

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Tinjauan Teori Umum**

Dalam penyusunan ini dicantumkan beberapa teori umum yang dapat dijadikan sebagai landasan teori, berikut penjelasan dari beberapa teori, diantaranya adalah sebagai berikut:

##### **2.1.1. Definisi Sistem**

Menurut (Winarto,2004) dalam jurnal (Asnawati & Kusuma, 2011:76-102), Sistem dapat diartikan sebagai suatu kumpulan yang terdiri dari beberapa bagian komponen yang bekerja sama untuk mencapai tujuan tertentu. Yang masing-masing komponen memiliki fungsi yang berbeda dengan yang lain, tetapi tetap bekerja sama.

Menurut (Amsyah, 2000) dalam jurnal (Asnawati & Kusuma, 2011) Sistem pada dasarnya adalah suatu susunan yang teratur dari kegiatan yang berhubungan satu sama lain dan prosedur-prosedur yang berkaitan dari organisasi atau lembaga yang bersifat kompleks.

##### **2.1.2. Karakteristik Sistem**

Menurut (Husda, 2012:112-114) suatu sistem biasanya mempunyai karakteristik tertentu yang dapat dijelaskan adalah sebagai berikut :

1. Komponen (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses secara keseluruhan.

2. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan.

3. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar sistem yaitu segala sesuatu diluar dari batas sistem yang mempengaruhi operasi dari suatu sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan atau merugikan.

4. Penghubung Sistem (*Interface*)

Penghubung sistem merupakan media penghubung antar satu subsistem dengan subsistem yang lainnya. Untuk membentuk satu kesatuan, sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya. Dengan kata lain *output* dari suatu subsistem akan menjadi *input* dari subsistem yang lainnya.

#### 5. Masukan Sistem (*Input*)

Masukan sistem merupakan energi yang dimasukkan ke dalam sistem. Masukan sistem dapat berupa masukan perawatan (*Maintenance Input*) dan masukan sinyal (*Signal Input*). Masukan perawatan (*Maintenance Input*) adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Masukan sinyal (*Signal Input*) adalah energi yang diproses untuk didapatkan keluaran.

#### 6. Keluaran Sistem (*Output*)

Keluaran sistem merupakan hasil dari energi yang diolah oleh sistem. Keluaran sistem meliputi keluaran yang berguna dan keluaran yang tidak berguna. Keluaran yang berguna, contohnya informasi yang dikeluarkan oleh komputer. Keluaran yang tidak berguna, contohnya panas yang dikeluarkan oleh komputer.

#### 7. Pengolah Sistem (*Process*)

Pengolah sistem merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan. Contohnya CPU pada komputer, bagian produksi yang mengubah bahan baku menjadi barang jadi, bagian akuntansi yang mengolah data transaksi menjadi laporan keuangan.

#### 8. Tujuan Sistem (*Goal*)

Setiap sistem pasti mempunyai tujuan ataupun sasaran yang mempengaruhi *input* yang dibutuhkan dan *output* yang dihasilkan. Dengan kata lain suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai

sasaran atau tujuannya. Sistem yang tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

### 2.1.3. Klasifikasi Sistem

Menurut (Husda, 2012:115-116), suatu sistem biasanya mempunyai klasifikasi dari beberapa sudut pandang, diantaranya yaitu :

1. Sistem Abstrak (*Abstract System*)

Sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik (Sistem Teologia yang merupakan suatu sistem yang menggambarkan hubungan Tuhan dengan Manusia).

2. Sistem Fisik (*Physical System*)

Merupakan sistem yang secara fisik sehingga setiap makhluk dapat melihatnya (Sistem Komputer, Sistem Akuntansi, Sistem Produksi).

3. Sistem Alamiah (*Natural System*)

Sistem yang terjadi melalui proses alam dalam artian tidak dibuat oleh manusia (Sistem Tata Surya, Sistem Galaxi, Sistem Reproduksi).

4. Sistem Buatan Manusia (*Human Made System*)

Sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut *human machine system* (Contoh: Sistem Informasi).

5. Sistem Tertentu (*Deterministic System*)

Sistem beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan (Contoh: Sistem Komputer).

6. Sistem Tak Tentu (*Probabilistic System*)

Sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas (Contoh: Sistem Manusia).

7. Sistem Tertutup (*Close System*)

Sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara *relative* tertutup, tidak benar-benar tertutup).

8. Sistem Terbuka (*Open System*)

Sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih spesifik dikenal juga yang disebut dengan sistem terotomasi, yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh satu atau lebih komputer sebagai bagian dari sistem yang digunakan dalam masyarakat modern.

#### **2.1.4. Definisi Data**

Menurut (Husda, 2012:117-18), Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah sesuatu yang terjadi pada saat tertentu. Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu metode untuk menghasilkan suatu informasi.

Menurut (Edhy Sutanta, 2004) dalam jurnal (Sembiring & Nurhayati, 2012:13-19), Data adalah sebagai bahan keterangan tentang kejadian nyata atau fakta-fakta yang dirumuskan dalam sekelompok lambang tertentu yang tidak acak yang menunjukkan jumlah, tindakan, atau hal-hal. Data dapat berupa catatan-catatan dalam kertas, buku, atau tersimpan sebagai file dalam basis data.

#### **2.1.5. Definisi Informasi**

Menurut (Husda, 2012:117), Informasi merupakan sesuatu yang memiliki arti yang sangat penting didalam mendukung proses pengambilan keputusan oleh pihak manajemen. Secara umum informasi dapat didefinisikan sebagai data yang diolah menjadi bentuk yang lebih berarti bagi yang menerimanya.

Menurut (Al Bahra, 2005) dalam jurnal (Sembiring & Nurhayati, 2012:13-19), Informasi adalah data yang telah diolah menjadi bentuk yang lebih berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun akan datang.

### **2.1.6. Kualitas Informasi**

Informasi yang berkualitas harus akurat, tepat pada waktunya dan relevan.

Menurut (Husda, 2012), kualitas informasi dapat dijelaskan sebagai berikut:

1. Akurat

Akurat berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bisa atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan terjadi gangguan yang dapat merubah atau merusak informasi tersebut.

2. Tepat Waktu

Tepat waktu berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak mempunyai nilai lagi. Saat ini mahalny nilai informasi disebabkan harus cepatnya informasi itu didapat sehingga diperlukan teknologi-teknologi mutakhir untuk mendapatkan, mengolah dan mengirimkannya.

3. Relevan

Relevan berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang berbeda-beda.

### **2.1.7. Definisi Sistem Informasi**

Menurut (Husda, 2012:119-120), Sistem informasi dapat didefinisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media prosedur-prosedur dan pengendalian yang

ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan.

Menurut (Nugraha, 2014:27-32), Sistem informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (*input*) menjadi keluaran (informasi), guna mencapai sasaran-sasaran perusahaan.

