

**APLIKASI PENCATATAN KERUSAKAN
DAN PERBAIKAN MESIN PENDINGIN
BERBASIS ANDROID**

SKRIPSI



**Oleh
Liana Ramayani
170210107**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN KOMPUTER
UNIVERSITAS PUTERA BATAM
TAHUN 2021**

**APLIKASI PENCATATAN KERUSAKAN
DAN PERBAIKAN MESIN PENDINGIN
BERBASIS ANDROID**

SKRIPSI

**Untuk memenuhi salah satu syarat
guna memperoleh gelar Sarjana**



Oleh

Liana Ramayani

170210107

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN KOMPUTER
UNIVERSITAS PUTERA BATAM
TAHUN 2021**

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini saya:

Nama : Liana Ramayani
NPM : 170210107
Fakultas : Teknik Dan Komputer
Program Studi : Teknik Informatika

Menyatakan bahwa Skripsi yang saya buat dengan judul:

**“APLIKASI PENCATATAN KERUSAKAN DAN PERBAIKAN MESIN
PENDINGIN BERBASIS *ANDROID*”**

Adalah hasil karya sendiri dan bukan “duplikasi” dari karya orang lain. Sepengetahuan saya, di dalam naskah Skripsi ini tidak terdapat karya ilmiah atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip di dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia naskah Skripsi ini digugurkan dan Skripsi yang saya peroleh dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku.

Demikian pernyataan ini saya buat dengan sebenarnya tanpa ada paksaan dari siapapun

Batam, 11 Juli 2021



Liana Ramayani
170210107

**APLIKASI PENCATATAN KERUSAKAN
DAN PERBAIKAN MESIN PENDINGIN
BERBASIS ANDROID**

SKRIPSI

**Untuk memenuhi salah satu syarat
guna memperoleh gelar Sarjana**

Oleh

Liana Ramayani

170210107

**Telah disetujui oleh Pembimbing pada tanggal
seperti tertera di bawah ini**



Batam, 12 Juli 2021

Nia Ekawati, S.Kom., M.SI.

Pembimbing

ABSTRAK

Showcase Cooler merupakan mesin pendingin atau kulkas yang digunakan untuk mengawetkan minuman atau makanan agar bisa disimpan lebih lama. Dalam melakukan perbaikan mesin pendingin yang masih menggunakan kertas secara manual mengakibatkan sering terjadinya kesalahan atau selisi data dalam *merecord* data, penggunaan kertas yang sangat boros karena harus menyalin dan merekap data untuk didistribusikan ke bagian yang lain. Pada saat ini android merupakan teknologi yang sangat kompleks karena sudah menggunakan sumber terbuka sebagai basis proyek sehingga menjadikan sistem operasi *smartphone* paling banyak digunakan pada saat ini. Adapun dukungan dari perangkat lunak yang digunakan dalam mengembangkan aplikasi ini yaitu *Andoid Studio* yang merupakan *software develoment* untuk membangun atau mengembangkan aplikasi android, *Java* yang berfungsi sebagai bahasa pemrograman yang biasanya digunakan untuk mmebuat *software*, dan *SQLite* merupakan sebuah database yang menggunakan penyimpanan *internal device* pengguna. Dalam penelitian ini metode pengujian yang digunakan adalah *Black Box Testing* atau disebut dengan pengujian yang fungsional karena pengujian perangkat lunak yang digunakan tanpa harus mengetahui struktur internal kode atau program. Aplikasi yang dihasilkan yaitu sistem pencatatan kerusakan dan perbaikan berbasis android yang digunakan untuk *maintenance*. Pemanfaatan teknologi ini agar *maintenance* dapat mencatat kerusakan perbaikan pada mesin menggunakan *smartphone* yang berbasis android agar bisa menghemat media buku teks atau kertas.

Kata Kunci : Android, *Showcase cooler*, pencatatan, SQLite, Android studio

ABSTRACT

Showcase Cooler is a cooling machine or refrigerator that is used to preserve drinks or food so that they can be stored longer. In repairing the cooling machine that still uses paper manually, it often results in errors or data discrepancies in recording data, the use of paper is very wasteful because they have to copy and recap data to be distributed to other parts. At this time android is a very complex technology because it uses open source as the project base so that it makes the smartphone operating system the most widely used at this time. The support for the software used in developing this application is Android Studio which is a development software for building or developing android applications, Java which functions as a programming language that is usually used to create software, and SQLite is a database that uses the user's internal storage device. In this study the testing method used is Black Box Testing or called functional testing because testing the software used without having to know the internal structure of the code or program. The resulting application is an android-based damage and repair recording system that is used for maintenance. Utilization of this technology so that maintenance can record repair damage to the machine using an Android-based smartphone in order to save on textbooks or paper media.

Keywords: Android, Android studio, Recording, Showcase cooler, SQLite,

KATA PENGANTAR

Puji Syukur Tercurahkan Kepada Allah SWT yang telah melimpahkan segala rahmat dan karuniaNya, sehingga penulis dapat menyelesaikan laporan tugas akhir yang merupakan salah satu persyaratan untuk menyelesaikan program studi strata satu (S1) pada Program Studi Teknik Informatika Universitas Putera Batam.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Karena itu, kritik dan saran akan senantiasa penulis terima dengan senang hati. Dengan segala keterbatasan, penulis menyadari pula bahwa skripsi ini takkan terwujud tanpa bantuan, bimbingan, dan dorongan dari berbagai pihak. Untuk itu, dengan segala kerendahan hati, penulis menyampaikan ucapan terima kasih kepada:

1. Rektor Universitas Putera Batam.
2. Ketua Program Studi Teknik Informatika
3. Nia Ekawati, S.Kom., M.SI. selaku pembimbing Skripsi pada Program Studi Teknik Informatika Universitas Putera Batam.
4. Dosen dan Staff Universitas Putera Batam
5. Kepada kedua Orang Tua dan keluarga tercinta
6. Bpk. Syafrial Syafri selaku store manager Hypermart Mega Mall Batam Center Kepulauan Riau dan Bpk. Agung Baramuli selaku Maintenance.
7. Ucapan terima kasih kepada teman-teman, Pitri Dayenti, Rosni aritonang, Alvin Rendi, Elis Haryati, Joel Sihombing, Bernad Siahaan, Eeng Nofia Yengsih, Serta rekan seperjuangan satu angkatan sekalian yang telah membantu dan mendukung pada penelitian ini.

Semoga Allah SWT. membalas kebaikan dan selalu mencurahkan hidayah serta taufik Nya, Amin.

A handwritten signature in black ink, appearing to be 'Liana Ramayani', written in a cursive style with a horizontal line underneath.

Batam, 15 Juli 2021

Liana Ramayani

DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN JUDUL	ii
SURAT PERNYATAAN	iii
HALAMAN PENGESAHAN	iv
ABSTRAK	vi
ABSTRACT	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	4
1.3 Pembatasan Masalah	5
1.4 Rumusan masalah.....	5
1.5 Tujuan Penelitian	5
1.6 Manfaat Penelitian	6
1.6.1 Manfaat teoritis	6
1.6.2 Manfaat praktis	6
BAB II KAJIAN PUSTAKA	7
2.1 Teori Dasar	7
2.1.1 <i>Android</i>	7
2.1.2 <i>Unified Modeling Language (UML)</i>	8
2.1.3 Pengujian Aplikasi	16
2.2 Teori Khusus.....	17
2.2.1 Mesin Pendingin	18
2.2.2 Pencatatan.....	19
2.2.3 Software Pendukung	20
2.2.3.1 <i>Android Studio</i>	20
2.2.3.2 <i>Java</i>	21
2.2.3.3 <i>SQLite</i>	23
2.3 Penelitian Terdahulu	25
2.4 Kerangka Pemikiran.....	30
BAB III METODE PENELITIAN	32
3.1 Desain Penelitian.....	32
3.2 Pengumpulan Data	34
3.3 Metode Perancangan Sistem.....	35
3.3.1 <i>Unified Modeling Language (UML)</i>	36

3.3.2	Desain Rancangan (<i>Story Board</i>).....	45
3.4	Lokasi dan Jadwal Penelitian.....	50
BAB IV HASIL DAN PEMBAHASAN		51
4.1	Hasil Penelitian	51
4.2	Pembahasan	54
4.1.1	Pengujian	55
4.3	Implementasi.....	57
BAB V SIMPULAN DAN SARAN.....		58
5.1	Simpulan.....	58
5.2	Saran.....	58
DAFTAR PUSTAKA.....		59
Lampiran 1 Pendukung Penelitian		62
Lampiran 2 Daftar Riwayat Hidup.....		67
Lampiran 3 Surat Keterangan Penelitian.....		68

