangkat teknologi lainnya. Terhitung sejak tahun 2013, 79% *market share* telah dikuasai oleh platform nomor satu hingga saat ini, platform tersebut merupakan Android. Sebagai sistem operasi, Android berfungsi sebagai *device* atau penghubung antara pengguna dan perangkat keras yang ada pada *smartphone* atau alat elektronik tertentu (Firly, 2018).

Sistem operasi Android awal mula dikembangkan oleh Android Inc. Selanjutnya, diambil alih oleh *Google* dengan sistem operasi pada tahun 2005

sebagia sistem operasi yang bersifat "*Open Source*". Sistem operasi tersebut dapat dimanfaatkan secara gratis. Tidak hanya ditujukan untuk Android ponsel saja, tetapi juga perangkat elektronik yang bergerak lainnya (Eko, 2019).

(Eko, 2019) menyimpulkan bahwa android memiliki daya pikat pada platform *opensource* karena banyak membuka peluang besar bagi semua pengembnag teknologi. Hal ini bertujuan agar dalam membuat dan mengembangkan fitur aplikasi dapat digunakan oleh seluruh pengguna Android.

Gambar 2.1 Logo *Android*

Sumber : (Data Penelitian, 2021)

2.1.2 *Unified Modeling Langueage* (UML)

Unified Modeling Languange (UML) mempunyai standar untuk menspesifikasi, serta membangun sistem perangkat lunak, dan memiliki alat pemodelan berbasis visual agar membantu proses pengembangan sistem.

Unified Modeling Langueage (UML) merupakan alat untuk mengembangkan sistem informasi dalam menggambarkan, merancang, dan mendokumentasikan sistem perangkat lunak. Perangcangan sistem dapat dibentuk

dalam proses yang digunakan untuk memahami kebutuhan sistem (Maharani, 2018).

UML masuk kedalam pengembangan yang menggunakan bahasa pemodelan dengan sistem perangkat lunak, UML juga memiliki pemodelan yang penting yaitu untuk menjelaskan aspek fungsionalitas sistem, dan UML menyediakan banyak diagram yang diperlukan guna menjelaskan sistem yang sedang dikembangkan (Kurniawan, 2018).

Ada beberapa model pendekatan dari UML, di bawah ini merupakan pendekatan dengan menggunakan *Use case Diagram, Activity Diagram, Class Diagram,* dan *Squence Diagram.*

1. *Use case diagram*

Use case Diagram adalah permodelan pendekatan sistem yang menggambarkan interaksi pengguna. *Use case Diagram* menerangkan aktor yang terlibat dan fungsi yang dapat digunakannya aktor tersebut.

Diagram *Use Case* terdiri dari:

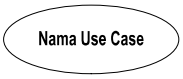
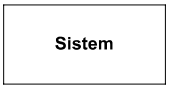
a. *Use Case*

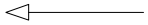
b. Actor

c. *Relationship*

d. *System boundary /* batas sistem (opsional)

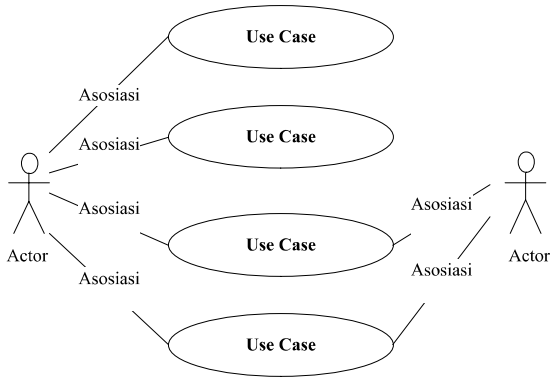
Tabel 2.1 Simbol-simbol *Use case diagram*

Simbol	Keterangan
	*Use Case* ▪ Sebagian besar yang mewakili sistem fungsional ▪ Batas sistem yang ditempatkan didalam ▪ Penamaan *usecase* yaitu label atau kata kerja dan juga diikuti kata benda
	*System Boundary* ▪ Nama pada *sistem boundary* terdapat dibagian atas ▪ Digambarkan dengan ruang lingkup sistem
	Asosiasi Boundary ▪ Menghubungkan antara usecase dengan aktor yang saling berinteraksi
<<include>>	*Include* ▪ Arah panah mengarah kepada *main use case.*
	▪ Relasi menggambarkan sebuah use case (*sub use case*) fungsinya aka dijalankan terlebih dahulu.
<<extend>>	EXTEND ▪ Relasi yang menggambarkan sebuah use case berdiri sendiri tanpa main use case dijalankan terlebih dahulu.
	▪ Menggambarkan relasi main *use case* yang bisa berdiri sendiri tanpa dijalankan terlebih dahulu.