#### **2.1.8. Komponen Sistem Informasi**

Menurut (Husda, 2012:120-122), Sistem informasi mempunyai enam buah komponen atau disebut juga dengan blok bangunan (*building block*). Keenam komponen ini harus ada bersama-sama dan membentuk satu kesatuan. Jika satu atau lebih komponen tersebut tidak ada, maka sistem informasi tidak akan dapat melakukan fungsinya, yaitu pengolahan data dan tidak dapat mencapai tujuannya, yaitu menghasilkan informasi yang relevan, tepat waktu dan akurat. Komponen-komponen dari sistem informasi ini dapat digambarkan sebagai berikut :

1. Blok Masukan (*Input Block*)

Input merupakan data yang masuk ke dalam sistem informasi.

2. Blok Model (*Model Block*)

Blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

Keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi (*Technology Block*)

Teknologi merupakan kotak alat (*tool box*) dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara menyeluruh.

5. Blok Basis Data (*Database Block*)

Basis data merupakan kumpulan dari data yang saling berhubungan satu sama lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Blok Kendali (*Control Block*)

Beberapa pengendalian yang dirancang secara khusus untuk menanggulangi gangguan-gangguan terhadap sistem.

### **2.1.9. *Software Development Life Cycle (SDLC)***

Menurut (A.S & Shalahuddin, 2011:24), SDLC atau *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik).

### 2.1.9.1. Model *Waterfall*

Menurut (A.S & Shalahuddin, 2011:26-28), Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linear (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*). Berikut adalah gambar model air terjun:

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk disokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multilangkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranlasikan kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

### 3. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

### 4. Pengujian

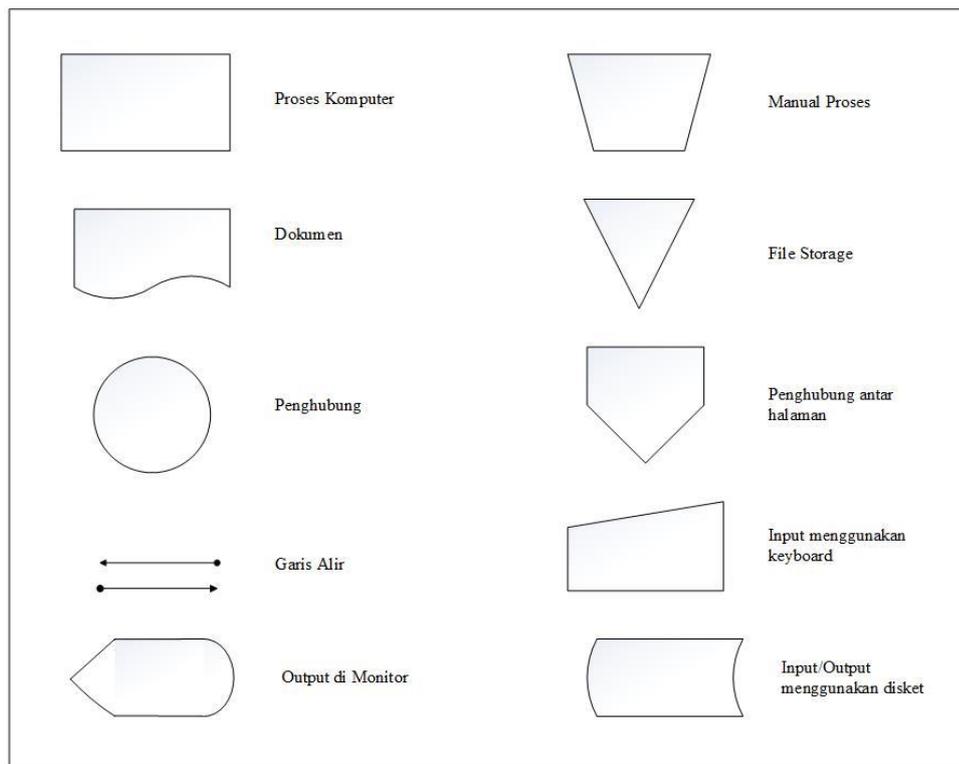
Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

### 5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

#### **2.1.10. Aliran Sistem Informasi**

Menurut (Ismael, 2017:147-156), Aliran Sistem Informasi (ASI) merupakan bagan yang menunjukkan arus pekerjaan serta keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur prosedur yang ada dalam sistem. Adapun simbol-simbol yang dapat digunakan pada Aliran Sistem Informasi (ASI) ini adalah sebagai berikut:



**Gambar 2.1** Aliran Sistem Informasi

## 2.2. Tinjauan Teori Khusus

Dalam penyusunan ini dicantumkan beberapa teori khusus yang menjelaskan tentang Pencarian Lokasi ATM BCA Berbasis Android, berikut penjelasan dari beberapa teori, diantaranya adalah sebagai berikut :

### **2.2.1. Definisi Perancangan Sistem**

Menurut (Hendarti & Haryanto, 2009:155-161), Perancangan Sistem adalah proses mengimplementasikan hasil-hasil dari analisis sistem ke dalam suatu rancangan sistem yang baru.

Menurut (AlFatta & Marco, 2015:63-91), Perancangan sistem mulanya diawali dengan menentukan segala keperluan yang akan memenuhi apa yang dibutuhkan oleh sistem, siapa yang mengambil langkah dan bagaimana cara menyesuaikan. Pada dasarnya perancangan sistem bergerak dari *input* menuju ke *output* sistem, yang terdiri dari *reports* dan file untuk memenuhi kebutuhan organisasi.

#### **2.2.1.1. Tahap-tahap Perancangan Sistem**

Menurut McLeod dalam jurnal (Hendarti & Haryanto, 2009:155-161), yang diterjemahkan oleh (Teguh, 2001), tahap-tahap perancangan sistem informasi adalah sebagai berikut:

1. Menyiapkan rancangan sistem yang terinci
2. Mengidentifikasi berbagai alternatif konfigurasi sistem
3. Mengevaluasi berbagai alternatif konfigurasi sistem
4. Memilih konfigurasi terbaik
5. Menyiapkan usulan penerapan
6. Menyetujui atau menolak penerapan sistem

### **2.2.2. Pengujian *Black-Box Testing***

Menurut (A.S & Shalahuddin, 2011:213-214), *Black-Box Testing* yaitu perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan.

### **2.2.3. Definisi *Unified Modeling Language (UML)***

Menurut (Herlawati, 2011) dalam jurnal (Nurhadiyan & Hidayatullah, 2017:46-52) *Unified Modeling Language (UML)* adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem. *Unified Modeling Language (UML)* adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya.

UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM.

