

BAB II

LANDASAN TEORI

2.1 Penelitian Terkait

Dalam penyusunan skripsi ini, penulis sedikit banyak terinspirasi dan mereferensi dari penelitian-penelitian sebelumnya yang berkaitan dengan latar belakang masalah pada skripsi ini. Penelitian terkait ini menjadi salah satu acuan dalam melakukan penelitian sehingga dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Dari penelitian terkait, tidak menemukan penelitian dengan judul yang sama seperti judul penelitian.

Tabel 2.1 Penelitian Terkait

NO	NAMA	JUDUL	SUMBER	METODE
1.	Sandro Alfeno1, Ririn Eka Cipta Devi, 2017	Implementasi Global Positioning System (GPS) dan Location Based Service (LSB) pada Sistem Informasi Kereta Api untuk Wilayah Jabodetabek	Jurnal Sisfotek Global	Metode penelitian yang digunakan terdiri dari beberapa tahapan yaitu tahapan perencanaan, tahapan analisis, serta tahapan merancang aplikasi.
2.	Julius Firdaus Nduru, 2020	Sistem Pelacakan Kendaraan Berbasis NodeMCU ESP8266 Dan Tampilan Maps Sesuai Tracking	Universitas Sumatera Utara, Medan	Metode Yang Digunakan metode perancangan
3.	Natriya Faisal R Achman, Sunardi, 2020	Simulasi Aplikasi Monitoring Keteta Api Berbasis Android	Jurnal Perkeretaapian Indonesi	Linier Sequential Model (LSM) merupakan langkah penelitian yang mempunyai 4 tahap yang berulang yaitu

				tahap analisis dan studi literatur, desain atau perancangan, pengkodean serta perakitan, dan pengujian.
4.	Prasetya Hari Prabowo, Rusydi Umar, 2016	Pencarian Dan Pemesanan Travel Berbasis Mobile dengan Google Maps API	Prosiding Annual Research Seminar 2016, unsri	Tahap pengembangan perangkat lunak yaitu analisis kebutuhan, perancangan, implementasi dan pengujian

2.2 Aplikasi

Menurut (Hasan Abdurahman dan Asep Ririh Riswaya, 2014), aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya, aplikasi merupakan suatu perangkat komputer yang siap pakai bagi user.

Pengertian aplikasi menurut Jogiyanto (1999:12) adalah penggunaan dalam suatu komputer, instruksi(*instruction*) atau pernyataan(*statement*) yang disusun sedemikian sehingga komputer dapat memproses input menjadi output.

Pengertian aplikasi menurut Kamus Besar Bahasa Indonesia adalah penerapan dari rancang system untuk mengolah data yang menggunakan aturan atau

ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Menurut Wikipedia, aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna.

2.3 Google Maps

Google Maps merupakan peta yang berbentuk virtual sebagai jasa yang gratis diberikan oleh *Google* dan tersedia secara *online*. Untuk menampilkan *Google Maps* dalam halaman web, kita memerlukan aplikasi *interface Google Maps API* yang dapat diakses melalui *javascript*. Pendaftaran *API Key* diperlukan untuk mengakses *Google Maps*. *Domain web* yang dibuat merupakan data pendaftaran *API Key*. Hal ini sangat membantu dalam pencarian lokasi suatu tempat yang memiliki luas wilayah yang besar atau wilayah yang belum kita ketahui sebelumnya. Dengan memanfaatkan *Google Maps API* pada aplikasi ini, dapat mempermudah konsumen atau pelanggan mendapatkan lokasi agen travel terdekat dari dimana pelanggan berada. *Google Maps* mempunyai banyak fasilitas yang dapat digunakan misalnya pencarian lokasi dengan memasukkan kata kunci, seperti kota, jalan, nama tempat. Fasilitas lain yaitu perhitungan rute perjalanan dari satu tempat ke tempat lainnya.