DAFTAR GAMBAR

Gambar 2.1	Logo android	8
Gambar 2.2	Contoh <i>usecase diagram</i>	11
Gambar 2.3	Contoh <i>Activity diagram</i>	13
Gambar 2.4	Contoh <i>Sequence diagram</i>	14
Gambar 2.5	Contoh <i>Class diagram</i>	16
Gambar 2.6	Logo Android Studio.....	20
Gambar 2.7	Logo <i>Java</i>	23
Gambar 2.8	Logo <i>SQLite</i>	24
Gambar 2.9	Kerangka pemikiran	30
Gambar 3.1	Desain penelitian.....	31
Gambar 3.2	<i>Usecase diagram</i>	35
Gambar 3.3	<i>Activity diagram add machine</i>	36
Gambar 3.4	<i>Activity diagram scan barcode</i>	37
Gambar 3.5	<i>Activity diagram menu search engine</i>	38
Gambar 3.6	<i>Activity diagram menu sort list</i>	39
Gambar 3.7	<i>Activity diagram menu print</i>	40
Gambar 3.8	<i>Sequence diagram menu add machine</i>	41
Gambar 3.9	<i>Sequence diagram menu scan barcode</i>	42
Gambar 3.10	<i>Sequence diagram menu print</i>	42
Gambar 3.11	<i>Class diagram</i>	43
Gambar 3.12	Rancangan aplikasi menu utama.....	44
Gambar 3.13	Rancangan aplikasi menu <i>add machine</i>	45
Gambar 3.14	Rancangan aplikasi menu <i>scan barcode</i>	46
Gambar 3.15	Rancangan aplikasi menu <i>sort list</i>	46
Gambar 3.16	Rancangan aplikasi menu <i>info machine</i>	47
Gambar 3.17	Menu <i>maintenance form change</i>	47
Gambar 3.18	Menu <i>maintenance form repair</i>	48
Gambar 3.19	Menu <i>maintenance form another</i>	48
Gambar 4.1	Layar tampilan pembuka aplikasi	50
Gambar 4.2	Layar tampilan menu utama <i>add machine</i>	51
Gambar 4.3	Layar tampilan menu utama <i>scan barcode</i>	51
Gambar 4.4	Layar tampilan menu utama <i>List machine</i>	52
Gambar 4.5	Layar tampilan menu utama <i>info machine</i>	52
Gambar 4.6	Proses awal pada <i>add machine</i>	54
Gambar 4.7	Prose menginput <i>name machine</i> dan <i>serial number</i>	55
Gambar 4.8	proses mengisi <i>maintenance form</i>	55
Gambar 4.9	Proses lihat <i>info machine</i>	56

DAFTAR TABEL

Tabel 2.1 Simbol-simbol <i>Use case diagram</i>	10
Tabel 2.2 Simbol-simbol <i>Activity diagram</i>	11
Tabel 2.3 Simbol-simbol <i>Sequence diagram</i>	13
Tabel 2.4 Simbol-simbol <i>Class diagram</i>	15
Tabel 3.1 Jadwal penelitian	49
Tabel 4.1 Pengujian fungsional menu	53

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini mesin pendingin mempunyai aspek yang banyak di Indonesia . Salah satu pengaplikasian mesin pendingin ini digunakan khususnya pada bidang usaha yang menjual berbagai jenis minuman. *Showcase cooler* atau mesin pendingin merupakan produk kulkas atau lemari pendingin yang menggunakan media kaca untuk memajanh minuman atau makanan yang ingin ditampilkan dan membuat produk bisa tetap dilihat dari luar. Biasanya Lemari pendingin seperti ini sering kita jumpai di rumah makan, restoran, dan warung makan. Lemari pendingin *showcase* mempunyai kaca seperti etalase dan memiliki perbedaan pada bentuk yang bervariasi. Suhu *showcase* ini juga disesuaikan dengan penempatan produk tertentu, sehingga kualitas minuman dan makanan yang ada didalam lemari pendingin lebih tahan lama tanpa menggunakan bahan pengawet. Kulkas minuman pendingin ini dengan tipe model kaca cembung dan tentunya berbeda dengan mesin pendingin untuk penyimpan es krim yang biasa ditemukan di beberapa toko klontong.

Berdasarkan penelitian (Kusbandono & Purwadi, 2016) mesin pendingin atau *showcase* memiliki fungsi yang pada umumnya di gunakan untuk mendinginkan minuman kemasan sehingga orang yang meminum minuman tersebut mendapatkan kesegaran, selain itu juga ada mesin *showcase* yang di gunakan untuk mengawetkan bahan makanan. Di dalam mesin pendingin atau *showcase* hanya mampu mendinginkan minuman kemasan yang siap di minum,

karena itu minuman tidak boleh dalam keadaan beku, karena minuman yang beku akan sulit di minum. Evaporator berfungsi untuk mengatur udara dingin yang ada di dalam *showcase*, suhu di dalam mesin pendingin atau *showcase* pada umumnya di rancang tidak boleh rendah dari 0°C dan tidak boleh lebih dari 12°C. *Showcase* memang di rancang agar minuman yang di dinginkan di dalam mesin pendingin bisa terlihat dari luar itu bertujuan agar pintu *showcase* tidak sering kali di buka karena jika mesin *showcase* tidak sering di buka, maka beban *showcase* tidak sering terganggu. Jika pintu sering di buka beban pendinginan menjadi lebih besar karena udara luar yang dengan mudah dapat masuk ke ruang pendingin minuman, hal ini yang akan membuat beban listrik menjadi besar.

Berdasarkan wawancara yang telah dilakukan dengan narasumber, saat ini *maintenance hypermart* masih melakukan pencatatan data setiap kali ada kerusakan dan juga perbaikan pada *showcase* atau mesin pendingin yang masih menggunakan kertas secara manual. Hal ini dapat mengakibatkan kesalahan yang di sebabkan oleh *maintenance* itu sendiri, karena tidak efektifnya penggunaan kertas, sering terjadi masalah seperti lupa mengisi *record* data didalam *form service report*, tidak terdata pada pelaporan dalam satu bulan, atau kertas *form* yang di gunakan hilang. Menurut data yang di berikan oleh narasumber sistem pencatatan data kerusakan dan perbaikan pada mesin pendingin hanya mencatat hari, tanggal, dan keterangan kerusakan yang terjadi. Maka dari itu di butuhkan sebuah aplikasi yang memudahkan *maintenance* di *hypermart*.

Berdasarkan penelitian (Saepudin, Rahayu & Kencanawaty, 2020) pencatatan yang di lakukan di PT. Haeng Nam masih di lakukan secara manual

menggunakan kertas, dampak dari penyajian informasi dan analisis kerusakan sampai harus menunggu 4 hari kerja dan dilakukan oleh 2 orang admin untuk menyelesaikannya, selain itu masalah yang sering terjadi pada saat pencatatan secara manual adalah sering terjadinya kesalahan atau selisih data, penggunaan kertas yang sangat boros, karena harus merekap data beberapa salinan untuk didistribusikan ke bagian yang lain. Dari permasalahan tersebut, penelitian ini dibuat bertujuan untuk mempermudah proses pengolahan data dalam penyajian informasi dan kerusakan dengan menggunakan aplikasi yang sudah berbasis *web* agar pencatatan data lebih akurat dan cepat, selain itu juga data bisa tersimpan dengan baik dan aman di dalam *database*.

Android resmi dirilis pada tahun 2007, dan saat ini masuk ke dalam teknologi yang sangat kompleks. Android memungkinkan produsen perangkat, operator nirkabel, dan pengembang aplikasi untuk dengan bebas memodifikasi dan mendistribusikan perangkat lunak. Android memiliki komunitas pengembang aplikasi yang besar, biasanya menggunakan versi bahasa pemrograman yang disesuaikan untuk memperluas fungsionalitas perangkat. Keterbukaan android juga mendorong untuk menggunakan *open source* sebagai basis proyek pengembangan aplikasi dengan menambahkan fitur untuk menjalankan android pada perangkat yang dirilis resmi menggunakan sistem operasi lain, dan *smartphone* mempunyai banyak peminat.