### 2.2.3.1. Diagram UML

Berikut adalah diagram UML menurut (A.S & Shalahuddin, 2011:120-144) yaitu:

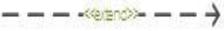
#### 1. *Use Case Diagram*

*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Ada dua hal utama pada *use case* yaitu pendefinisian aktor dan *use case*. Berikut adalah simbol-simbol yang ada pada diagram *use case*:

**Tabel 2.1** *Use Case Diagram*

Simbol	Deskripsi
<p data-bbox="422 1429 502 1460"><i>Actor</i></p> 	<p data-bbox="635 1393 1353 1720">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>

Table 2.1 Lanjutan

Simbol	Deskripsi
<p data-bbox="400 501 523 533"><i>Use Case</i></p> 	<p data-bbox="635 439 1353 636">Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>.</p>
<p data-bbox="384 763 539 795"><i>Association</i></p> 	<p data-bbox="635 725 1353 869">Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p data-bbox="416 1032 507 1064"><i>Extend</i></p> 	<p data-bbox="635 911 1353 1218">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, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.</p>
<p data-bbox="363 1312 560 1344"><i>Generalization</i></p> 	<p data-bbox="635 1263 1353 1406">Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p data-bbox="411 1671 512 1702"><i>Include</i></p> 	<p data-bbox="635 1476 1353 1673">Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p> <p data-bbox="635 1695 1353 1785">Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <ol data-bbox="655 1807 1353 1951" style="list-style-type: none"> <li>1. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.</li> </ol>

	2. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan.
--	---

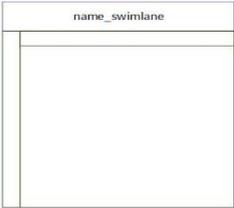
## 2. *Activity Diagram*

Diagram aktivitas atau *activity diagram* yaitu menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

**Tabel 2.2** *Activity Diagram*

<b>Simbol</b>	<b>Deskripsi</b>
<p><i>Status Awal</i></p> 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
<p>Aktivitas</p> 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
<p>Percabangan / decision</p>  <p>percabangan / decision</p>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.

**Table 2.2** Lanjutan

<p>Penggabungan / join</p> 	<p>Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.</p>
<p>Status / akhir</p> 	<p>Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.</p>
<p>Swimlane</p> 	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.</p>

### 3. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antarobjek.

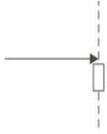
**Tabel 2.3** *Sequence Diagram*

<b>Simbol</b>	<b>Deskripsi</b>
<p><i>Actor</i></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, biasanya dinyatakan dinyatakan menggunakan kata benda di awal frase nama aktor.</p>

Table 2.3 Lanjutan

<p>Garis hidup / <i>lifeline</i></p> 	Menyatakan kehidupan suatu objek.
<p>Objek</p> <div style="border: 1px solid black; padding: 2px; width: fit-content;">nama objek : nama kelas</div>	Menyatakan objek yang berinteraksi pesan.
<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
<p>Pesan tipe create</p> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
<p>Pesan tipe <i>call</i></p> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
<p>Pesan tipe <i>send</i></p> 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
<p>Pesan tipe <i>return</i></p> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode mnghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

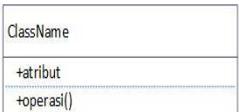
Table 2.3 Lanjutan

<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>.</p>
--	---

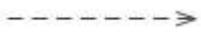
#### 4. Class Diagram

Diagram kelas yaitu menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun system

Tabel 2.4 Class Diagram

Simbol	Deskripsi
<p>Kelas</p> 	<p>Kelas pada struktur sistem.</p>
<p>Antarmuka / <i>interface</i></p> 	<p>Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.</p>
<p><i>Association</i></p> 	<p>Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>
<p><i>Asosiasi berarah / directed association</i></p> 	<p>Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>

**Table 2.4** Lanjutan

<p><i>Generalization</i></p> 	<p>Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).</p>
<p>Kebergantungan / <i>dependency</i></p> 	<p>Relasi antar kelas dengan makna kebergantungan antar kelas.</p>
<p>Agregasi / <i>aggregation</i></p> 	<p>Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>).</p>

#### 2.2.4. Aplikasi

Menurut (Juansyah, 2015:1-8), Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus *computer eksekutif*, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang di harapkan.

### 2.2.5. Java

Menurut (Lengkong, Sinsuw, & Lumenta, 2015:18-25), Java adalah bahasa berorientasi objek yang dapat digunakan untuk pengembangan aplikasi mandiri, aplikasi berbasis internet, serta aplikasi untuk perangkat-perangkat cerdas yang dapat berkomunikasi lewat internet atau jaringan komunikasi. Dalam Java ada 2 (dua) jenis program berbeda, yaitu aplikasi dan *applet*. Aplikasi adalah program yang biasanya disimpan dan dieksekusi dari komputer lokal sedangkan *applet* adalah program yang biasanya disimpan pada komputer yang jauh, yang dikoneksikan pemakai lewat *web browser*.

Menurut (Rofiq & Putri, 2017:171-178), Java adalah bahasa pemrograman serbaguna, dapat digunakan untuk membuat suatu program sebagaimana dapat dibuat dengan bahasa pemrograman lain seperti pascal atau C++. Java mendukung sumber daya internet *World Wide Web* atau yang biasa disebut dengan *web*, mendukung aplikasi klient/*server*, baik dalam jaringan local (LAN) maupun jaringan (WAN).

Dalam sejumlah literatur disebutkan bahwa Java merupakan hasil perpaduan sifat dari sejumlah Bahasa pemrograman, yaitu C, C++, Object-C, dan sebagainya. Selain itu Java dilengkapi dengan unsur keamanan, yang tak kalah penting adalah bahwa Java menambahkan paradigma pemrograman sederhana, jika mengenal bahasa pemrograman C atau C++, yang mengandalkan pointer dan merasakan kerumitannya, Java justru meninggalkannya, sehingga akan diperoleh kemudahan saat menggunakannya.

### **2.2.6. JavaScript**

Menurut (Rofiq & Putri, 2017:171-178), *Java Script* adalah bahasa pemrograman yang dapat memegang kontrol aplikasi, berorientasi objek murni dan digunakan secara prosedural. *Java Script* merupakan bahasa yang *case sensitive* seperti halnya bahasa pemrograman Java yaitu membedakan penulisan dengan huruf kecil dan huruf besar memberi arti yang berbeda. Java menggunakan kelas untuk membentuk suatu objek. Sejumlah kelas sudah tersedia dan dapat digunakan dengan mudah, bahkan dapat dikembangkan jauh melebihi konsep pewarisan, yang dimaksud pewarisan adalah sifat yang ada pada bahasa pemrograman berorientasi objek yang memungkinkan sifat-sifat suatu objek diturunkan dengan mudah ke objek lain.

### **2.2.7. Java Development Kit (JDK)**

Menurut (Juansyah, 2015:1-8), *Java Development Kit (JDK)* adalah sekumpulan perangkat lunak yang dapat digunakan untuk mengembangkan perangkat lunak yang berbasis Java, sedangkan JRE adalah sebuah implementasi dari *Java Virtual Machine* yang benar-benar digunakan untuk menjalankan program java. Biasanya, setiap JDK berisi satu atau lebih JRE dan berbagai alat pengembangan lain seperti sumber *compiler java, bundling, debuggers, development libraries* dan lain sebagainya.