	GENERALISASI / *GENERALIZATION* ▪ Menghubungkan antara use case yang umum dengan use case khusus.

Sumber : (Maharani, 2018)

Penerapan *Usecase Diagram* seperti pada gambar di bawah ini:



Gambar 2.2 Contoh *Use case diagram*
Sumber : (Data Penelitian, 2021)

1. *Activity diagram*

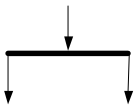
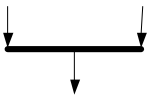
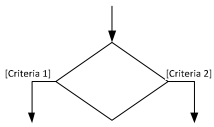
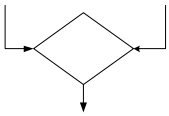
Activity diagram merupakan aksi yang proses kerja atau aktivitas dari sebuah sistem. *Activity Diagram* dibuat berdasarkan alur *Use case Diagram* agar dapat mempermudah dan memahami alur proses sistem.

Tabel 2.2 Simbol-simbol *Activity diagram*

Simbol	Keterangan
	Start poin ▪ Awal penelusuran ▪ *Star poin* untuk memulai sebuah aktivitas ▪ Hanya boleh digunakan 1 simbol pada sebuah

	aktivitas.
	*End Poin* ▪ Akhir penelursuran ▪ *End Poin* akhir aktivitas diagram ▪ >1 simbol *End Poin* hanya boleh digunakan pada sebuah aktivitas.

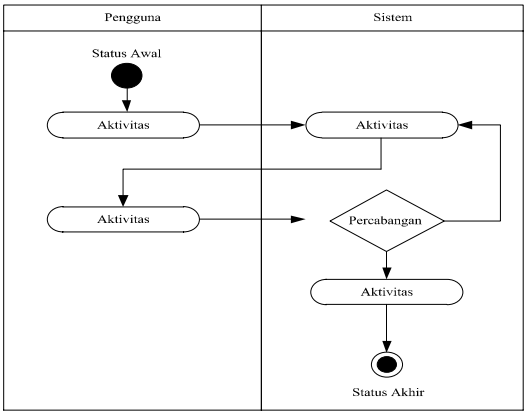
Tabel 2.2 (Lanjutan)

	*Activities* ▪ Akhir penelusuran ▪ Menggambarkan tentang aktivitas dari kata kerja ▪ Hanya memiliki sebuah aktivitas yaitu satu alur masuk dan satu alur keluar
	*Fork* ▪ Percabangan ▪ Satu aliran yang dikerjakan secara bersamaan
	*Join* ▪ Penggabungan ▪ Aliran yang disatukan untuk melanjutkan aktivitas
	*Decision poin* ▪ Pada tengah belah ketupat tida ada keterangan (pertanyaatn) *guards* harus dimiliki oleh *flowchart*
[ ]	*Guarid* ▪ Sebuah transisi ketika dilewati sebuah kondisi benar
	*Merge* ▪ Jalur keputusan akan kembali dan melewati *decision poin*

Swimlane	*Swimlane* ▪ Aktor yang didasarkan dan dikelompokan sebuah cara aktivitas ▪ Actor bisa ditulis nama actor ▪ Digambarkan secara horizontal dan vertikal dari *swimlame*

Sumber : (Maharani, 2018)

Penerapan *Activity diagram* seperti gambar di bawah ini:



Gambar 2.3 Contoh *Activity diagram*
Sumber : (Data Penelitian, 2021)

2. *Sequence diagram*

Sequence diagram menggambarkan interaksi yang ada didalam dan sekitar sistem. Jumlah gambaran *Squence Diagram* harus sama dengan jumlah *Use case Diagram* bertujuan untuk menggambarkan scenario pada *Use case Diagram*.