Berdasarkan penelitian (Siddik & Nasution, 2018) Android adalah perangkat lunak yang berjalan sesuai dengan sistem operasi. Android (SDK) *Standart Development Kit* menyediakan alat dan aplikasi antar muka pemrograman untuk

memulai pengembangan diperlukan aplikasi, dengan kata lain, gunakan data dan sumber file yang dibutuhkan aplikasi dan digabungkan dengan alat aplikasi yang menjadi paket android.

Untuk mengatasi masalah tersebut, pada penelitian ini di rancang sebuah aplikasi untuk menghemat penggunaan media kertas dan mempermudah kinerja *maintenance*. Berdasarkan uraian tersebut maka penelitian ini membahas tentang **“APLIKASI PENCATATAN KERUSAKAN DAN PERBAIKAN MESIN PENDINGIN BERBASIS ANDROID”**

1.2 Identifikasi Masalah

Adapun beberapa identifikasi masalah yang dapat diuraikan yaitu:

1. Saat ini *maintenance hypermart* masih melakukan pencatatan data setiap kali ada kerusakan dan juga perbaikan pada *showcase* atau mesin pendingin menggunakan kertas secara manual.
2. Dapat mengakibatkan beberapa kesalahan yang disebabkan oleh *maintenance* itu sendiri, karena tidak efektifnya penggunaan kertas.
3. Sering terjadi masalah seperti lupa mengisi *record* data didalam *form service report*.
4. Pelaporan data dalam satu bulan tidak terdata atau kertas *form* yang digunakan hilang.

1.3 Pembatasan Masalah

Adapun batasan masalah yang telah ditetapkan penelitian sebagai berikut :

1. Jenis mesin pendingin yang di teliti yaitu *showcase*.
2. Data di ambil dari *hypermart* mega mall Batam.
3. Aplikasi yang di rancang dalam penelitian ini menggunakan sistem operasi berbasis *Android*.
4. Wawancara pada penelitian ini di tujukan kepada *maintenance hypermart* yaitu Bapak Arif.
5. Software pendukung yang digunakan dalam pembuatan aplikasi adalah *Android Studio, Java, dan SQLite*.
6. Pengujian dalam aplikasi ini menggunakan *black box testing*

1.4 Rumusan masalah

Berdasarkan identifikasi masalah, maka rumusan masalah yang didapat, yaitu: Bagaimana aplikasi ini mempermudah dalam pengecekan *record* data pada mesin ?

1.5 Tujuan Penelitian

Tujuan penelitian yang di harapkan oleh penulis dalam melakukan kegiatan, adalah : Untuk menghemat penggunaan media buku teks atau kertas untuk mencatat data hasil kerusakan dan perbaikan pada mesin serta mempermudah kinerja *maintenance*.

1.6 Manfaat Penelitian

Manfaat penelitian dibagi menjadi dua yaitu manfaat teoritis dan manfaat praktis, sebagai berikut:

1.6.1 Manfaat teoritis

Adapun manfaat teoritis yang didapatkan adalah:

1. Dapat digunakan sebagai pelengkap untuk mempermudah sistem kerja *maintenance*.
2. Hasil penelitian ini dapat dijadikan acuan sebagai pengembangan aplikasi pencatatan kerusakan dan perbaikan mesin pendingin berbasis *Android*.

1.6.2 Manfaat praktis

1. Bagi pembaca

Dapat menambah ilmu pengetahuan dan pengalaman cara membuat aplikasi *image machine* berbasis *Android*.

2. Bagi penulis

Agar kedepannya penulis mampu menerapkan dan meningkatkan lebih banyak fitur pada aplikasi serta menerapkan sedikit pengalaman cara membuat aplikasi *image machine* berbasis *Android*.

3. Bagi Universitas Putera Batam

Menambah ilmu pengetahuan dapat memberikan rujukan serta sumbangan pemikiran bagi Universitas.

BAB II

KAJIAN PUSTAKA

2.1 Teori Dasar

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

2.1.1 Android

Android mempunyai biaya yang tidak mahal, sehingga menjadi pilihan bagi perusahaan teknologi, Android 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 Language (UML)*

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

Unified Modeling Language (UML) merupakan alat untuk mengembangkan sistem informasi dalam menggambarkan, merancang, dan mendokumentasikan sistem perangkat lunak. Perancangan 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 *Sequence Diagram*.


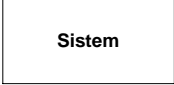



1. *Use case diagram*


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)

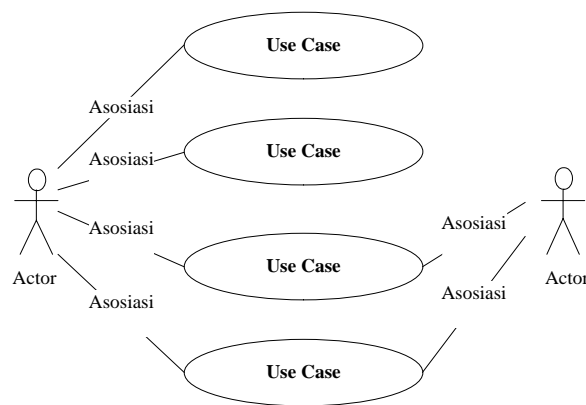
Tabel 2.1 Simbol-simbol *Use case diagram*

Simbol	Keterangan
	<p><i>Use Case</i></p> <ul style="list-style-type: none"> ▪ Sebagian besar yang mewakili sistem fungsional ▪ Batas sistem yang ditempatkan didalam ▪ Penamaan <i>usecase</i> yaitu label atau kata kerja dan juga diikuti kata benda
	<p><i>System Boundary</i></p> <ul style="list-style-type: none"> ▪ Nama pada <i>sistem boundary</i> terdapat dibagian atas ▪ Digambarkan dengan ruang lingkup sistem
	<p>Asosiasi Boundary</p> <ul style="list-style-type: none"> ▪ Menghubungkan antara usecase dengan aktor yang saling berinteraksi
<p><<include>></p>	<p><i>Include</i></p> <ul style="list-style-type: none"> ▪ Arah panah mengarah kepada <i>main use case</i>.
	<ul style="list-style-type: none"> ▪ Relasi menggambarkan sebuah use case (<i>sub use case</i>) fungsinya aka dijalankan terlebih dahulu.
<p><<extend>></p>	<p>EXTEND</p> <ul style="list-style-type: none"> ▪ Relasi yang menggambarkan sebuah use case berdiri sendiri tanpa main use case dijalankan terlebih dahulu.
	<ul style="list-style-type: none"> ▪ Menggambarkan relasi main <i>use case</i> yang bisa berdiri sendiri tanpa dijalankan terlebih dahulu.

	<p>GENERALISASI / <i>GENERALIZATION</i></p> <ul style="list-style-type: none"> ▪ 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*

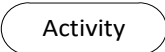
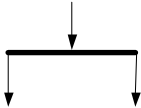
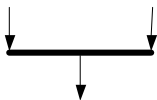
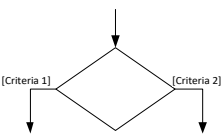

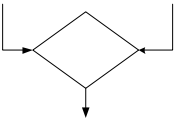
Activity diagram merupakan aksi yang proses kerja atau aktivitas dari sebuah sistem. *Activity Diagram* dibuat berdasarkan alur *Use case Diagram* agar dapat mempermudah dan memahami alur proses sistem.

Tabel 2.2 Simbol-simbol *Activity diagram*

Simbol	Keterangan
	<p><i>Start poin</i></p> <ul style="list-style-type: none"> ▪ Awal penelusuran ▪ <i>Star poin</i> untuk memulai sebuah aktivitas ▪ Hanya boleh digunakan 1 simbol pada sebuah

	aktivitas.
	<p><i>End Poin</i></p> <ul style="list-style-type: none"> ▪ Akhir penelusuran ▪ <i>End Poin</i> akhir aktivitas diagram ▪ >1 simbol <i>End Poin</i> hanya boleh digunakan pada sebuah aktivitas.