2.4 Location Based Service (LBS)

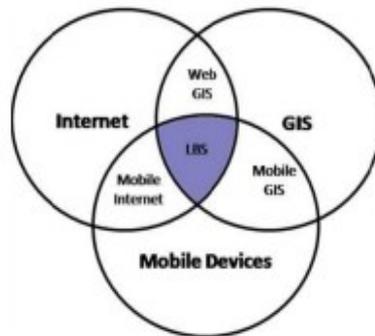
Layanan Berbasis Lokasi atau lebih dikenal dengan *Location Based Service (LBS)* istilah umum yang digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang kita gunakan. LBS adalah layanan informasi yang dapat diakses melalui *mobile device* dengan menggunakan *mobile network*, yang dilengkapi kemampuan untuk memanfaatkan lokasi dari *mobile device* tersebut. Terdapat dua unsur utama pada LBS yaitu:

a) *Location Manager (API Maps)*

Menyediakan *tools/source* untuk LBS, *Application Programming Interface (API) Maps* menyediakan fasilitas untuk menampilkan, memanipulasi maps/peta beserta *feature ± feature* lainnya seperti tampilan satelit, *street* (jalan), maupun gabungannya. Paket ini berada pada *com.google.adroid.maps*.

b) *Location Provider (API Location)*

Menyediakan teknologi pencarian lokasi yang digunakan oleh *device/perangkat*. *API Location* berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi *real-time*. *API Location* berada pada paket *android* yaitu dalam paket *android.location*. Dengan *Location Manager*, kita dapat menentukan lokasi kita saat ini, track gerakan/perpindahan, serta kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan .



Gambar 2.1 Teknologi *Location Based Services*.

2.5 *Microcontroller*

Mikrokontroler adalah sebuah system komputer yang seluruh atau sebagian besar elemennya dikemas dalam satu chip IC, sehingga sering disebut single chip microcomputer. Mikrokontroler merupakan system computer yang mempunyai satu atau beberapa tugas yang sangat spesifik. Elemen mikrokontroler tersebut diantaranya adalah:

- a. Pemroses (processor)
- b. Memori,
- c. Input dan output

Kadang kala pada microcontroller ini beberapa chip digabungkan dalam satu papan rangkaian. Perangkat ini sangat ideal untuk mengerjakan sesuatu yang bersifat khusus, sehingga aplikasi yang diisikan ke dalam komputer ini adalah aplikasi yang bersifat dedicated. Jika dilihat dari harga, mikrocontroller ini harga umumnya lebih murah dibandingkan dengan komputer lainnya, karena perangkatnya relatif sederhana. Mikrocontroller telah banyak digunakan di industri, walaupun penggunaannya masih kurang dibandingkan dengan penggunaan Programable Logic Control (PLC), tetapi mikrocontroller memiliki beberapa keuntungan dibandingkan dengan PLC. Ukuran mikrocontroller lebih kecil dibandingkan dengan suatu modul PLC sehingga peletakannya dapat lebih flexible.

2.6 Internet Of Think (IoT)

Internet of Things (IOT) adalah struktur di mana objek, orang disediakan dengan identitas eksklusif dan kemampuan untuk pindah data melalui jaringan tanpa memerlukan dua arah antara manusia ke manusia yaitu sumber ke tujuan atau interaksi manusia ke komputer. Internet of Things merupakan perkembangan keilmuan yang sangat menjanjikan untuk mengoptimalkan kehidupan berdasarkan sensor cerdas dan peralatan pintar yang bekerjasama melalui jaringan internet.

2.7 Kereta Api

Kereta api adalah sarana transportasi berupa kendaraan dengan tenaga gerak, baik berjalan sendiri maupun dirangkaikan dengan kendaraan lainnya, yang akan ataupun sedang bergerak di rel. Kereta api merupakan alat transportasi masal yang umumnya terdiri dari lokomotif (kendaraan dengan tenaga gerak yang berjalan

sendiri) dan rangkaian kereta atau gerbong (dirangkaian dengan kendaraan lainnya). Rangkaian

kereta atau gerbong tersebut berukuran relatif luas sehingga mampu memuat penumpang maupun barang dalam skala besar. Karena sifatnya sebagai angkutan massal efektif, beberapa negara berusaha memanfaatkannya secara maksimal sebagai alat transportasi utama angkutan darat baik di dalam kota, antarkota, maupun antarnegara.