Tabel 2.3 Simbol-simbol *Sequence diagram*

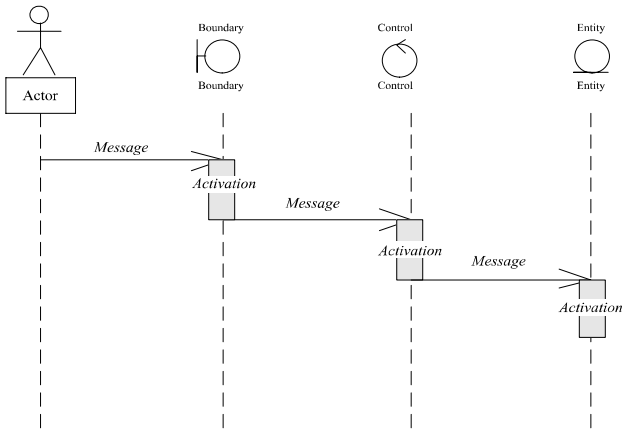
Simbol	Keterangan
	*Actor* ▪ Seorang yang berinteraksi dengan sistem
	*Boundary* ▪ Penghubung antara aktor dengan sistem
	*Control* ▪ Prilaku suatu sistem yang telah diatur aur kerja nya oleh suatu sistem

Tabel 2.3 (Lanjutan)

	*Entity* ▪ Suatu sistem yang menyimpan sebuah informasi ▪ Suatu sistem oleh struktru data yang digambarkan *entity*.
	*Activation* ▪ Suatu objek yang digambarkan kondisi interaksi ▪ Sebuah operasi oleh durasi aktif dengan berbanding lurus dengan panjang simbol.
	*Message* ▪ Urutan kejadian yang digambarkan oleh pesan antar objek

Sumber : (Maharani, 2018)

Penerapan *Sequence diagram* seperti gambar di bawah ini:



Gambar 2.4 Contoh *Sequence diagram*
Sumber : (Data Penelitian, 2021)

3. *Class diagram*

Class diagram menggambarkan tentang struktur dan pendefinisisan kelas, *package* dan objek hubungan kelas satu dengan kelas yang lain. *Class Diagram* terdiri dari tiga area yaitu, Nama, Atgribut (Variabel yang dimiliki oleh satu kelas) dan Metode / *Operation* (Fungsi yang dimiliki oleh suatu kelas).

Simbol-simbol *Class Diagram* dapat di lihat dalam tabel berikut:

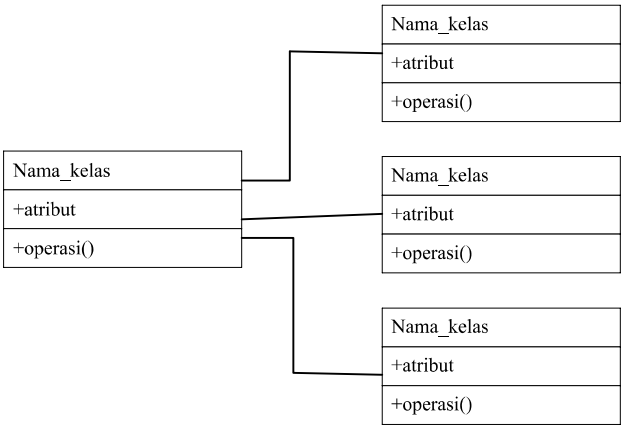
Tabel 2.4 Simbol-simbol *Class diagram*

Simbol	**Deskripsi**
Nama_kelas +atribut +operasi()	▪ Struktur sistem yang terdapat pada kelas
Antarmuka / *interface* Nama_interface	▪ Pemrograman berorioentasi objek pada konsep *interface*

Asosiasi/ *association*	▪ Makna umum dengan relasi antar kelas, asosiasi biasanya dilengkapi dengan *multiplicity*
Asosiasi berarah / *directed association*	▪ Satu kelas dengan kelas yang lain bermakna relasi antar kelas, *multiplicity* biasanya juga terdapat pada asosiasi
generalisasi	▪ Generalisasi-spesialisasi (umum khusus) bermakna dengan relasi antar kelas
Kebergantungan/ *dependency*	▪ Antar kelas tergantung pada makna dengan relasi antar kelas
Agregasi/ aggregation	▪ Relasi antar kelas dengan makna semua-bagian (*whole-part*)

Sumber : (Maharani, 2018)

Penerapan *Class Diagram* seperti gambar di bawah ini:



Gambar 2.5 Contoh *Class diagram*
Sumber : (Data Penelitian, 2021)

2.1.3 Pengujian Aplikasi

Black box testing merupakan alat pengujian yang fungsional yang menggunakan perangkat lunak untuk menguji dan mengetahui struktur internal kode program.

