

## BAB II LANDASAN TEORI

### 2.1 Tinjauan Studi

Dalam penelitian ini menggali informasi dan penelitian sebagai bahan referensi, baik mengenai permasalahan, metode, dan hasil fungsinya untuk memudahkan peneliti dalam melakukan penelitian sesuai seperti temanya dan membuat sistem yang baru dan bermanfaat. Pada tiga jurnal yang peneliti gunakan sebagai bahan referensi untuk acuan mengambil metode yang akan digunakan.

Menurut Nurul Azwanti (2017) pada jurnal yang berjudul “Perancangan E-Voting berbasis Web”. Pada Perumahan Taman Cipta Asri Blok Cemara RT 08 pemilihan ketua RT sebelumnya sudah pernah dilaksanakan. Tetapi, ketidaktahuan masyarakat terhadap jumlah persediaan kertas suara, menimbulkan pertanyaan karena dianggap terlalu tergesa-gesa dan tidak transparan. Hanya pada saat perhitungan suara seluruh masyarakat diundang oleh panitia pelaksana untuk hadir di salah satu rumah milik warga. Tidak layaknya perhitungan secara konvensional, *electronic voting* atau e-voting memiliki kelebihan yang bisa dipertimbangkan untuk mengatasi masalah pada perhitungan secara konvensional di antaranya, keakuratan, kenyamanan, dan mudah digunakan. Pemodelan yang digunakan di antaranya, *Use Case Diagram*, *Class Diagram*, *Activity Diagram*, dan juga *Deployment Diagram*. Dengan dukungan web memberikan kemudahan akses untuk warga dalam mencari informasi yang berkaitan dengan kandidat calon ketua RT. E-voting adalah sistem yang dirancang dapat mengurangi permasalahan proses pemilihan yang sebelumnya diketahui dalam bentuk kertas, meminimalkan kecurangan, serta memberikan alat media penampung data dengan menggunakan database bagi penyimpanan data warga. Selain itu semua proses pemungutan sampai perhitungan suara dilakukan secara komputerisasi oleh karena itu bisa terlaksana dengan cepat dan tepat[1].

Menurut Harma Oktafia Lingga Wijaya (2017) pada jurnal yang berjudul “E-Voting berbasis Website pada Pemilihan Kades di Rantau Jaya (Lake) dengan Keamanan Data Menggunakan *Enkripsi Base*”. Sistem Informasi dipakai untuk

mempermudah dalam pengelolaan serta penyimpanan data karena bisa menghasilkan informasi dengan tepat dan akurat. Bentuk informasi dan data saat ini dapat dibangun sesuai apa yang kita harapkan atau inginkan, dan banyak karakteristik yang dapat dimanfaatkan untuk mengembangkannya. Dari keterangan di atas, untuk mengatasi perkembangan teknologi yang ada maka perlu dibuat sistem informasi berbasis website, diharapkan dapat mempermudah penyampaian informasi dan meminimalisir terjadinya kecurangan, e-voting lebih memfokuskan pada proses penggunaan perangkat elektronik dalam mendukung kelangsungan proses serta model otomatisasi yang dapat memungkinkan campur tangan dengan minimal oleh individu atau seseorang dalam semua prosesnya. Sistem pemilihan di desa ini masih memakai sistem pemilihan konvensional, proses penghitungan suara dengan cara pemilihan konvensional tersebut masih banyak memiliki kelemahan di antaranya adalah pemilih keliru dalam hal menandai pada kertas suara, maka kebijakan keabsahan penandaan tersebut kurang jelas. Oleh karena itu, banyak jumlah suara yang dinyatakan tidak sah, kemudian yang kedua yaitu pada pengumuman hasil suara pemilihan yang lama karena harus melalui perhitungan manual dan seringnya terjadi kecurangan, serta keamanan data penting yang berada di dalam sistem karena pemilihan ini masalah krusial di masyarakat oleh karena itu dibutuhkan pengamanan data terhadap hasil pemilihan tersebut. Maka dalam keamanan data, peneliti menggunakan cara *enkripsi base 64*[2].