2.8 *Android*

Sistem Operasi yang digunakan dalam aplikasi ini adalah *android*. *Android* adalah sebuah sistem operasi pada *handphone* yang bersifat terbuka dan berbasis pada sistem operasi *Linux*. Dikarenakan bersifat terbuka, maka pengguna dapat mengembangkan aplikasi berdasarkan kebutuhannya (Ekawati, 2017).

Android adalah sistem operasi berbasis *Linux* bagi telepon seluler seperti telepon pintar dan komputer tablet. *Android* juga menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk berbagai macam piranti gerak (Nazruddin Safaat dalam Morrys E Saubaki, 2020).

2.4.4 Versi *Android*

1. *Android* Versi 1.1

Android memang diluncurkan pertama kali pada tahun 2007, namun sistem operasi ini mulai dirilis dan diterapkan ke berbagai *gadget* pada tanggal 9 Maret 2009 silam. *Android* versi 1.1 merupakan *Android* awal yang dimana versi ini baru memberikan sentuhan di beberapa aplikasinya seperti sistem antar muka bagi pengguna (*user interface*) yang lebih baik, serta beberapa aplikasi yang lain.

2. *Android* Versi 1.5 (*Cupcake*)

Pada bulan Mei 2009 *Android* kembali mengalami perubahan versi. *Android* versi 1.1 kemudian disempurnakan dengan *Android* versi 1.5 atau yang dikenal sebagai *Android Cupcake*.

3. *Android* Versi 1.6 (*Donut*)

Donut (versi 1.6) diluncurkan dalam tempo kurang dari 4 bulan semenjak peluncuran perdana *Android Cupcake*, yaitu pada bulan September 2009.

4. *Android* Versi 2.0/2.1 (*Eclair*)

Masih ditahun yang sama, *Android* kembali merilis operating sistem versi terbarunya, yaitu *Android* versi 2.0/2.1 *Eclair*. *Android Eclair* diluncurkan oleh Google 3 bulan setelah peluncuran.

5. *Android* Versi 2.2 (*Froyo: Frozen Yoghurt*)

Butuh 5 bulan bagi Google untuk melakukan regenerasi dari *Android Eclair* versi sebelumnya ke versi *Froyo Frozen Yoghurt*. Pada tanggal 20 Mei 2010, *Android* versi 2.2 alias *Android Froyo* ini dirilis.

6. *Android* Versi 2.3 (*Gingerbread*)

7 bulan kemudian *Android* kembali melakukan gebrakan dengan merilis kembali *Android* versi 2.3 atau yang dikenal sebagai *Android Gingerbread*

7. *Android* Versi 3.0/3.1 (*Honeycomb*)

Pada bulan Mei 2011 *Android* versi 3.0/3.1 atau *Android Honeycom* dirilis. *Android Honeycomb* merupakan sebuah sistem operasi *Android* yang tujuannya memang dikhususkan bagi penggunaan tablet berbasis *Android*.

8. *Android* Versi 4.0 (*Ice Cream Sandwich*)

Android ICS atau *Ice Cream Sandwich* juga dirilis pada tahun yang sama dengan *Honeycomb*, yaitu pada bulan Oktober 2011.

9. *Android* Versi 4.1 (*Jelly Bean*)

Android Jelly Bean merupakan versi *Android* yang terbaru pada saat ini. Salah satu *gadget* yang menggunakan sistem operasi *Jelly Bean* adalah Google

Nexus 7 yang diprakarsai oleh *ASUS*, vendor asal Taiwan yang juga menjadi teman satu kampung halaman dengan *Acer*.

10. *Android* Versi 4.4 (*Kit Kat*)

Kehadiran *Android* *kitkat* merupakan peluncuran produk OS anyar yang diluncurkan pada 4 september 2013, sebelumnya banyak kabar beredar jika kalau *Android* akan meluncurkan OS baru yang bernama *Android Key Lime Pie* namun setelah di analisa tidak sesuai dengan ejaan orang umum, sehingga namanya diganti dengan OS *Android Kit-Kat* yang sebagian besar orang sudah familiar dengan itu.