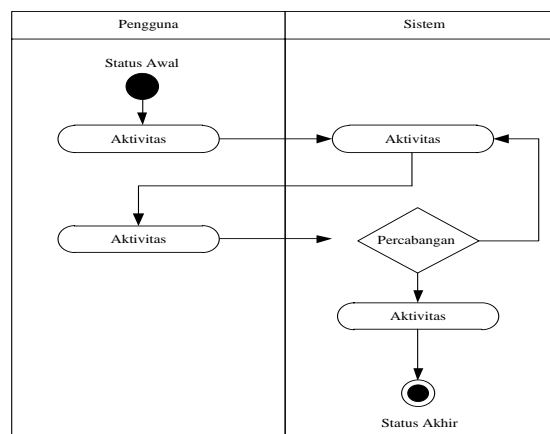
Tabel 2.2 (Lanjutan)

	<p><i>Activities</i></p> <ul style="list-style-type: none"> ▪ Akhir penelusuran ▪ Menggambarkan tentang aktivitas dari kata kerja ▪ Hanya memiliki sebuah aktivitas yaitu satu alur masuk dan satu alur keluar
	<p><i>Fork</i></p> <ul style="list-style-type: none"> ▪ Percabangan ▪ Satu aliran yang dikerjakan secara bersamaan
	<p><i>Join</i></p> <ul style="list-style-type: none"> ▪ Penggabungan ▪ Aliran yang disatukan untuk melanjutkan aktivitas
	<p><i>Decision poin</i></p> <ul style="list-style-type: none"> ▪ Pada tengah belah ketupat tida ada keterangan (pertanyaan) <i>guards</i> harus dimiliki oleh <i>flowchart</i>
	<p><i>Guarid</i></p> <ul style="list-style-type: none"> ▪ Sebuah transisi ketika dilewati sebuah kondisi benar
	<p><i>Merge</i></p> <ul style="list-style-type: none"> ▪ Jalur keputusan akan kembali dan melewati <i>decision poin</i>

<i>Swimlane</i>	<p><i>Swimlane</i></p> <ul style="list-style-type: none"> ▪ Aktor yang didasarkan dan dikelompokan sebuah cara aktivitas ▪ Actor bisa ditulis nama actor ▪ Digambarkan secara horizontal dan vertikal dari <i>swimlane</i>
-----------------	---

Sumber : (Maharani, 2018)

Penerapan *Activity diagram* seperti gambar di bawah ini:



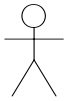


Gambar 2.3 Contoh *Activity diagram*

Sumber : (Data Penelitian, 2021)



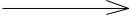
2. *Sequence diagram*

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

Tabel 2.3 Simbol-simbol *Sequence diagram*

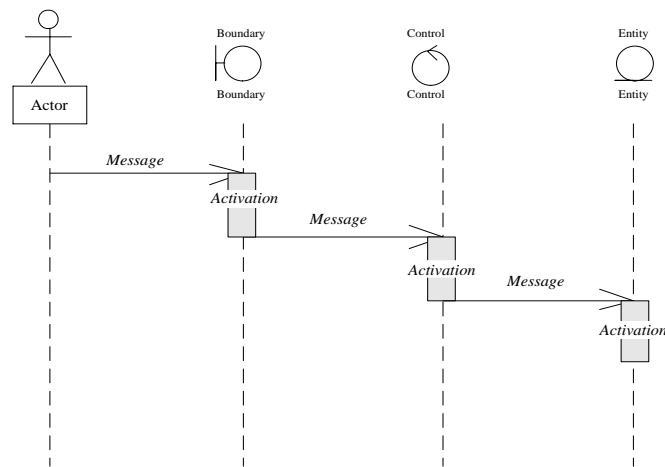
Simbol	Keterangan
	<p><i>Actor</i></p> <ul style="list-style-type: none"> Seorang yang berinteraksi dengan sistem
	<p><i>Boundary</i></p> <ul style="list-style-type: none"> Penghubung antara aktor dengan sistem
	<p><i>Control</i></p> <ul style="list-style-type: none"> Prilaku suatu sistem yang telah diatur aur kerja nya oleh suatu sistem

Tabel 2.3 (Lanjutan)

	<p><i>Entity</i></p> <ul style="list-style-type: none"> Suatu sistem yang menyimpan sebuah informasi Suatu sistem oleh struktru data yang digambarkan <i>entity</i>.
	<p><i>Activation</i></p> <ul style="list-style-type: none"> Suatu objek yang digambarkan kondisi interaksi Sebuah operasi oleh durasi aktif dengan berbanding lurus dengan panjang simbol.
	<p><i>Message</i></p> <ul style="list-style-type: none"> 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)




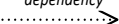

3. *Class diagram*

Class diagram menggambarkan tentang struktur dan pendefinisian kelas, *package* dan objek hubungan kelas satu dengan kelas yang lain. *Class Diagram* terdiri dari tiga area yaitu, Nama, Atribut (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:

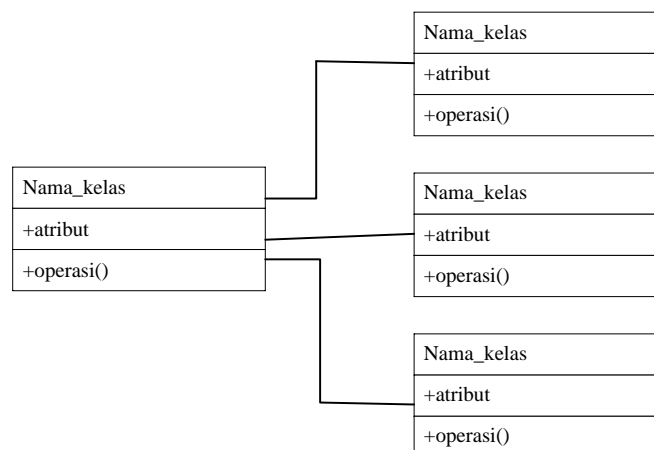
Tabel 2.4 Simbol-simbol *Class diagram*

Simbol	Deskripsi
<div style="border: 1px solid black; padding: 5px; width: fit-content;"> Nama_kelas +atribut +operasi() </div>	<ul style="list-style-type: none"> Struktur sistem yang terdapat pada kelas
Antarmuka / <i>interface</i>  Nama_interface	<ul style="list-style-type: none"> Pemrograman berorientasi objek pada konsep <i>interface</i>

Asosiasi/ <i>association</i> 	<ul style="list-style-type: none"> Makna umum dengan relasi antar kelas, asosiasi biasanya dilengkapi dengan <i>multiplicity</i>
Asosiasi berarah / <i>directed association</i> 	<ul style="list-style-type: none"> Satu kelas dengan kelas yang lain bermakna relasi antar kelas, <i>multiplicity</i> biasanya juga terdapat pada asosiasi
generalisasi 	<ul style="list-style-type: none"> Generalisasi-spesialisasi (umum khusus) bermakna dengan relasi antar kelas
Kebergantungan/ <i>dependency</i> 	<ul style="list-style-type: none"> Antar kelas tergantung pada makna dengan relasi antar kelas
Agregasi/ <i>aggregation</i> 	<ul style="list-style-type: none"> Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

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*) kesalahan 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 permasalahan 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 *Android 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 terintegrasi resmi untuk sistem operasi android untuk disematkan untuk pengembangan android yang dibangun pada perangkat lunak *Jet Brains'IntelliJ 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 lanjutan 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 mencakup 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 bahasa 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 (*General Public License*) yang menjadikan menulis dan mengedit aplikasi secara *opensource* serta didukung dengan dua kemampuan handal yakni disematkannya *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 disematkannya *Java Virtual 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 obesity with machine learning and electronic health record data**, 150 (2021) 104454. Penelitian ini memiliki tujuan yaitu membandingkan 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 dalam proses perawatan pelaksanaan pengobatan dan akhirnya kontrol hipertensi.

5. (Ejjiyi et al., 2021) **Design and development 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 multitreading 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 matematis 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 pemeriksaan 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

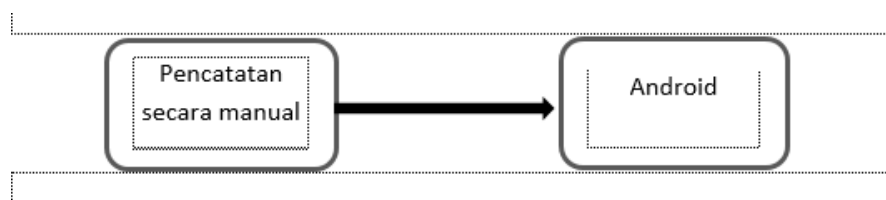
menggunakan sistem yang berbasis data dapat membantu penggajian yang lebih efektif dan efisien.