Menurut Yusfar Ilhaqul Choer dan Dede Kurniadi pada jurnal yang berjudul “Rancang Bangun *Electronic Voting* Pemilihan Kepala Daerah Kabupaten Garut”. Di Kabupaten Garut sistem pemilihan kepala daerahnya masih menggunakan cara konvensional. Sistem pemilihan konvensional mempunyai kelemahan-kelemahan yaitu sulitnya membuktikan keabsahan surat suara, dan proses penghitungan suara yang lama. Dari permasalahan di atas maka peneliti mengajukan usul untuk merancang dan membangun suatu sistem untuk melaksanakan pemilihan Kepda atau kepala daerah dengan menggunakan teknologi informasi yaitu E-voting atau *Electronic Voting*. Metodologi yang dipakai untuk pembuatan sistem e-voting ini menggunakan *Unified Approach* yang memiliki beberapa tahapan yaitu *Object Oriented Analysis*, dan *Object Oriented Design*, serta implementasi program memakai *Component Based Development*. Ada pun keluaran atau *outputnya* dari

penelitian ini yaitu menghasilkan aplikasi pemilihan kepala daerah(Kepda) Kabupaten Garut, yang bisa dibuat alternatif implementasi pemilihan kepala daerah(Kepda) Kabupaten Garut secara elektronik[3].

Dari hasil penelitian tersebut, peneliti ingin merancang dan mengimplementasikan aplikasi e-voting berbasis android yang bisa memberikan kemudahan dalam pengelolaan dan juga penyimpanan data yang bisa menghasilkan suatu sistem informasi dengan tepat dan akurat.

## 2.2 Tinjauan Pustaka

### 2.2.1 E-voting

Istilah e-voting (*electronic voting*) atau pemungutan suara elektronik bisa dipahami sebagai proses pemungutan atau pengambilan suara yang memungkinkan pemilih dalam memberikan suara dengan aman serta rahasia melalui media internet. E-voting adalah sistem yang menggunakan perangkat elektronik yang dapat mengolah media informasi digital untuk pembuatan surat suara, dan memberikan suara, serta menghitung hasil atau perolehan suara, menampilkan perolehan jumlah suara dan memelihara kemudian menghasilkan jejak audit[1].

### 2.2.2 Sejarah Android

Android adalah sistem operasi berbasis Linux yang dimodifikasi untuk perangkat bergerak (*mobile device*) yang terdiri dari sistem operasi, *middleware*, dan aplikasi-aplikasi utama. Awalnya, Android dikembangkan oleh Android Inc. Perusahaan ini kemudian dibeli oleh Google pada tahun 2005. Sistem operasi Android kemudian diluncurkan bersamaan dengan dibentuknya organisasi *Open Handset Alliance* tahun 2007. Selain Google, beberapa nama-nama besar juga ikut serta dalam *Open Handset Alliance*, antara lain Motorola, Samsung, LG, Sony Ericsson, T-Mobile, Vodafone, Toshiba, dan Intel[4].

Tabel 2.1 Versi Android

No.	Nama Kode	No. Versi	Tanggal Rilis	API Level
1	Alpha	-	Awal Tahun 2007	-
2	Beta	-	November 2007	-

3	Angel Cake	1.0	23 September 2008	1
4	Bettenberg	1.1	9 Februari 2009	2
5	Cupcake	1.5	27 April 2009	3
6	Donut	1.6	15 September 2009	4
7	Eclair	2.0 – 2.1	26 Oktober 2009	5 – 7
8	Froyo	2.2 – 2.2.3	20 Mei 2010	8
9	Gingerbread	2.3 – 2.3.7	6 Desember 2010	9 – 10
10	Honeycomb	3.0 – 3.2.6	22 Februari 2011	11 – 13
11	Ice Cream Sandwich	4.0 – 4.0.1	18 Oktober 2011	14 – 15
12	Jelly Bean	4.1 – 4.3.1	9 Juli 2012	16 – 18
13	Kitkat	4.4 – 4.4.4	31 Oktober 2013	19 – 20
14	Lollipop	5.0	15 Oktober 2014	21 – 22
15	Marshmallow	6.0	5 Oktober 2015	23
16	Nougat	7.0 – 7.1.2	22 Agustus 2016	24 – 25
17	Oreo	8.0 – 8.1	21 Maret 2017	26 – 27
18	Pie	9.0	7 Maret 2019	28

### 2.2.3 *Android Studio*

*Android Studio* merupakan sebuah IDE atau *Integrated Development Environment* bagi pengembangan aplikasi android, aplikasi ini dipublikasikan oleh *Google* pada tanggal 16 Mei 2013 dan tersedia secara gratis oleh lisensi *Apache 2.0*, *Android Studio* ini menggantikan *software* pengembangan *android* sebelumnya yaitu *Eclipse*[5].