Dalam pengujian aplikasi yaitu menggunakan pengujian tes kotak hitam, *black box* dapat digunakan untuk menguji perangkat lunak yang bersifat *Open Source*. Dalam pengujian *black box testing* berfokus pada perangkat lunak yang memiliki spesifikasi fungsional dari perangkat lunak(Mustaqbal, Firdaus, & Rahmadi, 2015).

Menurut (Mustaqbal et al., 2015) *black box testing* juga dapat diartikan sebagai tester yang mendefinisikan kumpulan kondisi dari pengetesan pada spesifikasi yang fungsional.

1. Fungsi yang tidak benar atau salah
2. (*Interface errors*) kesalah pada antarmuka
3. Kesalahan yang ada pada struktur data dan akses database
4. Kesalahan kinerja (*Performance errors*)
5. Kesalahan pada inisialisasi dan dan juga terminasi.

(Mustaqbal, Firdaus, & Rahmadi 2015) menyimpulkan bahwa *black box* digunakan untuk menguji perangkat lunak yang bersifat *open source* dan memiliki spesifikasi yang fungsional.

2.2 Teori Khusus

Teori khusus merupakan teori pendukung yang paling utama untuk terlaksananya penelitian ini.

2.2.1 Mesin Pendingin

Mesin pendingin merupakan kulkas sebuah alat rumah tangga yang berdaya listrik menggunakan *refrigerasi* atau proses pendingin yang berguna untuk membantu mengawetkan makanan atau minuman. Adapun kulkas atau mesin pendingin dibidang industri.

Menurut (Kusbandono & Purwadi, 2016) mesin pendingin atau *showcase* memiliki fungsi yang digunakan untuk mendinginkan minuman kemasan sehingga orang yang meminum minuman tersebut mendapatkan kesegaran, seperti mesin *showcase* yang di gunakan untuk mengawetkan bahan makanan dan minuman . Didalam mesin pendingin atau *showcase* biasanya hanya digunakan untuk mendinginkan minuman.

Mesin pendingin yang digunakan terkhusus untuk pendingin makanan dan minuman, mesin pendingin yang masih menggunakan refrigen masih memerlukan banyak energi listrik sehingga pemakaian listrik menjadi boros (Mirmanto, 2018).

Penulis mengambil kesimpulan menurut (Kusbandono & Purwadi, 2016) bahwa mesin pendingin mempunyai fungsi bermanfaat bagi kehidupan sehari-hari karena dengan mesin pendingin kita bisa mengawetkan bahan makanan dan minuman agar lebih tahan lama.

2.2.2 Pencatatan

Pencatatan merupakan proses mengisi data kedalam media sistem pencatatan data atau mendokumentasi suatu aktivitas, jika sistem pencatatan data tersebut masih menggunakan buku maka data dilakukan dengan menulis pada lembaran buku secara manual sedangkan jika sistem pencatatan menggunakan perangkat komputer.

Menurut (Saepudin et al., 2020) pencatatan secara manual mengakibatkan sering terjadinya kesalahan atau selisis data, penggunaan kertas yang sangat boros, karena harus merekap data beberapa salinan untuk di distribusikan ke bagian yang lain. Dari permasalah tersebut, penelitian ini di buat bertujuan agar pencatatan data lebih akurat dan cepat, selain itu juga data bisa tersimpan dengan baik dan aman di dalam *database.*

Pencatatan kerusakan secara manual masih sering digunakan manusia sepenuhnya, itu salah satu penyebab proses pencatatan perbaikan yang menghabiskan waktu yang lama (Mahardika, Sari, & Dewi, 2018).