### 2.2.8. PHP

Menurut (Sulihati & Andriyani, 2016:15-26), Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat web yang bersifat *server-side scripting*. PHP memungkinkan kita untuk membuat halaman web yang bersifat dinamis. PHP dapat dijalankan pada berbagai macam *Operating System (OS)*, misalnya *Windows, Linux* dan *Mac OS*. Selain *Apache*, PHP juga mendukung beberapa web server lain, misalnya *Microsoft IIS, Caudium, PWS* dan lain-lain. Seperti pernah disinggung sebelumnya bahwa PHP dapat memanfaatkan *database* untuk menghasilkan halaman web yang dinamis.

Menurut (Rofiq & Putri, 2017:171-178), PHP (*Hypertext Preprocessor*) pemrograman *interpreter* yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan, disebut juga sebagai pemrograman *Server Slide Programing*, hal ini dikarenakan seluruh prosesnya dijalankan oleh *server*. PHP adalah program yang dikembangkan secara bersama oleh para programmer dari seluruh dunia yang menekuni dunia *open source*, PHP dikembangkan khususnya untuk mengakses dan memanipulasi data yang ada di *database server open source* seperti MySQL karena memiliki tingkat kompatibilitas yang sangat baik.

### 2.2.9. MySQL

Menurut (Habibie, Isnanto, & Kridalukmana, 2016:50-59), MySQL merupakan sebuah perangkat lunak pembuat dan pengelola basis data yang bersifat *open source* dimana perangkat lunak ini berjalan disemua *platform* baik *Linux* maupun *Windows*. MySQL merupakan program yang dapat digunakan untuk aplikasi *multiuser* (banyak pengguna).

Menurut (Agustian, 2016:76-82), MySQL adalah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. Selain itu, bersifat *Open Source* (tidak perlu membayar untuk menggunakannya) pada berbagai *platform* (kecuali untuk jenis *Enterprise*, yang bersifat komersial). MySQL termasuk jenis RDBMS (*Relational Database Management System*). Itulah sebabnya, istilah seperti tabel, baris dan kolom digunakan pada MySQL. Pada MySQL, sebuah *database* mengandung satu atau lebih sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom

### 2.2.10. HTML

Menurut (Harison & Syarif, 2016:40-50), *HyperText Markup Language* (HTML) adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web Internet dan *formatting hypertext* sederhana yang ditulis kedalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan

kedalam format ASCII normal sehingga menjadi *home page* dengan perintah-perintah HTML. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (*Standard Generalized Markup Language*), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C). HTML dibuat oleh kolaborasi Caillau TIM dengan Berners-lee robert ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga penelitian fisika energi tinggi di Jenewa).

#### **2.2.11. CSS**

Menurut (Agustian, 2016:76-82), CSS adalah kependekan dari *Cascading Style Sheet*. CSS merupakan salah satu kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan atau layout halaman web supaya lebih elegan dan menarik. CSS adalah sebuah teknologi internet yang direkomendasikan oleh *World Wide Web Consortium* atau W3C pada tahun 1996. Awalnya, CSS dikembangkan di SGML pada tahun 1970, dan terus dikembangkan hingga saat ini. CSS telah mendukung banyak bahasa markup seperti HTML, XHTML, XML, SVG (*Scalable Vector Graphics*) dan Mozilla XUL (*XML User Interface Language*).

Menurut (Harmadya, AryaSasmita, & Wirdiani, 2015:577-588), *Cascading Style Sheet* (CSS) merupakan salah satu bahasa pemrograman web untuk mengatur beberapa komponen atau elemen dalam sebuah web sehingga lebih terstruktur dan

seragam dengan HTML atau script. CSS memiliki 3 versi, yaitu CSS1, CSS2 dan CSS3.

### **2.2.12. Android Studio**

Menurut (Juansyah, 2015:1-8), Android studio adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi Android dan bersifat *open source* atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 Mei 2013 pada *event Google I/O Conference* untuk tahun 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android.

Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT plugin (*Android Development Tools*). Android studio memiliki fitur:

- a. Projek berbasis pada *Gradle Build*
- b. Refactory dan pembenahan bug yang cepat
- c. Tools baru yang bernama “Lint” dikalim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
- d. Mendukung Proguard And App-signing untuk keamanan.
- e. Memiliki GUI aplikasi android lebih mudah
- f. Didukung oleh Google Cloud Platfrom untuk setiap aplikasi yang dikembangkan.



**Gambar 2.2** Android Studio

### **2.2.13. Hybrid Application**

Menurut (Harmadya et al., 2015:577-588), Intuisi dari aplikasi *hybrid* adalah menanamkan aplikasi *mobile* HTML5 ke dalam konten aplikasi *native*. Aplikasi *hybrid* merupakan kombinasi dari kelebihan yang dimiliki antara aplikasi web dan aplikasi *native*. Aplikasi *hybrid* mengkonversi aplikasi web *mobile* HTML5 ke aplikasi *native smartphone* yang ditargetkan. Konversi aplikasi web *mobile* ke dalam aplikasi *native* memerlukan dukungan perangkat lunak semacam *framework* pengembangan aplikasi *mobile*.

Kelebihan aplikasi *hybrid* adalah tahap pengembangannya yang relatif mudah karena memanfaatkan standar teknologi web. Aplikasi *hybrid* dapat berjalan pada hampir semua *platform mobile* yang dituju, artinya aplikasi *hybrid* dapat didistribusikan secara aplikasi *native* menuju ke *platform mobile* yang ditargetkan oleh pengembang. Distribusi aplikasi yang secara *native* memungkinkan aplikasi *hybrid* untuk dapat mengakses fitur-fitur perangkat keras seperti kamera, wifi, kalender dan file.

Menurut (Firdausy, Agus, & Astuti, 2017:30-37), *Hybrid application* adalah teknologi menggabungkan dua bahasa pemrograman atau dua teknologi menjadi satu, dengan adanya teknologi ini akan memperingan pekerjaan *developer* dalam mengembangkan aplikasinya ke segala jenis sistem operasi *smartphone*.

#### **2.2.14. Ionic Framework**

Menurut (SOPINGI, 2016:1-17), *Ionic* adalah kerangka ponsel HTML5 dengan fokus pada kinerja yang memanfaatkan akselerasi *hardware* dan tidak memerlukan pihak ketiga seperti *JavaScript library*. *Ionic* bekerja bersama-sama dengan *AngularJs* untuk membangun *user interface* aplikasi yang interaktif hanya dengan satu halaman. Aplikasi yang menggunakan *Ionic* biasa disebut aplikasi *hybrid*, yang pada dasarnya ialah website kecil yang berjalan di *shell browser* yaitu sebuah aplikasi yang memiliki akses ke lapisan *platform* asli dari sebuah *device*. Aplikasi *hybrid* memiliki banyak manfaat jika dibandingkan dengan aplikasi *native*, khususnya dalam hal mendukung *platform* dan kecepatan pengembangan.