Jendela utama pada *Android Studio* terdiri beberapa letak logikanya yang diidentifikasi sebagai berikut:

1. ***Toolbar***: memungkinkan *developer* melakukan beberapa jenis tindakan, salah satunya menjalankan aplikasi *Android* dan fitur-fitur yang ingin diluncurkan.

2. **Menu navigasi:** membantu *developer* bernavigasi antara proyek dan membuka berkas untuk diubah atau di-*edit*. Menu ini memberikan tampilan yang simpel pada jendela utama proyek.
3. **Jendela editor:** di mana *developer* dapat membuat dan mengubah kode. Sesuai jenis berkas, editor ini bisa berubah. Seperti saat editor menampilkan *Layout Editor*, atau saat akan menampilkan berkas tata letak.
4. **Panel jendela fitur:** terletak di samping luar jendela IDE. Berisikan tombol yang memungkinkan *developer* memperlebar atau mengecilkan jendela fitur.
5. **Jendela fitur:** memberi *developer* akses ke suatu perintah tertentu seperti halnya pengelolaan proyek, pencarian/penelusuran, kontrol versi, dan sebagainya.
6. **Status bar:** menampilkan status proyek dan IDE, serta segala pesan atau peringatan.

#### 2.2.4 *Android Software Development Kit (Android SDK)*

Android SDK adalah tools *Application Programming Interface (API)* yang dibutuhkan untuk memulai pembangunan atau pengembangan aplikasi pada platform Android memakai bahasa pemrograman *Java*. Pada Android SDK terdiri dari *libraries, debugger, handset emulator, dokumentasi, kode contoh, dan tutorial*. SDK memungkinkan pengembang membangun aplikasi diperuntukan platform Android SDK, Android mencakup proyek contoh atau sampel dengan perangkat pengembangan, kode sumber, emulator, dan perpustakaan (*library*) yang dibutuhkan untuk membuat aplikasi Android. Aplikasi yang ditulis menggunakan pemrograman *Java* kemudian dijalankan di *Dalvik, virtual machine* yang dirancang khusus menggunakan *embedded* yang dijalankan di atas kernel *Linux*[5].

#### 2.2.5 *Java Development Kit*

*Java Development Kit* atau *JDK* adalah Paket fungsi *Application Programming Interface* dengan pemrograman *Java*, meliputi *Java Virtual Machine (JVM)* dan *Java Runtime Environment (JRE)*[5]. *JDK* terdiri dari lingkungan yang dijalankan program berada di atas *Operating System (OS)*,

sebagaimana diperlukannya oleh para programmer untuk meng-*compile*, memperbaiki *bug(s)* yang ada dan menjalankan beberapa tambahan dari program-program intinya yang ditulis menggunakan pemrograman Java. Teknologi pemrograman Java memiliki tiga aspek penting yaitu: *application programming interface*, *programming-language spesification*, dan *virtual-machine spesification*. Sebuah JDK dibutuhkan untuk pembangunan atau pengembangan fungsionalitas Java, namun tidak untuk pembangunan atau pengembangan di bahasa pemrograman yang lain. JDK yang dapat dipakai untuk membuat program-program Android yaitu JDK 5 dan JDK 6 atau versi terbarunya[6]. JDK harus ter-*install* pada komputer untuk melakukan proses pembangunan aplikasi berbasis java, tetapi tidak harus ter-*install* di komputer yang menjalankan aplikasi yang sedang dibangun dengan java.

#### 2.2.6 *Kotlin*

*Kotlin* merupakan bahasa pemrograman yang berbasis *Java Virtual Machine(JVM)* yang dikembangkan oleh *JetBrains*[7]. *Kotlin* juga merupakan bahasa pemrograman pragmatis untuk android yang mengkombinasikan dengan *Object Oriented(OO)* dan pemrograman fungsional. *Kotlin* juga bahasa pemrograman yang interoperabilitas yang membuat bahasa ini bisa digabungkan dalam satu proyek dengan bahasa pemrograman *Java*. Bahasa pemrograman ini juga dapat digunakan atau dipakai untuk pembangunan atau pengembangan aplikasi berbasis desktop, web dan bahkan untuk *backend*. Keuntungan-keuntungan yang mungkin bisa didapatkan jika pembangunan atau pengembangan aplikasi beralih menggunakan *Kotlin* untuk mengembangkan aplikasi di atas platform JVM adalah sebagai berikut:

- Bisa mengatasi `NullPointerException` yang pada umumnya terdapat pada *Java*.
- Penulisan kode lebih ringkas, jelas, dan mudah dibaca dibandingkan dari kode yang ditulis dengan memakai bahasa *Java*.
- Mudah dipelajari.
- Dukungan IDE yang mempermudah dalam pemrograman.