Penulis mengambil kesimpulan menurut (Saepudin et al., 2020) karena dalam proses pencatatan secara manual masih memungkinkan banyaknya terjadi kesalahan atau selisi data, dan penggunaan secara manual menggunakan kertas juga sangat boros karena harus merekap data salinan yang banyak menghabiskan kertas.

2.2.3 Software Pendukung

Software pendukung untuk membuat aplikasi yang berbasis *android* di perlukan beberapa *software,* yaitu *Android Studio* dan *java SQLite.*

2.2.3.1 *Android Studio*

Android Studio merupakan perangkat lunak yang dikembangkan oleh *Google*. *Android Studio* merupakan IDE pemrograman android yang menggantikan dari yang sebelumnya adalah *Eclipse*. *Google* menghentikan pengembangan terhadap *Eclipse* dan berfokus hanya kepada pengembangan *Andorid Studio* saja. Hal ini dikarenakan *Android Studio* memiliki fitur yang mudah sehingga menunjang para pembuat program level dasar hingga ahli. *Android Studio* juga dilengkapi dengan *library* yang bisa langsung digunakan oleh para pengembangan aplikasi.

Gambar 2.6 Logo *Android Studio*
Sumber : (Data Penelitian, 2021)

Android Studio terintegritas resmi untuk sistem operasi android untuk disematkan untuk pengembangan android yang dibangun pada perangkat lunak Jet Brains'Inttellij IDE (*Integrated Development*

Environment) yang berfungsi sebagai *platform* untuk aplikasi android. *Android Studio* yang dipublikasikan oleh Google diwadahi oleh lisensi Apache 2.0 dengan pengembangan lanjutkan sehingga *Android Studio* dijalankan menggunakan sistem operasi Linux, Windows dan MacOS. Pada versi terbaru dari *Android Studio* ditanamkan didalamnya fitur terbaru, mengalokasikan tata letak, *string UI* dan gambar bitmap (Yudho, 2019).

Para pengembang aplikasi berbasis android yang menggunakan perangkat lunak *Android Studio* memerlukan sebuah. Dengan menggunakan bahas *Java* dapat berjalan di mesin Dalvik. Tutorial dalam membangun aplikasi berbasis android. Peranan yang dilakukan oleh SDK memungkinkan untuk pengembang agar membuat aplikasi yang mecakup sampel dengan kode sumber. (Maiyana, 2018).

Kesimpulan diambil menurut (Yudho, 2019) Sejatinya aplikasi *Android Studio* memerlukan tempat penyimpanan yang menghabiskan *memory*. Terlepas dari hal itu, *Android Studio* mempunyai kelebihan-kelebihan yang mempermudah untuk mendukung pembuatan dan pengembangan aplikasi.

2.2.3.2 *Java*

Java dapat dijalankan diberbagai macam komputer termasuk juga *smartphone*, *java* termasuk kedalam nahasa pemrograman bersifat

general purpose dan didesain secara khusus agar dapat mengimplementasikannya secara minimal.

Mengadopsi dari E-Book (Hadiprakoso, 2021a). Versi pertama Java, yakni Java 1.0 dirilis untuk umum pada tahun 1995 dari perusahaan Sun Microsystem Inc untuk melambangkan kelebihan bahwa java merupakan bahasa pemrogramana yang bersifat multi platform. Java diwadahi dalam lisensi GPL (*Ganeral Public License*) yang menjadikan menulis dan mengedit aplikasi secara *opensource* serta didukung dengan dua kemampuan handal yakni disematkanya *Java Virtuaol Machine* dan gaya bahasa c/c++. Bahasa pemrograman java menyediakan berbagai jenis edisi dalam membuat aplikasi *mobile* hingga membuat aplikasi *client-server*. Dalam bahasa pemrograman Java, kode program di-compile terlebih dahulu oleh Java compiler (Javac) yang masuk kedalam bentuk bytecode (file dengan ekstensi class). Untuk menjalankan program bytecode diinterpretasikan oleh Java Virtual Machine. Diperlukan perangkat lunak Java Virtual Machine (JVM) termasuk kedalam JRE dan Java Class Library (JCL) merupakan sekumpulan yang dibutuhkan dalam menjalankan program Java. Dalam membuat sebuah program Java dibutuhkan JDK (Java Develoment Kit). JDK terdiri dari JVM dan JCL serta Java compiler (Javac) yang diperlukan pada saat meng-compile kode program.