11. *Android* Versi 5.0.2 (*Lollipop*)

Android Lollipop merupakan keberadaan OS *Android* yang memang saat ini sudah menjadi trend baru di industri *smartphone*, hal ini tak lepas dari keunikan dan kelebihan yang banyak di miliki dari OS tersebut. Kehadiran *Android* versi ini amat di nanti oleh sekian banyak orang karna diharapkan sistem operasi *Lollipop* ini bias lebih baik dibandingkan versi-versi sebelumnya.

12. *Android* Versi 6.0 (*Marshmallow*)

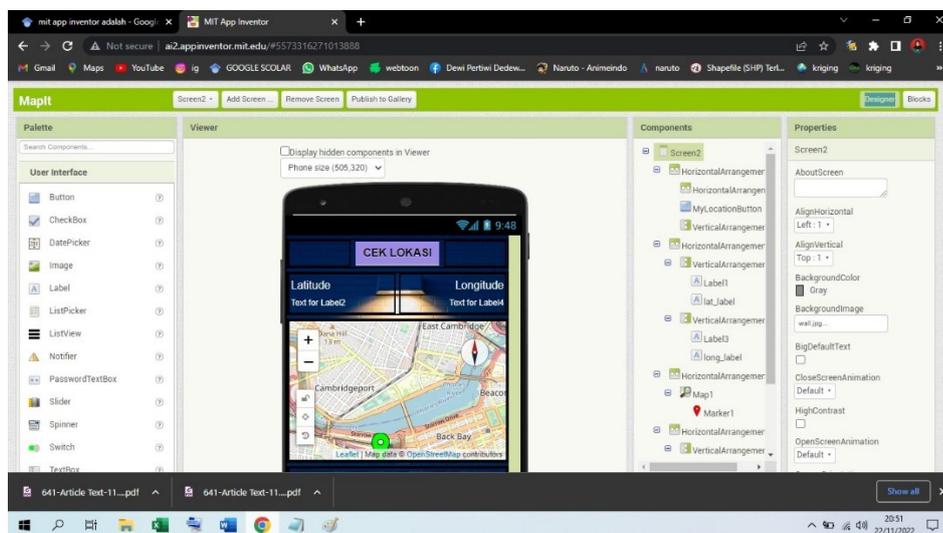
Android 6.0 Marshmallow adalah versi dari sistem operasi *mobile Android*. Pertama kali diperkenalkan Mei 2015 di Google I / O di bawah kode nama *Android M*, secara resmi dirilis pada Oktober 2015. *Android Marshmallow* memperkenalkan model izin aplikasi di desain ulang sekarang ada hanya delapan kategori izin, dan aplikasi yang tidak lagi secara otomatis diberikan semua hak akses mereka ditentukan pada waktu instalasi.

2.9 Perangkat Lunak Yang Digunakan

2.9.1 MIT App Inventor

MIT App Inventor adalah bahasa pemrograman berbasis blok visual. Ini awalnya produk Google, namun kemudian dirilis sebagai perangkat lunak sumber terbuka yang dikelola oleh *Massachusetts Institute of Technology*. *MIT*

App Inventor memungkinkan *nonprogrammers* untuk membuat aplikasi *mobile* untuk perangkat yang menjalankan sistem operasi Android dan mampu mengembangkan semua jenis aplikasi, tidak hanya animasi. *MIT App Inventor* memiliki dua jendela utama: Designer Komponen untuk membangun antar muka pengguna dan Editor Blok untuk menentukan perilaku aplikasi. Perancang Komponen memungkinkan pemilihan "komponen" (objek dari kelas tertentu) yang akan ditambahkan, dengan menyeretnya ke tata letak layar, sebagai bagian dari aplikasi. Setelah bagian ditambahkan ke aplikasi, propertinya dapat ditetapkan. Editor Blok menggunakan teka-teki "kerangka" gambar (metode kelas terkait) yang digabungkan untuk menentukan perilaku aplikasi sebagai respons terhadap berbagai peristiwa. Pengguna dapat menguji aplikasi mereka secara langsung di ponsel Android atau emulator dari ponsel yang berjalan di komputer. *MIT App Inventor* memungkinkan siswa untuk dapat fokus pada aspek-aspek yang menarik dari pemecahan masalah desain pemrograman dan dapat membangun aplikasi *mobile* yang kreatif. *MIT App Inventor* server juga dapat menyimpan semua proyek serta memungkinkan pengguna bisa masuk ke akun mereka untuk melanjutkan pekerjaan kapan dan dimana saja.