Menurut (Rofiq & Putri, 2017:171-178), *Framework Ionic* adalah sekumpulan teknologi yang dikembangkan untuk membangun aplikasi *mobile hybrid* yang *powerful*, cepat, mudah dan juga memiliki tampilan yang menarik. *Ionic* menggunakan *AngularJS* sebagai *framework* berbasis web dan menggunakan *Cordova* untuk membangun aplikasi *mobile*. *Ionic Framework* merupakan *framework* HTML5 yang masih baru, dirilis 2 tahun lalu. *Framework* ini membantu Anda dalam mengembangkan aplikasi *mobile* dengan teknologi web seperti HTML, CSS dan *Javascript*, *Ionic*.

### **2.2.15. Android Virtual Device (Emulator)**

Menurut (Sinsuw & Najoan, 2013:1-10), *Android Virtual Device* merupakan *emulator* untuk menjalankan aplikasi android. Setiap AVD terdiri dari sebuah profil perangkat keras yang dapat mengatur pilihan untuk menentukan fitur *hardware emulator*. Misalnya, menentukan apakah menggunakan perangkat kamera, apakah menggunakan keyboard QWERTY fisik atau tidak, berapa banyak memori *internal*, dan lain-lain. AVD juga memiliki sebuah pemetaan versi Android, maksudnya kita menentukan versi dari *platform* Android akan berjalan pada *emulator*. Pilihan lain dari AVD, misalnya menentukan *skin* yang kita ingin gunakan pada *emulator*, yang memungkinkan untuk menentukan dimensi layar, tampilan, dan sebagainya. Kita juga dapat menentukan *SD Card Virtual* untuk digunakan dengan di *emulator*.

### **2.2.16. Android SDK (Software Development Kit)**

Menurut (Sinsuw & Najoan, 2013:1-10), Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Beberapa fitur-fitur Android yang paling penting adalah mesin *Virtual Dalvik* yang dioptimalkan untuk perangkat *mobile*, *integrated browser* berdasarkan *engine open source WebKit*, Grafis yang dioptimalkan dan didukung oleh *libraries grafis 2D*, grafis 3D berdasarkan spesifikasi *opengl ES 1.0* (Opsional akselerasi perangkat keras), kemudian *SQLite* untuk penyimpanan data (*database*). Fitur-fitur android

lainnya termasuk media yang mendukung audio, video, dan gambar, juga ada fitur *bluetooth*, EDGE, 3G dan WiFi, dengan fitur kamera, GPS, dan kompas.

### **2.2.17. Android**

Menurut (Lengkong et al., 2015:18-25), Android merupakan subset perangkat lunak untuk perangkat *mobile* yang meliputi sistem operasi, *middleware*, dan aplikasi inti yang dirilis oleh Google. Android adalah sistem operasi bergerak (*mobile operating system*) yang mengadopsi sistem operasi Linux, namun telah dimodifikasi. Android diambil alih oleh Google pada tahun 2005 dari Android, Inc sebagai bagian strategi untuk mengisi pasar sistem operasi bergerak. Google mengambil alih seluruh hasil kerja Android termasuk tim yang mengembangkan Android.

#### **2.2.17.1. Fitur Android**

Menurut (Lengkong et al., 2015:18-25), fitur ini mendukung penggantian komponen dan penggunaan kembali komponen yang sudah dibuat (*reusable*). Seperti pada umumnya, *framework* memiliki keuntungan dalam proses pengkodean karena kita tidak perlu membuat kodingan untuk hal-hal yang pasti dilakukan seperti kodingan menampilkan gambar, kodingan konek database, dan lain-lain.

1. Mesin *Virtual Dalvik*

Lingkungan dimana aplikasi android akan bekerja.

2. *Integrated Browser*

Berdasarkan *Open Source* engine *WebKit*.

3. *Grafis*

Dengan adanya fitur ini, kita bisa membuat aplikasi grafis 2D dan 3D karena Android memiliki *library OpenGL ES 1,0*.

4. *SQLite*

Tugas dari fitur ini adalah berperan dalam penyimpanan data. Bahasanya mudah dimengerti dan merupakan sistem *database* nya android.

5. *Media Support*

Fitur yang mendukung audio, video dan gambar.

6. *GSM Telephony*

Tidak semua android punya fitur ini karena fitur ini tergantung dari *smartphone* yang dimiliki.

7. *Bluetooth, EDGE, 3G, WiFi*

Fitur ini tidak selalu tersedia pada android karena tergantung *hardware* atau *smartphone*.

8. Dukungan Perangkat Tambahan

Android dapat memanfaatkan kamera, layar sentuh, *accelerometer*, *magnetometers*, GPS, akselerasi 2D, dan Akselerasi 3D.

9. *Multi-Touch*

Kemampuan layaknya *handset* modern yang dapat menggunakan dua jari atau lebih untuk berinteraksi dengan perangkat.

#### 10. Lingkungan *Development*

Memiliki fitur *emulator*, *tools*, untuk *debugging*, profil dan kinerja memori dan *plugin* untuk IDE Eclipse.

#### 11. *Market*

Seperti kebanyakan handphone yang memiliki tempat penjualan aplikasi, *Market* pada android merupakan katalog aplikasi yang dapat di *download* dan di *install* pada handphone melalui internet.

### 2.2.17.2. Versi-versi Android

Menurut (Lengkong et al., 2015:18-25), ada beberapa versi android sebagai berikut:

#### 1. *Android 1.5 Cupcake*

*Cupcake* dirilis 30 April 2009. *Cupcake* menjadi versi android pertama yang menggunakan nama makanan. Konon katanya versi ini seharusnya versi 1.2, namun Google memutuskan untuk membuat revisi besar dan membuatnya menjadi versi *1.5 Cupcake* adalah kue kecil yang dipanggang dalam cetakan berbentuk cup.

#### 2. *Android 1.6 Donut*

*Android V1.6, codename Donut*, dirilis pada 15 September 2009. Pada versi ini diperbaiki beberapa kesalahan *reboot*, perubahan fitur foto dan video dan integrasi pencarian yang lebih baik. Donat merupakan panganan berbentuk cincin. Bulat bolong tengah. Adonan donat dimasak dengan cara digoreng dan biasanya disajikan dengan topping di atasnya.

3. *Android 2.0/2.1 Eclair*

*Android 2.0/2.1 Eclair* dirilis 26 Oktober 2009. *Eclair* adalah makanan penutup yakni kue yang biasanya berbentuk persegi panjang yang dibuat dengan krim di tengah dan lapisan cokelat di atasnya.

4. *Android 2.2 Froyo*

Dirilis 20 Mei 2010. Menggunakan *codename Froyo*, yang merupakan makan penutup yang nama merek sebuah produk yang terbuat dari *Yoghurt*. *Froyo* singkatan dari *Frozen Yoghurt*, *Froyo* adalah *yoghurt* yang telah mengalami proses pendinginan, sehingga secara terlihat sama seperti es krim.