Java saat ini merupakan bahasa yang sangat populer dan *java* diciptakan dengan beberapa edisi, contohnya yaitu J2EE digunakan untuk

Aplikasi *Enterprise* dan J2ME di gunakan untuk Aplikasi *Mobile. Java* populer dikarenakan java dapat dijalankan diberbagai platform sistem operasi dan dikenal dengan istilah *Write One, Run Anywhere* karena kompabilitasnya (Adam, Firman 2018).

Penulis mengambil kesimpulan menurut (Hadiprakoso, 2021b)) karena menulis dan mengedit aplikasi secara *opensource* serta didukung dengan dua kemampuan handal yakni disematkanya *Java Virtuaol Machine* dan gaya bahasa c/c++. Bahasa pemrograman java menyediakan berbagai jenis edisi dalam membuat aplikasi *mobile* hingga membuat aplikasi *client-server*. Dalam bahasa pemrograman Java, kode program di-compile terlebih dahulu oleh Java compiler (Javac) yang masuk kedalam bentuk bytecode (file dengan ekstensi class).

Gambar 2.7 Logo *Java*
Sumber : (Data Penelitian, 2021)

2.2.3.3 *SQLite*

SQLite atau sering disebut mesin database merupakan penyimpanan *internal device* yang andal, mandiri, cepat dan berfitur lengkap. File database *SQLite* sebagai wadah untuk mengirim projek agar dapat dibaca pada antar sistem. yang digunakan ialah pemanggilan secara langsung melalui pemrograman API. Mekanisme seperti itu pasti akan membawa dampak baik karena bisa mengurangi *Overhead*, *latency time,* dan lebih sederhana dalam keseluruhan.

SQLite ialah database *open source* yang di sematkan di android. *SQLite* mendukung fungsi database relasional standart, seperti sintaks *SQL*, fungsi transaksi dan fungsi pernyataan yang di siapkan. Selain itu, sedikit saja memori saat *runtime. SQLite* merupakan pustaka perangkat lunak yang dapat berdiri sendiri, tanpa server tidak perlu mengkonfigurasi, dan mesin transaksi database *SQL*. *SQLite* juga termasuk kedalam mesin database yang paling populer yang digunakan di dunia (Putra, Budi, & Kadafi, 2020).

SQLite sering disebut sebagai sistem manajemen database yang *embedded* yang ringan karena tidak memerlukan aplikasi server khusus, program *engine* SQLite dapat disebut bagian dari aplikasi. SQLite merupakan sebuah *librari in-process* yang mengimplementasi *engine* database yang *self-contained*. SQLite mempunyai kode yaitu pulic domain, karena bebas untuk digunakan baik secara pribadi maupun komersial (Sidik, 2020).

Kesimpulan yang diambil menurut (Sidik, 2020) karena SQLite tidak memerlukan aplikasi yang khusus pada server dan SQLite juga bebas digunakan secara pribadi maupun komersial.

Gambar 2.8 Logo *SQLite*
Sumber : (Data Penelitian, 2021)

2.3 Penelitian Terdahulu

Penelitian terdahulu sebagai referensi utama untuk pengembangan aplikasi untuk melakukan penelitian:

1. (Saepudin et al., 2020) **Perancangan aplikasi pencatatan data kerusakan produksi PT Haeng Nam berbasis web,** e-ISSN: 2715-8756. Penelitian ini dikembangkan untuk mengembangkan sebuah sistem baru yang terkomputerisasi dalam pengolahan dan pencatatan data kerusakan secara online. Karena saat ini sistem pengolahan data laporan masih menggunakan sistem secara manual. Bahasa pemrograman yang digunakan untuk

merancang aplikasi pencatatan yaitu bahasa pemrograman *JavaScript* dan *PHP*, untuk menyimpan data dalam *database* menggunakan *MySQL*. Hasil dari penelitian ini ialah admin bisa mengakses data pencatatan kerusakan produksi dengan mudah melalui *website*.