Gambar 2.2 MIT App Inventor

2.9.2 *Firestore*

Firestore adalah *Cloud Service Provider* dan *Backend as a Service* yang dimiliki oleh Google. *Firestore* merupakan solusi yang ditawarkan oleh Google untuk mempermudah dalam pengembangan aplikasi *mobile* maupun web dan bersifat *Realtime Database*. *Firestore* memiliki banyak SDK yang memungkinkan untuk mengintegrasikan layanan ini dengan *Android*, *iOS*, *Javascript*, *C++* hingga *Unity*. Pada penggunaan *Firestore* diperlukan akses internet dalam menjalankan aplikasi tersebut. Dikarenakan data yang tersimpan pada tempat penyimpanan berbasis *Online* (Nadia Firly, 2019).

2.9.2.1 Fitur *Firestore*

Setelah API *Firestore* dimasukkan ke Dalam aplikasi *Android* atau *iOS*, pengembang bisa menggunakan fitur *Firestore* dengan coding yang simpel. Beberapa fitur yang disediakan *Firestore*, yaitu :

1. *Analytics*

Fitur ini membuat para pengembang aplikasi dapat mengerti bagaimana para pengguna menggunakan aplikasi mereka. SDK *analytics* menangkap sendiri data yang dibutuhkan oleh para *developer*. *Dashboard* juga menyediakan detil seperti pengguna paling aktif atau fitur apa saja yang paling digunakan dalam aplikasi tersebut. *Analytics* juga menyediakan data yang telah dirangkum.

2. *Authentication*

Fitur ini membuat para pengembang dapat mengizinkan pengguna untuk mengakses aplikasi. *Firestore* menyediakan fitur login melalui Gmail, Github, Twitter, Facebook dan juga autentikasi buatan sendiri.

3. *Messaging*

FCM (*Firestore Cloud Messaging*) membuat para pengguna dapat mengirimkan pesan ke berbagai platform tanpa biaya tambahan. *Messaging* juga dapat digunakan untuk kebutuhan notifikasi.

4. *Real-time Database*

Database di *Firebase* adalah database berbasis *cloud* dan tidak membutuhkan SQL untuk mengambil dan menyimpan data atau bisa disebut juga NoSQL. Database ini sangatlah cepat dan dapat diandalkan yang artinya data dapat dibaharui dan disinkronisasikan dengan cepat. Data juga dijaga meskipun pengguna kehilangan koneksi internetnya.

5. *Storage*

Firebase juga menyediakan fasilitas penyimpanan. *Firebase* dapat menyimpan dan mengambil konten seperti gambar, video, dan audio langsung dari SDK. Meng-*upload* dan men-*download* juga dilakukan di *background*. Data yang disimpan akan aman dan hanya pengguna yang diijinkan yang dapat mengaksesnya.

6. *Hosting*

Firebase juga menyediakan fitur *hosting*. *Firebase* mengirimkan konten web secara cepat dan konten selalu dikirim dengan aman.

7. *Crash reporting*

Fitur *crash reporting* di *Firebase* membuat pengembang dapat mengetahui kesalahan ketika terjadi *crash*.