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)

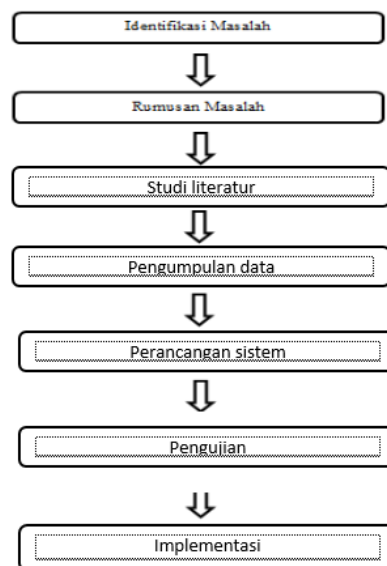
Kerangka pemikiran diatas menggambarkan bahwa pada langkah pertama mengidentifikasi masalah yang terjadi dilapangan tentang pencatatan secara manual dengan melakukan interview langsung terhadap maintenance yang bersangkutan serta mencari dokumen pendukung seperti buku dan jurnal ilmiah yang keterkaitan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

BAB III

METODE PENELITIAN

3.1 Desain Penelitian

Dalam desain penelitian ini dirancang sebuah tampilan aplikasi yang dibuat, Aplikasi dirancang di sesuaikan dengan kebutuhan pengguna maka dari itu dibuat beberapa langkah untuk merancang dimulai dari tahapan.



Gambar 3.1 Desain penelitian
Sumber : (Data penelitian, 2021)

1. Identifikasi masalah

Saat ini *maintenance hypermart* masih melakukan pencatatan data setiap kali ada kerusakan dan juga perbaikan pada *showcase* atau mesin pendingin dengan kertas secara manual. Dapat mengakibatkan kesalahan yang disebabkan oleh *maintenance* itu sendiri, karena tidak efektifnya penggunaan kertas. Sering

terjadi masalah seperti lupa mengisi *record* data didalam *form service report*. Data yang dimasukkan tidak *valid* atau *form service report* nya hilang.

2. Rumusan masalah

Berdasarkan identifikasi masalah yang telah di bahas, maka rumusan masalah yang dapat di ambil, yaitu: Bagaimana aplikasi ini mempermudah dalam pengecekan *record* data pada mesin.

3. Studi literatur

Studi literatur memiliki komponen pendukung seperti referensi dari Buku, Ebook, Jurnal.

4. Pengumpulan Data

Proses dalam pengumpulan data melalui wawancara, wawancara dilakukan agar didapatkan data secara akurat dan lebih spesifik. Wawancara dilakukan melalui chat *via WhatsApp* dengan salah satu *maintenance* yaitu Bpk. Arif yang bekerja di *Hypermart Mega Mall*.

5. Perancangan aplikasi

Perancangan aplikasi ini disesuaikan dengan kebutuhan maka dari itu di perlukan bebrapa perangkat lunak pendukung , seperti android studio yang berperan untuk membangun atau mengembangkan aplikasi android, dibutuhkan juga *Java* untuk membuat sebuah *software*, dan *SQLite* digunakan sebagai sebuah *database* yang menggunakan penyimpanan internal *device* pengguna.

6. Pengujian

Aplikasi yang sudah dirancang dan dihasilkan harus melalui uji coba agar mengetahui letak kesiapan dari sebuah aplikasi yang dirancang. Tahapan pengujian dilakukan guna untuk melihat kinerja dari aplikasi yang telah dirancang apakah sudah sesuai dengan yang diinginkan, sehingga bisa dilakukan perbaikan dalam desain atau kode program dalam aplikasi.

7. Implementasi

Setelah proses perancangan, maka selanjutnya untuk melakukan implementasi. Proses ini membuat pengguna agar terbiasa dengan sistem yang baru digunakan. Implementasi dilakukan agar aplikasi dapat digunakan terutama untuk *maintenance* yang ada di *Hypermart Mega Mall*.

3.2 Pengumpulan Data

Ada dua metode yang digunakan untuk pengumpulan data yaitu studi pustaka dan wawancara:

1. Studi Pustaka

Studi Pustaka mempunyai sumber data yang diperoleh dalam penelitian untuk mendukung rancangan aplikasi serta memberikan gambaran.

a. Buku referensi

Buku yang digunakan terkait yaitu tentang *Java*, *android studio*, *android*, dan juga buku tentang *SQLite* (Firly, 2018)(Eko, 2019)(Yudho, 2019)(Sidik, 2020)(Hadiprakoso, 2021b)

a. Jurnal Ilmiah

Jurnal ilmiah yang digunakan yang memiliki E-ISSN atau ISSN serta terindex oleh Google Scholar, dan SINTA.(Hadiprakoso, 2021b)(Maiyana, 2018)(Ejjiy et al., 2021)(Siddik & Nasution, 2018)(Saepudin et al., 2020)(Kusbandono & Purwadi, 2016)(Sutrisno & Karnadi, 2021)(Susilo, 2018)(Soewito et al., 2019)(Mustaqbal et al., 2015)(Mulyati & Wardono, 2019)(Hutabri & Putri, 2019)(Febrianto & Ikbali, 2020)(Soewito et al., 2019)(Muljanto, 2020)

2. Wawancara

Wawancara yang dilakukan ada dua tahapan, yang pertama wawancara dengan *maintenance* yaitu kepada Bpk. Arif selaku *maintenance* yang bekerja di *Hypermart* Mega Mall. Dari hasil wawancara yang diperoleh yaitu bagaimana proses pencatatan perbaikan dan kerusakan yang dilakukan di *Hypermart*.

Wawancara yang kedua yaitu bagaimana mengimplementasikan aplikasi tersebut untuk menunjang dan juga mempermudah sistem kerja yang dilakukan oleh *maintenance*.

3.3 Metode Perancangan Sistem

Perancangan yang dilakukan pada penelitian ini menggunakan *Unified Modeling Language (UML)* yang termasuk seperti Usecase diagram, Activity diagram, Sequence diagram, dan juga class diagram.

3.3.1 Unified Modeling Language (UML)

Unified Modeling Language (UML) memiliki sebuah hubungan yang erat dalam menggunakan dan merancang perangkat lunak, pada aplikasi ini pengguna dapat mengakses fitur yang ada didalam aplikasi seperti menambah, mengedit, menghapus, menyimpan dan juga mencetak.

1. Use case



Gambar 3.2 Use case diagram

Sumber : (Data penelitian, 2021)

Use case diatas adalah cara bagaimana pengguna mengakses aplikasi.

Berikut merupakan keterangan *use case* diatas:

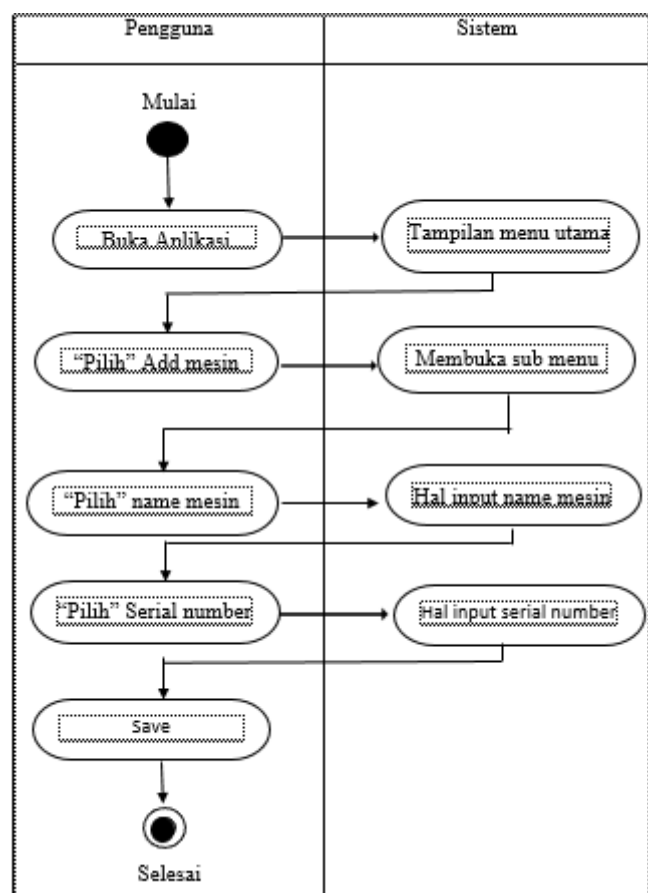
- 1) *Add mesin* ketika diklik maka akan muncul *add image*, *name machine*, *serial number* dan *save*. Pada *add image* maka secara otomatis akan muncul *camera* dan *directory* yang berfungsi untuk mengambil gambar mesin dan juga menginput gambar secara manual.