### 2.2.7 *Firebase*

*Firebase* adalah *Backend as a Service*(BaaS) yang saat ini dimiliki oleh *Google*. *Firebase* juga merupakan solusi yang ditawarkan dari *Google* untuk mempermudah pengembangan aplikasi *mobile*. Dua fitur yang menarik dari *Firebase* yaitu *Firebase Real Time Database* dan *Firebase Remote Config*. Selain itu, terdapat fitur-fitur pendukung aplikasi yang memerlukan *push notification* adalah *Firebase Notification Console* merupakan salah satu fiturnya.

*Firebase Database* merupakan suatu penyimpanan basis data non-SQL yang memungkinkan untuk menyimpan beberapa tipe data. Tipe data tersebut antara lain *Long*, *String*, dan *Boolean*. Pada data *Firebase Database* tersebut disimpan pada objek *JSON tree*. Dan tidak seperti halnya basis data SQL, penyimpanan ini tidak ada tabel-tabel dan baris-baris pada basis data non-SQL. Ketika ada penambahan data, maka data tersebut menjadi node-node di struktur *JSON*. Node merupakan simpul yang berisi suatu data dan bisa mempunyai cabang berupa node yang lainnya yang berisi suatu data juga. Proses pengisian data pada *Firebase Database* dikenal juga dengan istilah *push*.

Selain *Firebase Database*, *Firebase* juga menyediakan layanan-layanan lain yang dimanfaatkan pada pengembangan aplikasi ini. Dan layanan tersebut yaitu *Storage*, *Firebase Authentication*, dan *Cloud Messaging*. Pengembangan aplikasi, layanan lain yang digunakan saat pengembangan aplikasi yaitu *Firebase Storage*. Layaknya sebuah penyimpanan awan(*cloud*), *Firebase Storage* memungkinkan para pengembang mengunggah(*upload*) atau mengunduh(*download*) sebuah berkas[8].




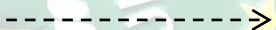


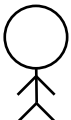
### 2.2.8 *Unified Modelling Language*

*Unified Modeling Language* atau disingkat *UML* yaitu sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan artifacts dari suatu sistem *software*, digunakan untuk memodelkan bisnis, dan sistem-sistem non-*software* yang lain. *UML* terdiri dari *use case diagram*, *activity diagram*, dan *sequence diagram*[9]. Ada jenis-jenis diagram pada *UML* yang biasa digunakan untuk perancangan, di antaranya:

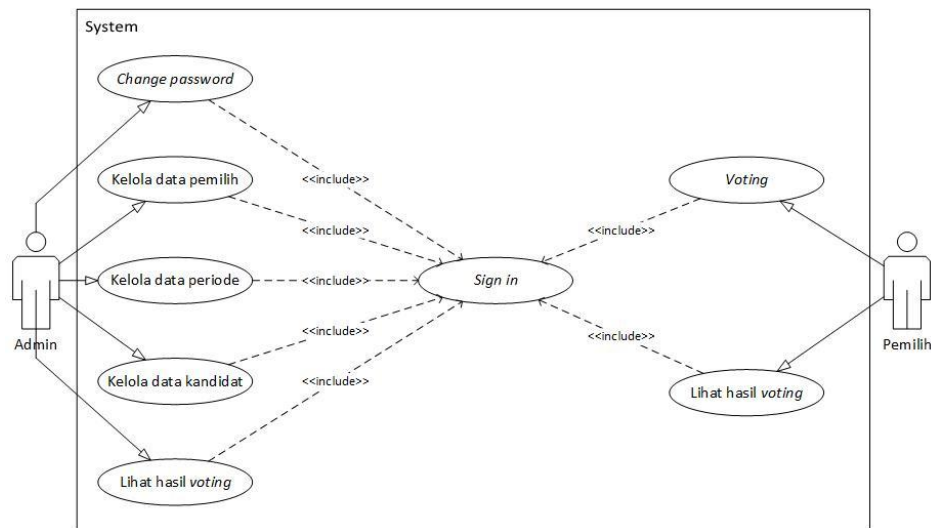
## 1. Use Case Diagram

*Use case diagram* yaitu gambaran atau suatu pola yang menunjukkan kebiasaan atau kelakuan sistem, serta mempersentasikan sebuah interaksi antara aktor dengan sistem[1]. Berikut adalah bentuk simbol dan fungsi dari *use case diagram* tersebut:

Tabel 2.2 Simbol *Use Case Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Use Case</i>	Deskripsi pada urutan aksi yang ditampilkan sistem.
2.		<i>Collaboration</i>	Interaksi elemen lainnya yang bekerja sama menyediakan perilaku yang besar dari jumlah yang ada.
3.		<i>Generalization</i>	Hubungan pada objek anak berbagi perilaku serta struktur data dari objek induk.
4.		<i>Include</i>	Memungkinkan 1 use case menggunakan fungsionalitas yang disediakan oleh use case yang lain.
5.		<i>Extend</i>	Memungkinkan 1 use case secara optimal menggunakan fungsilitas dari <i>use case</i> yang lain.
6.		<i>Association</i>	Menghubungkan antara objek satu ke objek yang lain.
7.		<i>Actor</i>	Pengguna sistem yang berhubungan dengan sistem lain atau waktu.



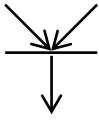
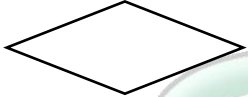

Gambar 2.1 Contoh *Use Case Diagram*

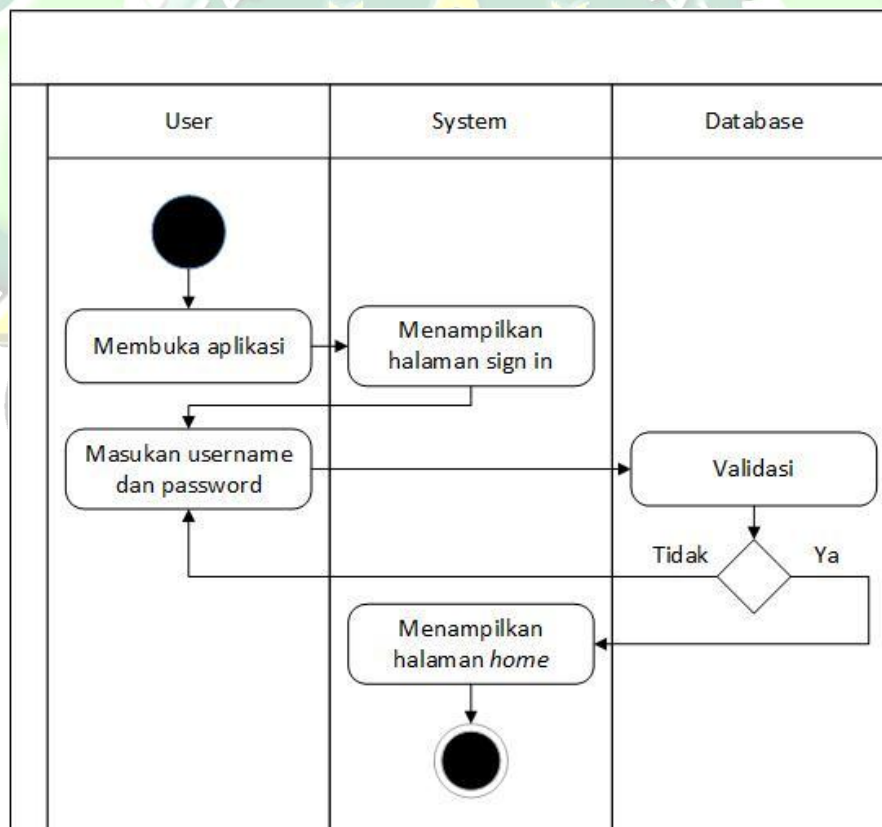
## 2. *Activity Diagram*

*Activity Diagram* menggambarkan berbagai alir-alir aktivitas dalam suatu sistem yang sedang dalam rancangan, bagaimana masing alir itu berawal, decision yang akan mungkin terjadi, serta bagaimana mereka berakhir[1]. Berikut adalah bentuk simbol yang digunakan dalam penggambaran *Activity Diagram*:

Tabel 2.3 Simbol *Activity Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Start point</i>	Merupakan awal dari sebuah aktivitas.
2.		<i>End point</i>	Merupakan akhir dari sebuah aktivitas
3.		<i>Activities</i>	Menggambarkan suatu proses atau kegiatan
4.		<i>Fork</i> (percabangan)	Merupakan symbol yang digunakan untuk menggambarkan kegiatan

			yang dilakukan dengan cara paralel menjadi satu
5.		<i>Join</i> (penggabungan)	Dipakai untuk menunjukkan atau memperlihatkan adanya dekomposisi.
6.		<i>Decision point</i>	Digunakan untuk menentukan pilihan <i>True or False</i> /Benar atau Salah.
7.		<i>Swimlane</i>	Digunakan untuk membagi <i>activity diagram</i> .





Gambar 2.2 Contoh Activity Diagram

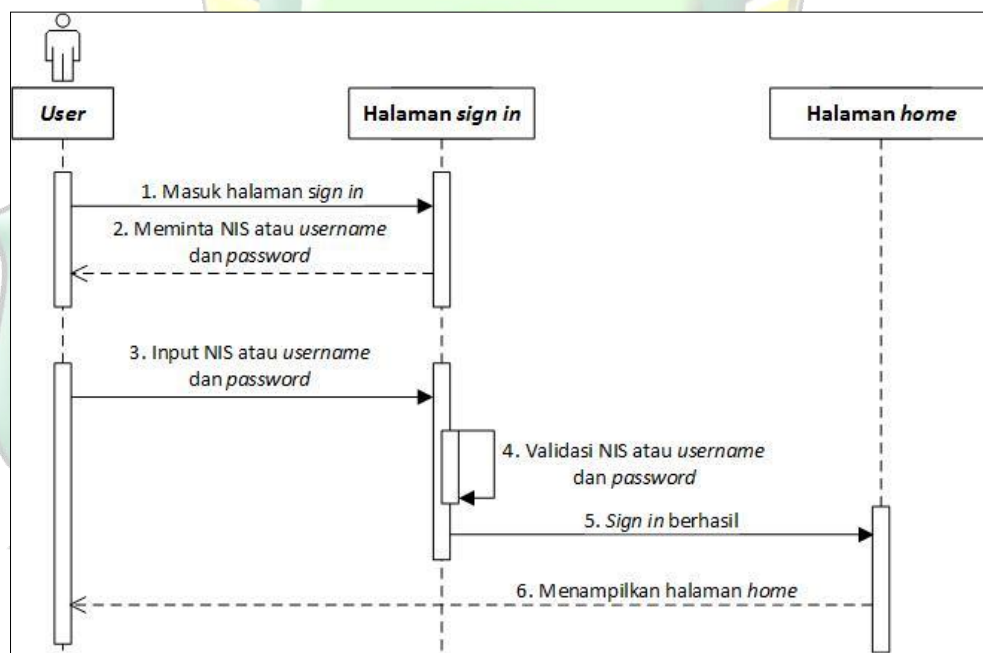
### 3. *Sequence Diagram*

*Sequence Diagram* menggambarkan suatu interaksi antar objek pada dalam dan pada sekitar sistem (termasuk *display*, pengguna, dan sebagainya) dalam bentuk *message* yang dipaparkan atau digambarkan terhadap waktu[1]. Berikut adalah bentuk simbol-simbol yang digunakan untuk penggambaran *Sequence Diagram*:

Tabel 2.4 Simbol *Sequence Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Entity class</i>	Merupakan bagian dari suatu sistem yang berisi tentang kumpulan kelas berupa sebuah entitas yang membentuk suatu gambaran sistem dan menjadi acuan penyusunan basis data.
2.		<i>Boundary class</i>	Merupakan kumpulan class yang menjadi suatu <i>interface</i> atau interaksi di antara satu <i>actor</i> dengan sistem seperti tampilan <i>form</i> .
3.		<i>Control class</i>	Merupakan objek yang berisi suatu logika aplikasi yang pastinya tidak mempunyai tanggung jawab entitas.
4.		<i>Message</i>	Simbol mengirim pesan antar- <i>class</i>
5.		<i>Recursive</i>	Menggambarkan pengiriman pesan kepada dirinya sendiri.