2.10 Perangkat Keras Yang Digunakan

2.10.1 *Arduino*

Arduino merupakan *platform* yang terdiri dari *software* dan *hardware*. *Hardware Arduino* sama dengan mikrocontroller pada umumnya hanya pada *arduino* ditambahkan penamaan pin agar mudah diingat. *Software Arduino* merupakan *software open source*. Sehingga *software* ini digunakan untuk membuat dan memasukkan program ke dalam *arduino*. Pemrograman *arduino* tidak sebanyak tahapan mikrocontroller konvensional karena *arduino* sudah didesain mudah untuk dipelajari, sehingga para pemula dapat mulai belajar mikrocontroller dengan *arduino*. (Sulaiman dalam JauhariArifin, dkk, 2016).

Arduino adalah suatu perangkat prototype elektronik berbasis mikrokontroller yang fleksibel dan open-source, perangkat keras dan perangkat lunaknya mudah digunakan. (Dave Michael, Dian Gustina, 2019).



Gambar 2.3 Arduino

2.10.2 NodeMCU ESP8266

NodeMCU adalah sebuah *platform IoT* yang bersifat *open source*. Terdiri dari perangkat keras berupa *System on Chip (SoC) ESP8266* buatan *Espressif System*, *firmware* dari *NodeMCU* menggunakan bahasa pemrograman scripting *Lua*, bersifat *open source* dan banyak tersedia salah satunya oleh *user* di *GitHub*. Istilah *NodeMCU* secara umum sebenarnya mengacu pada *firmware* yang digunakan dari perangkat keras development kit. (Abizar Rachman dkk, 2020).

2.11 Wifi

WiFi (Wireless Fidelity) merupakan bentuk pemanfaatan teknologi *Wireless Local Area Network (WLAN)* pada lokasi-lokasi publik. *WiFi* memiliki pengertian yaitu sekumpulan standar yang digunakan untuk Jaringan Lokal Nirkabel (*Wireless Local Area Networks-WLAN*) yang didasari pada spesifikasi *IEEE802.11* (Sandi, M dalam LuhPutuAyu Sri Aryaningrum, Rina PudjiAstuti, Arfianto Fahmi, 2016).

2.11.1 Spesifikasi *Wifi*

Secara teknis operasional, Wi-Fi merupakan salah satu varian teknologi komunikasi dan informasi yang bekerja pada jaringan dan perangkat WLANs (*wireless local area network*).

Tabel 2.2 Spesifikasi *Wifi*

Spesifikasi <i>Wifi</i>		
Spesifikasi	Kecepatan	Frekuensi Band
802.11b	11 Mb/s	2.4 GHz
802.11a	54 Mb/s	5 GHz
802.11g	54 Mb/s	2.4 GHz
802.11n	100 Mb/s	2.4 GHz

Dengan kata lain, Wi-Fi adalah nama dagang (*certification*) yang diberikan pabrikan kepada perangkat telekomunikasi (Internet) yang bekerja di jaringan WLANs dan sudah memenuhi kualitas interoperability yang dipersyaratkan. Wi-Fi dirancang berdasarkan spesifikasi IEEE 802.11. Sekarang ini ada empat variasi dari 802.11, yaitu: 802.11a, 802.11b, 802.11g, and 802.11n.

2.12 Metode Pengembang Perangkat Lunak

2.12.1 Metode *Prototype*

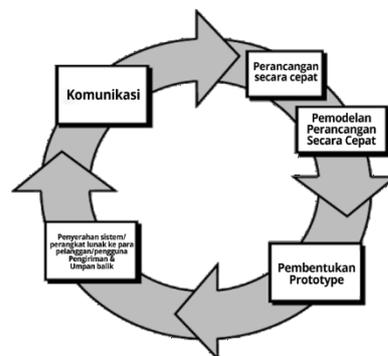
Metode pengembang perangkat lunak yang digunakan pada aplikasi ini adalah metode *prototype*. Dalam melakukan perancangan sistem yang akan dikembangkan dapat menggunakan metode *prototype*. Metode ini cocok digunakan untuk mengembangkan sebuah perangkat yang akan dikembangkan kembali. Kemudian membuat sebuah rancangan kilat yang selanjutnya akan dievaluasi kembali sebelum diproduksi secara benar.