- 2) *Scan barcode* mesin akan membuka kamera setelah itu mendeteksi *barcode* pada mesin.
- 3) Menampilkan data *record* mesin maka akan muncul menu *search engine*, *sort list*, dan *print*. Menu tersebut berfungsi untuk mencari nomor mesin menampilkan list untuk mencari data mesin dan juga *mendownload* serta mencetak berkas.

2. Activity Diagram

A. menu *Add* mesin

Berikut merupakan gambaran alur menu *Add* mesin:



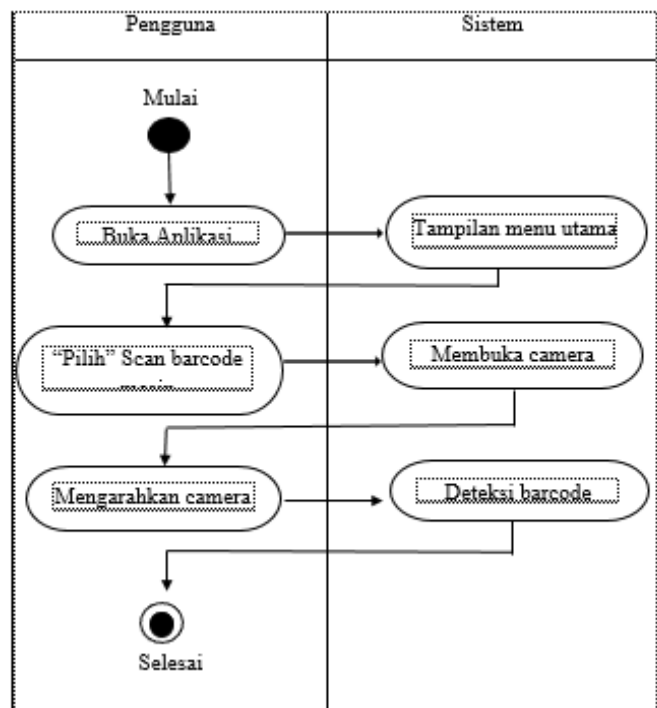
Gambar 3.3 Activity diagram Add Mesin

Sumber : (Data penelitian, 2021)

- 1) Saat aplikasi dibuka maka layar akan menampilkan menu utama.
- 2) Aplikasi akan terbuka dan pengguna menekan tombol *Add* mesin untuk memilih maka akan terbuka sub menu.
- 3) Pengguna menekan tombol *Add image* maka kamera akan terbuka.
- 4) Pengguna membuka *name* mesin maka akan ditampilkan halaman input nama mesin.
- 5) Pengguna menekan tombol *serial number* maka halaman input *serial number* akan terbuka.
- 6) *Save*
- 7) *Selesai*

B. Menu *Scan barcode*

Berikut merupakan gambaran alur *activity diagram* menu *Scan barcode*:



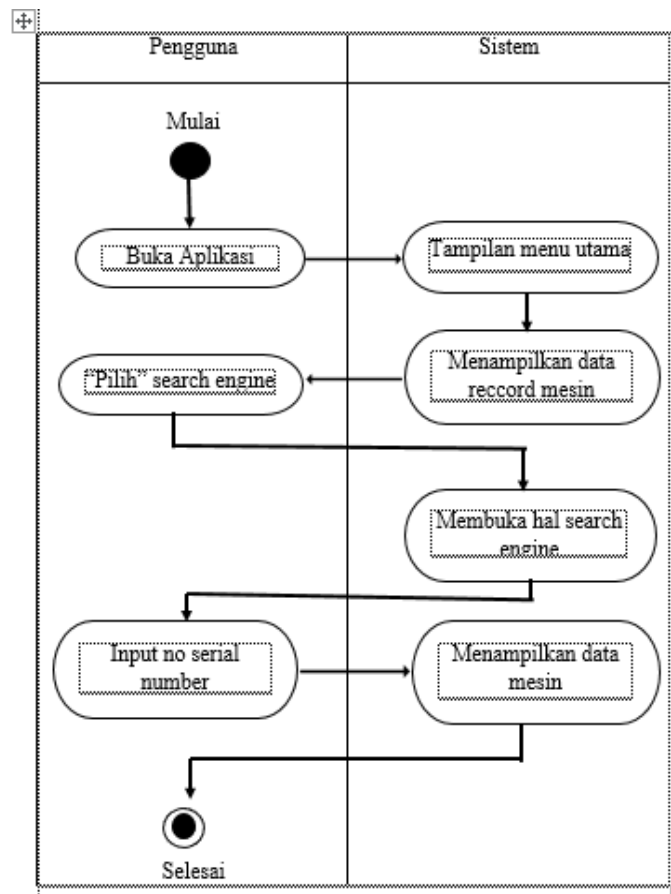
Gambar 3.4 Activity diagram Scan Barcode

Sumber : (Data penelitian, 2021)

- 1) Saat aplikasi terbuka maka akan ditampilkan menu utama.
- 2) Saat pengguna menekan tombol *Scan barcode* maka kamera akan terbuka.
- 3) Kamera akan diarahkan untuk mendeteksi *barcode*.
- 4) Selesai

C. Menu *Search engine*

Berikut merupakan gambaran alur *activity diagram* menu *Search engine*:



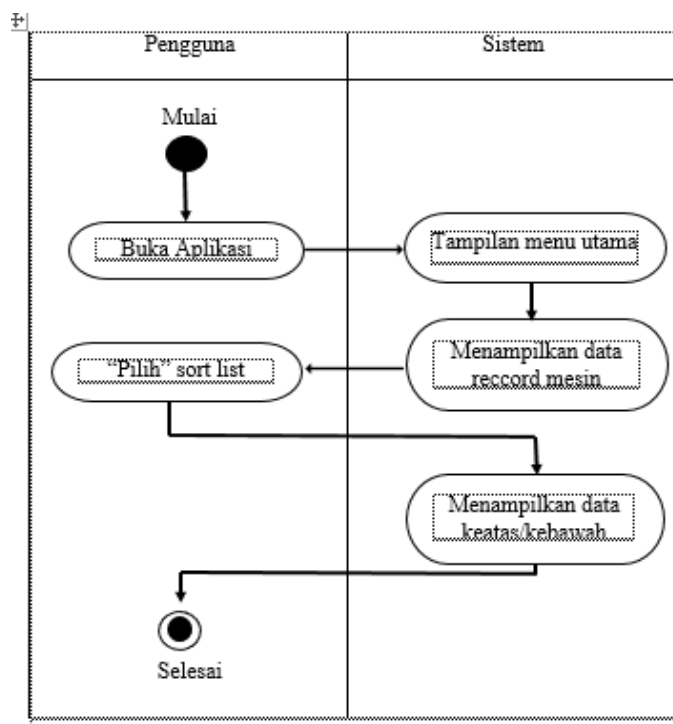
Gambar 3.5 *Activity diagram* menu Search Engine

Sumber : (Data penelitian, 2021)

- 1) Saat aplikasi dibuka maka tampilan menu utama akan ditampilkan dan langsung menampilkan data *record* mesin.
- 2) Pengguna menekan tombol *search engine* maka akan membuka halaman *search engine*.
- 3) Pengguna menekan tombol input *serial number* maka akan menampilkan data mesin.
- 4) Selesai

D. Menu *Sort list*

Berikut merupakan gambaran alur *activity diagram* menu *Sort list*:



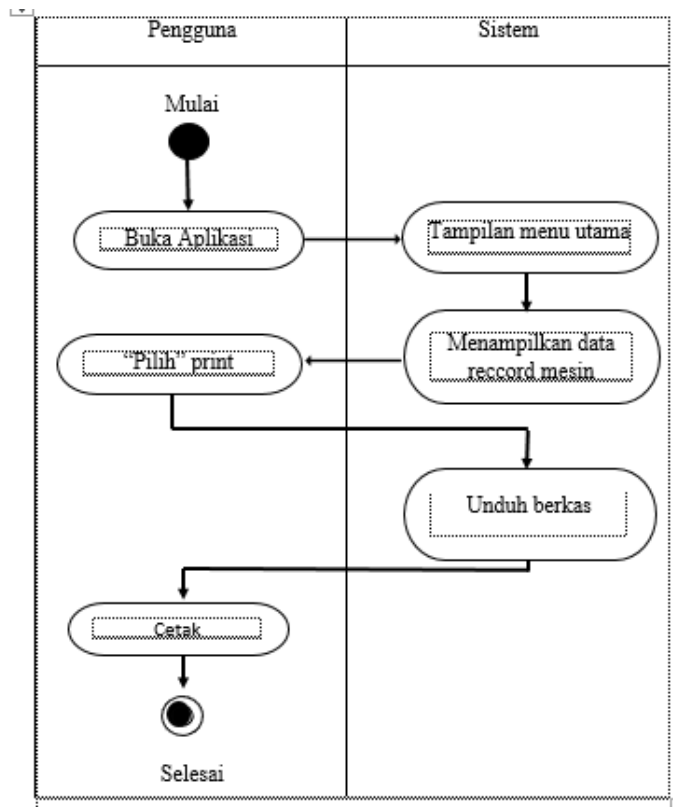
Gambar 3.6 Activity diagram menu *Sort List*