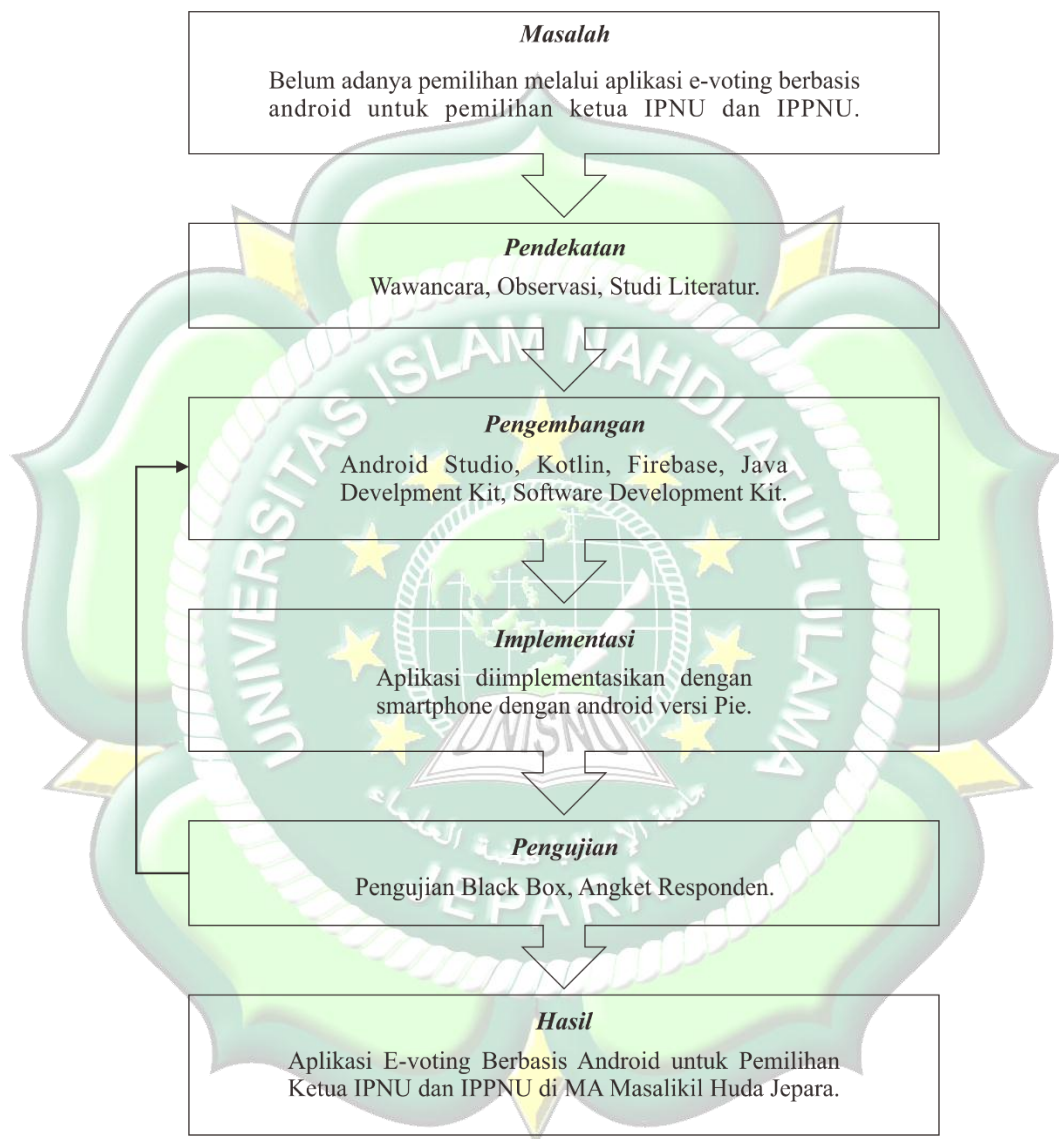
6.		<i>Activation</i>	Mewakili eksekusi pada operasi dari objek, durasi aktivitas operasi berbanding lurus dengan panjang kotak.
7.		<i>Lifeline</i>	Garis titik yang terhubung pada objek, selama <i>lifeline</i> terdapat <i>activation</i> .



Gambar 2.3 Contoh Sequence Diagram

### 2.3 Kerangka Berpikir

Pada tahapan ini, peneliti membuat suatu kerangka pemikiran yang menjadi bentuk gambaran penelitian dari awal hingga akhir dalam melakukan penelitian yang menjelaskan konsep hingga memberikan pandangan terhadap penelitian yang akan dilakukan.



Gambar 2.4 Kerangka Berpikir