5. *Android 2.3 Gingerbread*

*Android versi 2.3 Gingerbread* dirilis resmi tanggal 6 Desember 2010. *Gingerbread* merupakan jenis kue kering yang dengan rasa jahe. Kue jahe biasanya dibuat pada perayaan hari libur akhir tahun di Amerika. Biasanya cemilan kering ini dicetak berbentuk tubuh manusia.

6. *Android 3.0 Honeycomb*

Dirilis tanggal 22 February 2011. *Honeycomb* adalah sereal sarapan manis yang sudah dibuat tahun 1965 oleh *Posting Sereal*. Seperti namanya, *Honeycomb*/sarang lebah, sereal ini terbuat dari potongan jagung berbentuk sarang lebah dengan rasa madu.

7. *Android 4.0 Ice Cream Sandwich*

*Android 4.0-4.0.2 API Level 14* dan *4.0.3-4.0.4 API Level 15* pertama dirilis 19 Oktober 2011. Dinamai *Ice Cream Sandwich*. *Ice Cream Sandwich* adalah

lapisan es krim, biasanya rasa vanilla yang terjepit di antara dua kue coklat, dan biasanya berbentuk persegi panjang.

8. *Android 4.1 Jelly Bean*

*Android Jelly Bean* diluncurkan pertama kali pada Juli 2012, dengan berbasis *Linux Kernel 3.0.31*. Terdiri dari *Android 4.1 API Level 16*, *Android 4.2 API Level 17*, *Android 4.3 API Level 18*. Penamaan *Jelly Bean* mengadaptasi nama sejenis permen dalam beraneka macam rasa buah. Ukurannya sebesar kacang merah. Permen ini keras di luar tapi lunak di dalam serta lengket bila di gigit.

9. *Android 4.4 KitKat*

*Android 4.4 Kitkat API level 19*. Google mengumumkan *Android KitKat* (dinamai dengan izin dari *Nestle* dan *Hershey*) pada 3 september 2013. Dengan tanggal rilis 31 Oktober 2013. *KitKat* merupakan merk sebuah coklat yang dikeluarkan oleh *Nestle*. Rilis berikutnya setelah nama *KitKat* diperkirakan banyak pengamat akan diberi nomor 5.0 dan dinamai '*Key Lime Pie*'.

### **2.2.18. Definisi Sistem Informasi Geografis (SIG)**

Menurut (Prahasta, 2009) dalam jurnal (Marlena & Aspriyono, 2014:161-167), Sistem informasi geografis adalah informasi yang di rancang untuk untuk bekerja dengan data yang tereferensi secara spasial atau koordinat-koordinat geografis atau dengan kata lain, SIG merupakan sistem basis data dengan kemampuan-kemampuan khusus (terkait) data yang tereferensi secara geografis. SIG dapat di uraikan menjadi beberapa sub-sistem sebagai berikut:

1. *Data Input*: Sub-Sistem ini bertugas untuk mengumpulkan, mempersiapkan, dan menyimpan data yang spasial dan atributnya dari berbagai sumber, sub-sistem ini pula yang bertanggungjawab mengonversikan atau mentransformasikan format-format data aslinya ke dalam format (*native*) yang dapat di gunakan oleh perangkat SIG yang bersangkutan.
2. *Data Output*: Sub-sistem ini bertugas untuk menampilkan atau menghasilkan keluaran (termasuk mengekspornya ke format yang di kehendaki) seluruh atau sebagian basis data (Spasial) baik dalam bentuk *softcopy* maupun *hardcopy* seperti halnya tabel, grafik, *report*, peta dan lain sebagainya.
3. *Data Management*: Sub-sistem ini mengorganisasikan baik data spasial maupun tabel-tabel atribut terkait kedalam sebuah sistem basis data sedemikian rupa hingga mudah dipanggil kembali atau di *retrieve* (di *load* ke memori), di *update*, dan di edit.
4. *Data Manipulation dan Analisis*: Sub-sistem ini menentukan informasi-informasi yang dapat dihasilkan oleh SIG. Selain itu sub-sistem ini jga melakukan manipulasi (Evaluasi dan penggunaan fungsi-fungsi dan operator matematis dan logika) dan pemodelan untuk menghasilkan informasi yang di harapkan.

#### **2.2.18.1. Kemampuan Sistem Informasi Geografis (SIG)**

Menurut (Prahasta, 2009) dalam jurnal (Marlena & Aspriyono, 2014:161-167), Sistem informasi geografis mempunyai kemampuan untuk menghubungkan

berbagai data pada suatu titik tertentu di bumi, menggabungkannya, menganalisis dan akhirnya memetakan hasilnya:

1. Memasukkan dan mengumpulkan data geografis (spasial dan atribut).
2. Mengintegrasikan data geografis.
3. Memeriksa, *mengupdate* (mengedit) data geografis.
4. Menyimpan atau memanggil kembali data geografis.
5. Mempresentasikan atau menampilkan data geografis.
6. Mengelola, memanipulasi dan menganalisis data geografis.
7. Menghasilkan *output* data geografis dalam bentuk peta tematik (*view* dan *layout*), tabel, grafik (*chart*) laporan, dan lainnya baik dalam bentuk *hardcopy* maupun *softcopy*.

#### **2.2.18.2. Komponen Sistem Informasi Geografis**

Menurut (Muryono dkk, 2007) dalam jurnal (Asnawati & Kusuma, 2011:76-102), komponen utama SIG secara garis besar terdiri dari perangkat keras (*hardware*), perangkat lunak (*software*), organisasi, dan pengguna. Kombinasi antara keempat komponen ini akan menentukan keberhasilan suatu aplikasi SIG. Sistem Informasi Geografi (SIG) dapat dibagi menjadi empat komponen yaitu:

1. Perangkat Keras

Perangkat komputer tambahan untuk mendukung kegiatan SIG antara lain monitor untuk menyajikan hasil *digitizer*, dan *scanner* untuk memasukkan

data spasial serta printer dan plotter untuk penyajian data dalam bentuk *hardcopy*.

## 2. Perangkat Lunak SIG

Perangkat lunak SIG meliputi program dan *user interface* untuk menjalankan perangkat keras. Perangkat lunak yang banyak digunakan antara lain *ArcView*, *ArcInfo*, *AutoCAD Map*, *MapInfo*, *IDRISI*, *ArcGIS*, *GRASS*, dan *ILWIS*. Perangkat lunak tersebut digunakan sesuai dengan spesifikasi dan kemampuannya dalam menyelesaikan masalah SIG. *User interface* pada tiap perangkat lunak SIG memiliki daftar menu, graphical icons, dan perintah-perintah.

## 3. Sumber Daya Pengguna

Sumber daya pengguna sama pentingnya dengan perangkat lunak dan perangkat keras. Sumber daya pengguna penting untuk menentukan sasaran dan hasil pembuatan sistem dan memberikan pertimbangan serta alasan penggunaan SIG.