Sumber : (Data penelitian, 2021)

- 1) Saat aplikasi dibuka maka akan ditampilkan menu utama.
- 2) Saat tombol *sort list* di tekan maka akan ditampilkan data *record* mesin untuk menampilkan data bisa diakses keatas atau kebawah.
- 3) Selesai

E. Menu *Print*

Berikut merupakan gambaran alur *activity diagram* menu *print*:



Gambar 3.7 Activity diagram menu Print

Sumber : (Data penelitian, 2021)

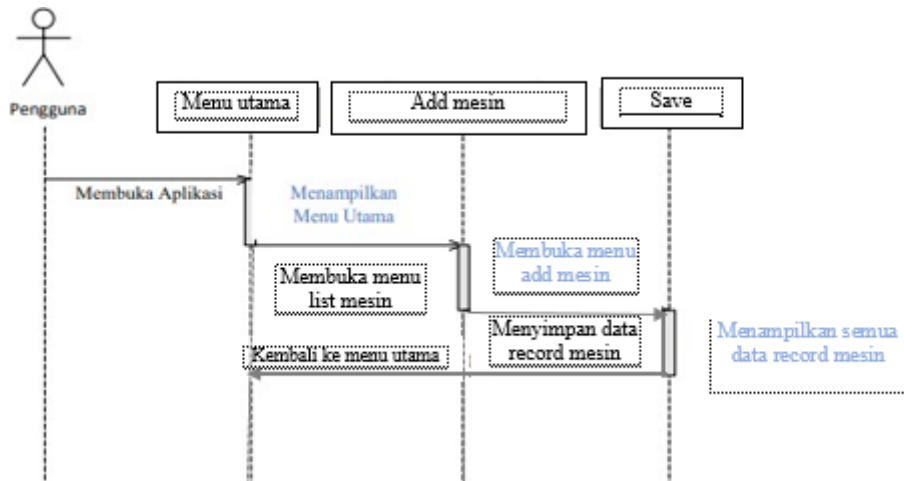
- 1) Saat aplikasi dibuka maka akan ditampilkan menu utama dan langsung menampilkan data *record* mesin.
- 2) Pengguna menekan tombol *print* maka akan mengunduh berkas.
- 3) Selesai

3. Sequence diagram

A. Menu Add mesin

Berikut merupakan *sequence diagram* yang dapat diuraikan pada aplikasi

Image Machine:



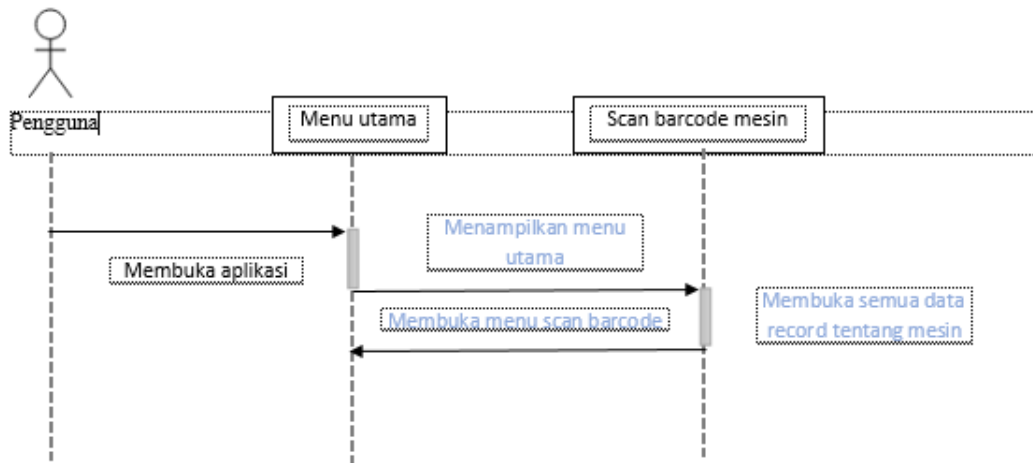
Gambar 3.8 Sequence diagram menu Add Mesin

Sumber : (Data penelitian, 2021)

- 1) Saat aplikasi dibuka *Image Machine* maka menu utama akan ditampilkan.
- 2) Saat tombol *list mesin* ditekan maka *Add mesin* akan terbuka.
- 3) Pengguna akan menyimpan data *record mesin* yang telah diinput.
- 4) Maka aplikasi akan menampilkan semua data *record mesin* .
- 5) Selesai

B. Menu *Scan barcode mesin*

Berikut merupakan gambaran alur pada menu *Scan barcode mesin*:



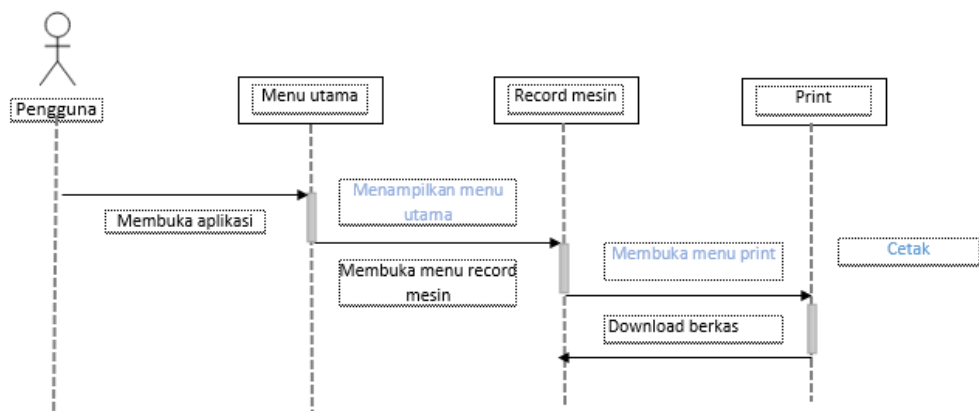
Gambar 3.9 Sequence diagram menu Scan Barcode

Sumber : (Data penelitian, 2021)

- 1) Saat aplikasi dibuka *Image Machine* akan ditampilkan layar pembuka dari aplikasi dan menampilkan menu utama.
- 2) Saat tombol *Scan barcode* mesin di tekan maka akan membuka semua data *record* tentang mesin.
- 3) Selesai

C. Menu *Print*

Berikut merupakan gambaran alur *activity diagram* pada menu *print*:



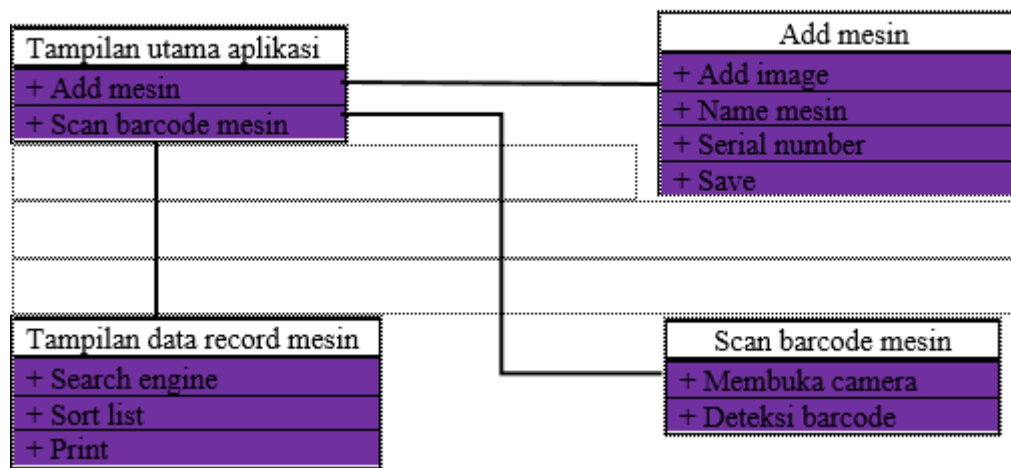
Gambar 3.10 Sequence diagram menu Print

Sumber : (Data penelitian, 2021)