2. (Soewito, Gunawan, & Rusli, 2019) **use of android smart phones as a tool for absences**, Procedia Computer Science: 157 (2019) 238-246. Saat ini sistem absensi masih memiliki kendala terutama absensi bagi karyawan yang bekerja diluar kantor. Semua karyawan memiliki satu *smartphone* maka dalam penelitian ini penulis memperkenalkan sistem absensi menggunakan *smartphone*, sistem kehadiran yang telah dirancang dapat menghemat 66% waktu untuk mencatat laporan, dan sistem ini juga dapat menampilkan data riwayat kehadiran termasuk laporan bagi setiap karyawan yang mendukung penerapan sumber daya manusia. Sistem absensi ini juga bisa mengurangi resiko yang sering terjadi seperti, menipulasi data.
3. (Pang, Forrest, Lê-Scherban, & Masino, 2021) **Prediction of early childhood obecity with machine learning and electronic healt record data,** 150 (2021) 104454. Penelitian ini memiliki tujuan yaitu membangdingkan tujuh model pembelajaran mesin yang dikembangkan untuk memprediksi obesitas pada anak. Kesimpulan yang dapat diambil pada penelitian ini yaitu prediksi obesitas anak usia dini yang dikembangkan dari kohort terbesar dan dilaporkan hingga saat ini.
4. (Shoenbill et al., 2020) **Identifying patterns and predictors of life style modification in electronic,** 136 (2020) 106061. Penelitian ini bertujuan

untuk menganalisis menggunakan metode statistik pembelajaran mesin untuk mengidentifikasi prediktor dan waktu untuk memodifikasi gaya hidup. Catatan kesehatan elektronik dapat meningkatkan pemahaman tentang waktu modifikasi gaya hidup dan karakteristik pasien, ini dapat menginformasikan info perawatan dapam proses perawatan pelaksanaan pengobatan dan akhirnya kontrol hipertensi.

5. (Ejiyi et al., 2021) **Design and develoment of android application for educational institutes,** Conference Series 1769(2021)012066 doi: 10.1088/1742-6496/1769/1/012066. Dasar aplikasi pemrograman Java merupakan bahasa pemrograman yang digunakan untuk menulis aplikasi android. Karena eksekusinya yang efektif, Java *Virtual Machine* (JVM) standart digunakan. *Google* telah menciptakan mesin virtual (VM) yang telah disesuaikan yang disebut *Dalvik*. VM Dalvik meluncurkan *bytecode Java* dan menggunakan atribut inti *Linux* seperti multitreding dan manajemen memori dibangun dalam bahasa *Java*. Android berisi beberapa pustaka yang berbasis Java dikhususkan untuk pengembangan *Java* android, seperti *android.app*, *android.content*, *android database*, *android.os*, *android opengl*, *android.text*.
6. (Mulyati & Wardono, 2019) **Kreatifitas matetatis siswa pada pembelajaran *Discovery Learning* dengan media berbasis android studio,** Meskipun android studio membutuhkan banyak memori, tetapi ini dapat di selesaikan dengan kelebihan yang dimiliki android studio, berikut fitur yang dimiliki oleh android studio. Instant run merupakan fitur yang

dimiliki oleh android studio yang dikompilasi dan menjalankan program membutuhkan waktu yang lama, tetapi tidak untuk yang kedua kali dan seterusnya karena proses selanjutnya akan lebih cepat. Code editor pintar merupakan fitur auto completion yang dimiliki oleh android studio yang menampilkan saran code yang ingin diketikkan dan IDE membuat para program dimanjakan dengan kemudahan dalam melakukan build, apabila kita melakukan kompilasi dan menjalankan aplikasi, karena APK yang sudah terbentuk secara otomatis sehingga para pembuat program tidak perlu repot lagi untuk build aplikasi kembali. Dapat membuat aplikasi untuk semua perangkat android, tidak hanya membuat aplikasi untuk perangkat *smartphone*. sebagai proses pembelajaran terjadi bila tidak disajikan dengan pelajaran yang berbentuk finalnya.