## 4. Infrastruktur

Infrastruktur merujuk pada organisasi, administrasi dan lingkungan budaya yang dibutuhkan untuk operasi SIG. Infrastruktur meliputi ketrampilan, standar data, *clearinghouses*, dan pola organisasi umum.

### **2.2.19. Google Maps API**

Menurut (Mahdia & Noviyanto, 2013:162-171), *Google Maps API* adalah sebuah layanan (*service*) yang diberikan oleh Google kepada para pengguna untuk

memanfaatkan *Google Map* dalam mengembangkan aplikasi. *Google Maps API* menyediakan beberapa fitur untuk memanipulasi peta, dan menambah konten melalui berbagai jenis *services* yang dimiliki, serta mengizinkan kepada pengguna untuk membangun aplikasi *enterprise* di dalam websitenya. Pengguna dapat memanfaatkan layanan-layanan yang ditawarkan oleh *Google Maps* setelah melakukan registrasi dan mendapatkan *Google Maps API Key*. *Google* menyediakan layanan ini secara gratis kepada pengguna di seluruh dunia.

#### **2.2.20. Definisi Pemetaan**

Menurut (Harison & Syarif, 2016:40-50), Istilah pemetaan seringkali digunakan pada ilmu matematika untuk menunjukkan proses pemindahan informasi dari satu bentuk ke bentuk lainnya. Proses tersebut serupa dengan yang dilakukan oleh *kartografer*, yaitu memindahkan informasi dari permukaan bumi ke dalam kertas. Hasil dari pemindahan informasi tersebut dinamakan peta atau *map*.

##### **2.2.20.1. Proses Pemetaan**

Menurut (Permanasari, 2007) dalam jurnal (Koniyo, 2013:17-26), terdapat tiga tahapan pemetaan yang harus dilakukan yaitu :

#### **1. Tahap Pengumpulan Data**

Langkah awal dalam proses pemetaan dimulai dari pengumpulan data.

Data merupakan suatu bahan yang diperlukan dalam proses pemetaan.

Keberadaan data sangat penting artinya, dengan data seseorang dapat melakukan analisis evaluasi tentang suatu data wilayah tertentu. Data yang

dipetakan dapat berupa data *primer* atau data *sekunder*. Data yang dapat dipetakan adalah data yang bersifat spasial, artinya data tersebut terdistribusi atau tersebar secara keruangan pada suatu wilayah tertentu. Pada tahap ini data yang telah dikumpulkan kemudian dikelompokkan dahulu menurut jenisnya seperti kelompok data kualitatif atau data kuantitatif.

Pengenalan sifat data sangat penting untuk simbolisasi atau penentuan dan pemilihan bentuk simbol, sehingga simbol tersebut akan mudah dibaca dan dimengerti. Setelah data dikelompokkan dalam tabel-tabel, sebelum diolah ditentukan dulu jenis simbol yang akan digunakan. Untuk data kuantitatif dapat menggunakan simbol batang, lingkaran, arsir bertingkat dan sebagainya, melakukan perhitungan-perhitungan untuk memperoleh bentuk simbol yang sesuai.

## 2. Tahap Penyajian Data

Langkah pemetaan kedua berupa penyajian data. Tahap ini merupakan upaya melukiskan atau menggambarkan data dalam bentuk simbol, supaya data tersebut menarik, mudah dibaca dan dimengerti oleh pengguna (*users*). Penyajian data pada sebuah peta harus dirancang secara baik dan benar supaya tujuan pemetaan dapat tercapai.

## 3. Tahap Penggunaan Peta

Tahap penggunaan peta merupakan tahap penting karena menentukan keberhasilan pembuatan suatu peta. Peta yang dirancang dengan baik akan dapat digunakan atau dibaca dengan mudah. Peta merupakan alat untuk melakukan komunikasi, sehingga pada peta harus terjalin interaksi antar

pembuat peta (*map maker*) dengan pengguna peta (*mapusers*). Pembuat peta harus dapat merancang peta sedemikian rupa sehingga peta mudah dibaca, diinterpretasi dan dianalisis oleh pengguna peta. Pengguna harus dapat membaca peta dan memperoleh gambaran informasi sebenarnya dilapangan (*real world*).

#### **2.2.21. Location Based Service (LBS)**

Menurut (Lengkong et al., 2015:18-25), Teknologi *Location Based Service* (LBS) merupakan salah satu bagian dari implementasi *mobile* gis yang lebih cenderung memberikan fungsi terapan sehari-hari seperti menampilkan direktori kota, navigasi kendaraan, pencarian alamat serta jejaring sosial dibanding fungsionalitas pada teknologi GIS populer untuk *Field Based GIS*. Dua unsur utama LBS adalah :

1) *Location Manager (API Maps)*

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

2) *Location Providers (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 *android.location*. Dengan *Location Manager*, kita dapat menentukan lokasi kita saat ini dan rute menuju tempat tertentu.

### **2.2.22. Global Positioning System (GPS)**

Menurut (Mahdia & Noviyanto, 2013:162-171), GPS atau *Global Positioning System*, merupakan sebuah alat atau sistem yang dapat digunakan untuk menginformasikan penggunanya dimana lokasinya berada (secara global) di permukaan bumi yang berbasis satelit. Data dikirim dari satelit berupa sinyal radio dengan data digital. Dimanapun pengguna tersebut berada, maka GPS bisa membantu menunjukkan arah. Layanan GPS ini tersedia gratis. Awalnya GPS hanya digunakan hanya untuk kepentingan militer, tapi pada tahun 1980-an dapat digunakan untuk kepentingan sipil. GPS dapat digunakan dimanapun juga dalam 24 jam. Posisi unit GPS akan ditentukan berdasarkan titik-titik koordinat latitude dan longitude.

### **2.2.23. Anjungan Tunai Mandiri (ATM)**

Menurut (Djumhadi & Fadilah, 2009:48-56), Anjungan Tunai Mandiri/*Automatic Teller Machine* (ATM) merupakan salah satu produk perbankan yang berbentuk seperangkat alat elektronik yang di sediakan untuk melayani transaksi perbankan yang sangat memasyarakat saat ini. Salah satu dari *electronic delivery channel* ini sudah menjadi ujung tombak/andalan perusahaan yang bergerak dalam jasa keuangan khususnya di bidang perbankan. Perangkat elektronik tersebut dapat melayani berbagai transaksi perbankan dari nasabah bank seperti pengambilan uang

tunai, pengecekan saldo rekening tabungan/giro, transfer dana dan pembayaran tagihan tanpa perlu dilayani oleh petugas bank. Kelangsungan operasi ATM dalam melayani transaksi nasabah harus terpelihara dengan sebaik-baiknya. Keberadaan, kelancaran dan keberagaman transaksi ATM dapat memberikan image yang baik khususnya masyarakat yang menjadi nasabah perbankan.