Prototype bukanlah merupakan sesuatu yang lengkap, tetapi sesuatu yang harus dievaluasi dan dimodifikasi kembali. Segala perubahan dapat terjadi

pada saat *prototype* dibuat untuk memenuhi kebutuhan pengguna dan pada saat yang sama memungkinkan pengembang untuk lebih memahami kebutuhan pengguna secara lebih baik (Pressman 2012).

Berikut adalah tahapan dalam metode *prototype* :

- a. Komunikasi dan pengumpulan data awal, yaitu wawancara terhadap pihak yang terkait dalam penelitian dan analisis terhadap kebutuhan pengguna (wawancara terhadap owner atau karyawan salon)
- b. Perencanaan Secara Cepat yaitu pembuatan desain secara umum untuk selanjutnya dikembangkan kembali.
- c. Pembentukan *Prototype* yaitu pembuatan perangkat prototype termasuk pengujian dan penyempurnaan.
- d. Evaluasi terhadap *Prototype*, yaitu mengevaluasi prototype dan memperhalus analisis terhadap kebutuhan pengguna.
- e. Perbaikan *Prototype*, yaitu pembuatan tipe yang sebenarnya berdasarkan hasil dari evaluasi *prototype*.
- f. Produksi akhir, yaitu memproduksi perangkat secara benar sehingga dapat digunakan oleh pengguna.



Gambar 2.4 Metode *Prototype* (Pressman, 2012)

2.13 Pengujian *Black-Box Testing*

Metode pengujian pada aplikasi ini menggunakan *Black Box Testing* yaitu yang berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program.

Black Box Testing bukanlah solusi alternatif dari *White Box Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing* (Pressman 2012). *Black Box Testing* cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

2.14 Sistem Pemodelan

2.14.1 *Unified Modelling Language (UML)*

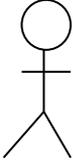
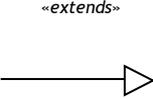
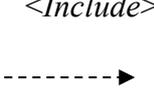
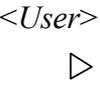
Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek. *Unified Modeling Language (UML)* bukanlah merupakan bahasa pemrograman tetapi model-model yang tercipta berhubungan langsung dengan berbagai macam bahasa pemrograman berorientasi obyek, seperti Java (Syafitri 2016). UML tersusun atas sejumlah elemen grafis membentuk diagram-diagram. Dalam penelitian ini melakukan desain hanya 2 diagram yaitu *Use Case Diagram* dan *Activity Diagram*.

2.14.1.1 *Use Case Diagram*

Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem ini berinteraksi dengan dunia luar. *Use case diagram* dapat digunakan untuk memperoleh kebutuhan sistem dan memahami

bagaimana sistem seharusnya bekerja (Syafitri 2016). Berikut ini adalah simbol-simbol yang ada pada *Use Case Diagram* terdapat pada tabel 2.1

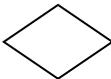
Tabel 2.3 Simbol *Use Case Diagram*

Keterangan	Simbol	Deskripsi
<i>Use Case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; dinyatakan dengan menggunakan kata kerja diawal-awal frase nama <i>use case</i>
Aktor		Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar itu sendiri. Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang
Asosiasi		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
Ekstensi		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> , dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek.
Generalisasi		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan/ <i>include/</i> <i>uses</i>	 	Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> : a. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahandijalankan <i>Include</i> berarti <i>use case</i> yang tambahan akanselalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahandijalankan.

2.14.1.2 Activity Diagram

Activity Diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, keputusan yang mungkin terjadi, dan bagaimana suatu aktivitas berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa kegiatan. Aktivitas menggambarkan proses yang akan berjalan, (Syafitri 2016). Berikut ini adalah simbol-simbol yang ada pada *Activity Diagram* terdapat pada tabel 2.4.

Tabel 2.4 Simbol *Activity Diagram*

Keterangan	Simbol	Deskripsi
Status awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan		Asosiasi percabangan dimanajika ada pilihan aktivitas lebih dari satu.
Penggabungan		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
<i>Swimlane</i>		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
Status Akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.