- 1) Saat aplikasi dibuka maka *Image machine* akan ditampilkan layar pembuka dari aplikasi dan menampilkan menu utama.
- 2) Saat tombol *record* mesin di tekan maka *download* berkas.
- 3) Pengguna membuka dengan menekan tombol *print* maka berkas akan tercetak.
- 4) Selesai

4. *Class diagram*

Class diagram merupakan variabel yang saling berhubungan yang menggambarkan tentang aliran sistem. Berikut merupakan alur *class diagram* pada aplikasi *Image Machine*:



Gambar 3.11 *Class diagram*
Sumber : (Data penelitian, 2021)

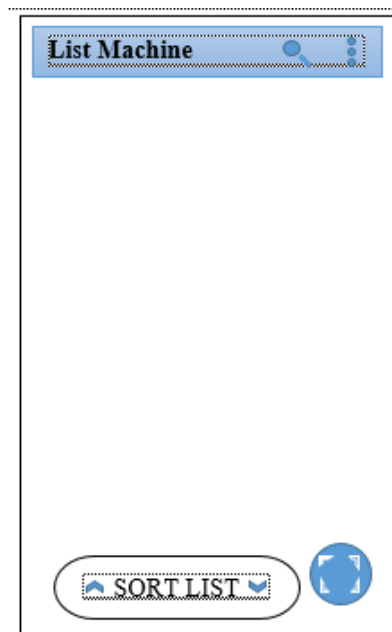
3.3.2 Desain Rancangan (*Story Board*)

Desain rancangan merupakan rancangan aplikasi yang telah di desain secara umum mengenai aplikasi yang dihasilkan supaya mempermudah dalam pembuatan aplikasi tersebut:

Perancangan antarmuka yang akan ditampilkan kepada pengguna adalah:

A. Tampilan halaman utama aplikasi

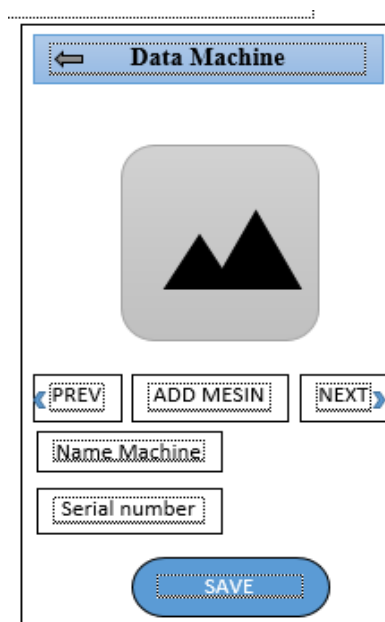
Pada layar aplikasi sebelum mengakses ke halaman berikutnya pengguna terlebih dahulu ditampilkan dengan halaman utama *List machine* yang belum terisi oleh data mesin.



Gambar 3.12 Rancangan aplikasi Menu Utama
Sumber : (Data penelitian, 2021)

B. Rancangan tampilan aplikasi menu *Add mesin*

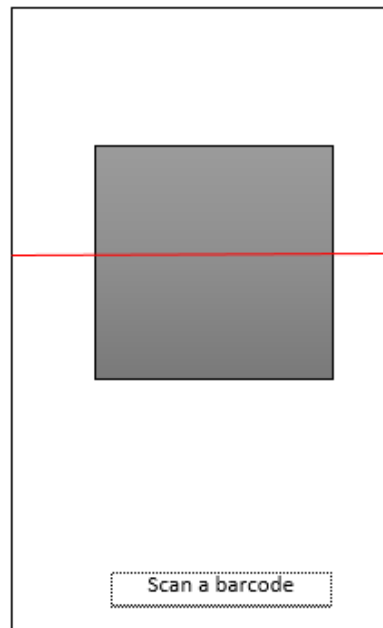
Pengguna masuk menu *Add Machine* ada dua menu pilihan yaitu yang pertama langsung menggunakan camera dan yang kedua diambil dari directory



Gambar 3.13 Rancangan aplikasi Menu *Add mesin*
Sumber : (Data penelitian, 2021)

b. Rancangan tampilan aplikasi menu *Scan Barcode*

Pada menu ini pengguna dapat mengarahkan kamera ke *barcode* mesin maka dengan itu riwayat mesin akan diketahui secara otomatis.

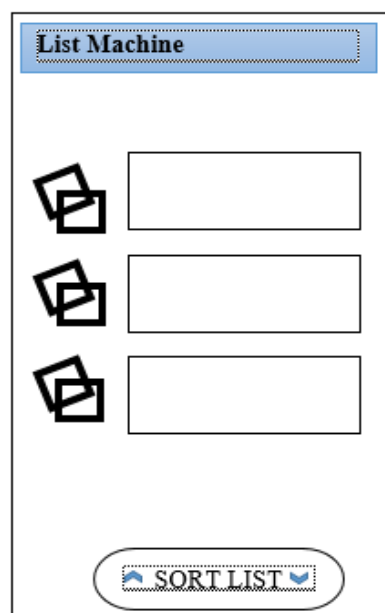


Gambar 3.14 Rancangan aplikasi Menu *Scan Barcode*

Sumber : (Data penelitian, 2021)

c. Rancangan tampilan aplikasi menu *Sort List*

Pada tampilan menu *Sort List* yang sudah terisi pengguna dapat melihat dan memilih semua data mesin dan untuk melihat list mesin keatas atau kebawah.



Gambar 3.15 Rancangan aplikasi Menu *Sort List*

Sumber : (Data penelitian, 2021)

d. Tampilan menu *Info Machine*

Pada menu ini pengguna dapat melihat *serial number*, *name*, *last maintenance*, *maintenance*, dan *best problem* yang ada pada mesin.

Gambar 3.16 Rancangan aplikasi Menu *Info Machine*

Sumber : (Data penelitian, 2021)

e. Tampilan menu *Maintenance Form*

Pengguna dapat mengisi keterangan perbaikan mesin dari *change sparepart*.

Gambar 3.17 Menu *Maintenance Form change*

Sumber : (Data penelitian, 2021)

f. Tampilan menu *Maintenance Form*

Pengguna dapat mengisi keterangan perbaikan mesin dari *Repair*.

The screenshot shows a web form titled "Maintenance Form". At the top, there are three buttons: "PREV" (left arrow), "ADD MESIN" (orange), and "NEXT" (right arrow). Below these is a label "Maintenance:" followed by a text input field. Underneath is a "Type:" label with three radio button options: "Change", "Repair" (which is selected with a checkmark), and "Another". Below the radio buttons is a label "Information" followed by a large text area. At the bottom of the form is a blue "SAVE" button.

Gambar 3.18 Menu *Maintenance Form repair*

Sumber : (Data penelitian, 2021)

g. Tampilan menu *Maintenance Form*

Pengguna dapat mengisi keterangan perbaikan mesin dari *Another*.

The screenshot shows a web form titled "Maintenance Form". At the top, there are three buttons: "PREV" (left arrow), "ADD MESIN" (orange), and "NEXT" (right arrow). Below these is a label "Maintenance:" followed by a text input field containing "Machine Problem". Underneath is a "Type:" label with three radio button options: "Change", "Repair", and "Another" (which is selected with a checkmark). Below the radio buttons is a label "Set type name of this maintenance" followed by a text input field. At the bottom of the form is a blue "SAVE" button.

Gambar 3.19 Menu *Maintenance Form Another*

Sumber : (Data penelitian, 2021)

3.4 Lokasi dan Jadwal Penelitian

Penelitian dilakukan di *Hypermart* Mega Mall, Batam, Kepulauan Riau.

Untuk mempermudah dalam setiap proses penelitian diperlukan jadwal kegiatan.

Berikut jadwal dalam penelitian ini.

Tabel 3.1 Jadwal penelitian

Kegiatan	Waktu kegiatan																			
	Maret 2021				April 2021				Mei 2021				Juni 2021				Juli 2021			
	Minggu ke				Minggu ke				Minggu ke				Minggu ke				Minggu ke			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Pengajuan judul	■	■																		
Penyusunan BAB I			■	■																
Penyusunan BAB II					■	■	■	■												
Penyusunan BAB III									■	■	■	■								
Penyusunan BAB IV													■	■	■	■				
Penyusunan BAB I-V																	■	■	■	■
Pengumpulan skripsi																				■

Sumber: (Data penelitian, 2021)