### 2.3. Penelitian Terdahulu

Penelitian terdahulu dari penelitian ini adalah sebagai berikut:

**Tabel 2.5** Daftar Penelitian Terdahulu

No	Judul Penelitian	Nama Peneliti	Hasil Penelitian
1.	Rancang Bangun Mobile GIS ( <i>Geographic Information System</i> ) Pencarian Lokasi ATM BNI di Semarang Pada Media Ponsel Berbasis Android (Susanto & Zulliansyah, 2013:59-63)	Ajib susanto dan Taufiq Zulliansyah	Tujuan pembuatan aplikasi adalah terciptanya aplikasi yang mampu memenuhi kebutuhan nasabah BNI akan informasi lokasi ATM BNI, memudahkan nasabah yang mobile untuk menemukan lokasi ATM BNI di kota Semarang yang ditampilkan dalam bentuk peta digital pada ponsel bersistem operasi android. Metode pengembangan sistem digunakan metode Agile XP dengan fase yaitu explorasi, perencanaan, publikasi, produksi, perawatan dan akhir hidup sistem.

Table 2.5 Lanjutan

2.	Aplikasi Pencarian Informasi dan Lokasi Tempat Makan Pada Perangkat <i>Mobile</i> Berbasis Android (Layona & Yulianto, 2016:9-16)	Rita Layona dan Budi Yulianto	Aplikasi pencari lokasi tempat makan ini informatif dan mudah digunakan oleh pengguna. Fitur dan informasi yang disediakan dalam aplikasi ini cukup membantu pengguna saat mencari tempat makan. Aplikasi ini memiliki tampilan yang menarik. Aplikasi ini dapat membantu pengguna dalam menemukan lokasi tempat makan yang dipilih menggunakan bantuan rute jalan dari posisi pengguna.
3.	Implementasi <i>Location Based Service</i> Berbasis Android Untuk Mengetahui Posisi <i>User</i> (Anwar, Jaya, & Kusuma, 2014:121-133)	Badrul Anwar, Hendra Jaya, dan Putra Indra Kusuma	Dengan menganalisa dan merancang aplikasi pencarian lokasi <i>user</i> maka tersedia aplikasi yang memberikan informasi seputar keberadaan <i>user</i> . Penentuan jarak di ambil dari <i>longitude</i> dan <i>latitude user</i> . Aplikasi ini hanya dapat menampilkan lokasi <i>user</i> dan aplikasi ini tidak bisa digunakan secara <i>offline</i> .

Table 2.5 Lanjutan

4.	Perancangan Penunjuk Rute Pada Kendaraan Pribadi Menggunakan Aplikasi <i>Mobile GIS</i> Berbasis Android Yang Terintegrasi Pada <i>Google Maps</i> (Lengkong et al., 2015:18-25)	Hendra Nugraha Lengkong, Alicia A.E. Sinsuw, ST., MT. dan Arie S.M Lumenta, ST., MT.	<i>Google Map API</i> dan <i>Google Places API</i> merupakan komponen utama dalam <i>Mobile GIS</i> Penunjuk Rute Kendaraan Pribadi. <i>Global Positioning System</i> (GPS) berperan penting dalam menentukan keakuratan posisi pengguna. Aplikasi ini dapat berjalan di OS versi <i>Jelly Bean</i> dengan baik serta minimum spesifikasi yaitu OS <i>Gingerbread</i> . Dapat memudahkan <i>user</i> (pengguna) dalam mencari jalur/rute yang akan ditempuh.
5.	Sistem Informasi Geografis Letak Lokasi Rumah Sakit dan Apotek Kota Bengkulu Berbasis Android (Marlena & Aspriyono, 2014:161-167)	Deti Marlana dan Hari Aspriyono	Hasil akhir dari penelitian ini adalah berupa aplikasi sistem informasi letak lokasi rumah sakit dan apotek berbasis android. Sistem informasi letak lokasi rumah sakit dan apotek ini mempermudah dinas kesehatan kota dan masyarakat untuk mengetahui letak lokasi Rumah sakit dan apotek di kota Bengkulu.

Table 2.5 Lanjutan

6.	Sistem Informasi Geografis Pemetaan Lokasi ATM BNI Terdekat di Kota Serang Berbasis Android (Nurhadiyan & Hidayatullah, 2017:46-52)	Thoha Nurhadiyan dan A. Syamsul Hidayatullah	Tujuan di buat aplikasi ini agar bisa digunakan oleh semua nasabah BANK BNI dan aplikasi ini mampu memberikan informasi lokasi ATM terdekat dan mengetahui ATM yang sedang rusak. Sistem informasi Geografi ATM BNI diwilayah kota serang dapat diterapkan oleh pengguna android dimanapun dan kapanpun, sehingga dapat memudahkan pengguna untuk mengakses ATM BNI terdekat diwilayah kota serang.
7.	WEB SIG (Sistem Informasi Geografis) untuk Fasilitas Umum (Studi Kasus di Kota Yogyakarta) (Susanti & Andayanti, 2014:373-382).	Erma Susanti dan Dina Andayati	Pengembangan aplikasi Sistem Informasi Geografis (SIG) dalam bentuk web untuk fasilitas umum di suatu wilayah menggunakan <i>Google Maps API</i> memberikan kemudahan dalam mengakses dan menampilkan informasi dalam bentuk <i>spasial</i> . Selanjutnya diharapkan dengan adanya sistem tersebut kemudahan dalam akses informasi lokasi tidak terbatas hanya untuk menampilkan titik lokasi saja.

Table 2.5 Lanjutan

8.	Layanan Berbasis Lokasi dan Sistem Informasi Master Berbasis Peta Google untuk Panduan Tur (Yang & Hsu, 2015:1-19)	Sheng-Yuan Yang, Chun-Liang Hsu	Makalah ini bertujuan untuk mengembangkan layanan berbasis lokasi yang didukung sistem <i>Dr.What-Info</i> , menguasai sistem <i>multi-agen</i> mengenai apa informasinya, menggunakan peta Google dan teknologi pengenalan citra sebagai penyedia informasi pariwisata dan sebagai perencana rute bagi wisatawan. Secara khusus, mengingat situs arkeologi di Kota Taipei Baru, Taiwan, untuk tujuan pengujian, sistem yang disajikan ditunjukkan tidak hanya sebagai penyedia informasi mengenai tempat-tempat wisata populer, tetapi juga sebagai perangkat navigasi GPS berperforma tinggi untuk membimbing pengguna menuju tujuan yang mereka inginkan.
9.	Lokasi Mesin Teller Otomatis Perbankan Berdasarkan Konvolusi (Aldajani & Alfares, 2009:1194-1201)	Mansour A. Aldajani dan Hesham K. Alfares	Tujuannya adalah untuk meminimalkan jumlah ATM untuk memenuhi semua permintaan pelanggan dalam wilayah geografis tertentu. Akhirnya, hasil simulasi menunjukkan keefektifan algoritma ini dalam memecahkan masalah penempatan ATM.