7. (William & Rinabi, 2018) **Rancang bangun aplikasi sistem pakar berbasis android untuk memprediksi kerusakan pada mesin sepeda motor yamaha R25,** ISSN: 2579-8790. Penelitian ini rancang menggunakan aplikasi berbasis android agar mempermudah pengguna motor yamaha R25 dalam mengidentifikasi masalah kerusakan pada mesin yang terjadi dan menjadikannya referensi bagi pengguna dan bengkel dalam mengambil keputusan setiap melakukan perbaikan sepeda motor agar lebih efisien dan hemat biaya.
8. (Riandy & Henry, 2017) **Perancangan dan pembuatan aplikasi pemeriharaan untuk memantau kondisi mesin pabrik berbasis android di PT. X ,** Vol 5. No 2 (2017). Penelitian yang terkait saat ini pencatatan

yang digunakan masih secara manual dalam bentuk dokumen fisik untuk melakukan proses perawatan mesin yang sedang dijalankan. Disini aplikasi yang digunakan ialah bahasa pemrograman *HTML, CSS, PHP, JavaScript, Java, database* dan *MySQL*. Hasil dari pengembangan aplikasi ini dapat terintegrasinya informasi perusahaan yaitu pencatatan pelaporan pemeriksaaan item serta data laporan yang telah diverifikasi berdasarkan waktu yang telah ditentukan. Dapat disimpulkan bahwa aplikasi ini dapat mempermudah kinerja maintenance untuk mengetahui status kondisi mesin saat ini.

9. (Muljanto, 2020) **Pencatatan dan pebukuan Via aplikasi akuntansi UMKM di Sidoarjo,** ISSN: 2477-6289. Penelitian ini membahas tentang permasalahan yang sering terjadi dalam pencatatan dan pembukuan hingga laporan keuangan. Hasil dari penelitian ini agar pelaku UMKM mau dan disiplin dalam menggunakan aplikasi UMKM yang mudah digunakan secara *mobile* dengan media *smartphone*.
10. (Cosmas, Joni, & Fergyanto, n.d.) **Perancangan sistem informasi penggajian berbasis web (studi kasus di Rumah Sakit St. Elisabeth),** ISSN: 2460-3465. Penelitian ini bertujuan untuk memberikan solusi pada sistem penggajian di Rumah Sakit St. Elisabeth yang saat ini belum menerapkan sistem yang terintegrasi dan masih menggunakan perhitungan secara manual. Masalah yang timbul pada sistem yang belum terintegrasi ialah pemborosan penyimpanan, perhitungan yang tidak akurat, dengan

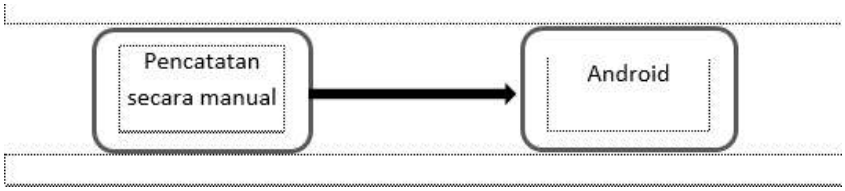
menggunakan sistem yang berbasis data dapat membantu penggajian yang lebih efektif dan efisien.

2.4 Kerangka Pemikiran

Metode kerangka pemikiran yaitu menjelaskan tentang alur atau rencana dari sebuah penelitian. Kerangka ini mempunyai konsep yang menggambarkan antaran variabel yang satu dengan lainnya.

Latar belakang yang di dapat , diidentifikasi beberapa permasalahan ialah kecerobohan bagi *maintenance* karena resiko pencatatan data masih menggunakan kertas secara manual yang bisa menyebabkan terjadinya kesalahan pada saat mengisih *record* data, karena belum efektifnya sistem pencatatan kerusakan dan perbaikan yang digunakan saat ini maka dibutuhkan sebuah aplikasi yang dapat membantu mempermudah kinerja *maintenance hypermart.*

Berikut ini merupakan bagan dari kerangka pemikiran:



Gambar 2.9 Kerangka pemikiran
Sumber : (Data penelitian, 2021)

Kerangka pemikiran diatas menggambarkan bahwa pada langkah pertama mengidentifikasi masalah yang terjadi dilapangan tentang pencatatan secara manual dengan melakukan interview langsung terhadap maintenance yang bersangkutan serta mencari dokumen pendukung seperti buku dan jurnal ilmiah yang keterkaitan dengan sistem pencatatan kerusakan pada mesin. Langkah kedua yakni aplikasi akan diterapkan dengan menggunakan sistem berbasis android (*smartphone*).