

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Tinjauan Studi**

Menurut Ali Khumaidi (2018) dalam penelitiannya yang berjudul “Perancangan Aplikasi Marketplace Order Baju Pintar Menggunakan Web Responsif Untuk Memudahkan Customer Mendesain Sesuai Selera” menyatakan bahwa masalah yang dimiliki seseorang ketika ingin memiliki baju yang ukuran, desain serta bahan yang sesuai selera maka mereka harus datang ke butik yang sudah populer dan mahal, namun jika ingin membuat secara mandiri serta mencari bahan untuk bajunya sendiri maka akan membutuhkan waktu yang lama dan biaya yang mahal jika tidak dapat membandingkan harganya dan menguras waktu dan energi yang cukup tinggi. Mencari penjahit juga susah apalagi jika bajunya buru-buru mau dipakai dan penjahit tidak memiliki standar harga, sehingga pengguna jasa tidak dapat memilih dengan leluasa. Melihat masalah tersebut peneliti membuat perancangan aplikasi marketplace yang khusus untuk butik, toko bahan, dan penjahit. aplikasi ini dikembangkan dengan menggunakan GRAPPLE (Guidelines for Rappid Application Engineering). [2]

Menurut Nita Novianti, Ridwan Setiawan (2016), dalam penelitiannya yang berjudul “Pengembangan Sistem Informasi Jasa Menjahit Berbasis *WEB* Pada Ganesha *Tailor* Garut” . Perancangan ini di latar belakang karena proses transaksi masih dilakukan secara manual, data pelanggan tidak terorganisir dengan baik sehingga kesulitan dalam pencarian data pelanggan yang telah lama yang ingin menjahitkan kembali bajunya disini. Oleh karena itu, peneliti membuat sistem informasi berbasis web menggunakan model UML (*Unified Modelling Language*) yang menghasilkan sebuah sistem informasi yang dapat melakukan proses pengolahan data transaksi secara terorganisir. [3]

Menurut Muhammad Faisal Akbar (2011), dalam penelitiannya yang berjudul “Analisis dan Perancangan Sistem Informasi Jasa Menjahit Pada Berkah *Tailor* Jakarta Utara”. Latar belakang penelitian ini adalah karena proses pelayanan informasi yang diberikan kepada pelanggan masih dilakukan secara manual, oleh karena itu penulis merancang sebuah aplikasi sistem flowup

pelanggan berbasis pemrograman yang dapat membantu proses pelayanan menjadi cepat dan efisien dari segi biaya dan waktu. Model sistem yang diusulkan menggunakan bentuk flowchart dan diagram arus data (DAD). Dengan adanya aplikasi sistem informasi ini diharapkan dapat membantu pengolahan data, pencarian data dan pembuatan laporan data secara terkomputerisasi sehingga laporan yang dihasilkan menjadi berkualitas, serta dapat meminimalisir tingkat kesalahan yang biasanya dilakukan dalam sistem manual. [4]

Dengan melihat penelitian yang dilakukan sebelumnya diatas maka dapat disimpulkan bahwa perancangan aplikasi yang berjalan di bidang jasa menjahit baju masih sangat dibutuhkan untuk membantu sebuah organisasi bisnis baik itu skala kecil maupun besar dalam memberikan informasi mengenai produk yang dihasilkan kepada masyarakat sehingga memudahkan masyarakat dalam proses pemesanan barang yang lebih praktis dan cepat dari segi waktu maupun tenaga.

Perbedaan penelitian sebelumnya dengan penelitian yang dilakukan sekarang ini yang berjudul **“Penerapan Metode RAD (*Rapid Application Development*) Pada Perancangan Aplikasi Marketplace Jasa Menjahit Baju Berbasis Web Menggunakan *CodeIgniter*”** adalah penelitian ini bersifat marketplace khusus untuk jasa penjahit baju yang artinya semua pemilik baju bisa registrasi ke dalam aplikasi ini dan masyarakat bisa lebih leluasa untuk memilih jasa penjahit mana yang sesuai dengan keinginan mereka. Selain itu, masyarakat juga bisa memilih model baju yang sudah tersedia dan bisa juga menentukan ukuran mereka dengan menginput sendiri ukuran baju mereka, aplikasi ini juga dilengkapi fitur pembayaran dengan menginputkan nomer rekening yang dituju, maka baju jahitan yang telah di pilih sebelumnya akan langsung di proses oleh penjahit. Dalam aplikasi ini admin dapat menerima laporan data pengguna yang telah registrasi, data usaha yang masuk, laporan pemesanan, dan laporan pembayaran. aplikasi ini merupakan perantara bagi masyarakat untuk memesan baju jahit kepada penjahitnya langsung serta menjadi tempat untuk para pemilik usaha jasa menjahit untuk memberikan informasi mengenai model baju yang di produksi kepada masyarakat. Metode yang digunakan yaitu metode RAD (*Rapid Application Development*) yang

tahapan perancangan sistemnya lebih terstruktur. Serta pembuatan sistem yang menggunakan *framework CodeIgniter* dengan database *Mysql*.

## 2.2. Tinjauan Pustaka

### 2.2.1. Web

*Web* merupakan kata singkat dari *World Wide Web* (WWW), merupakan layanan jaringan internet yang berupa informasi. Dengan *web*, *user* akan memperoleh dan menemukan informasi dengan cara mengikuti *link* (*hyperlink*) yang akan di tampilkan oleh aplikasi *web browser*. Dengan adanya teknologi *web*, *user* tidak perlu menuliskan perintah apapun jika ingin berinteraksi dengan data yang tersimpan di dalam *web server*, karena permintaan data dari *web server* bisa dilakukan dengan hanya mengklik *link* yang tersedia di dalam dokumen.

Keuntungan yang didapat dari layanan *web* adalah sebagai berikut :

- Informasi mudah diakses oleh semua pengguna internet di dunia
- Konfigurasi *server* dapat dilakukan dengan mudah
- Instalasi (*upload*) aplikasi hanya dilakukan sekali saja, *user* tidak perlu melakukan instalasi aplikasi di setiap komputer yang ingin mengakses aplikasi tersebut
- Tidak tergantung pada *platform*, Informasi dan aplikasi bisa diakses dari komputer yang memiliki sistem operasi berbeda

Dengan adanya teknologi *web*, informasi akan ditampilkan dalam bentuk dokumen yang disebut halaman *web*. Halaman *web* adalah hasil terjemahan kode-kode program dari *file-file* yang tersimpan di dalam komputer yang bertindak sebagai *web server*. Komputer yang mengakses informasi disebut *web client*. Informasi yang tersimpan di *web server* dapat diakses melalui *web client* melalui aplikasi *web browser*. Contoh *web browser* yang banyak digunakan oleh pengguna internet adalah Internet Explorer dan Mozilla Firefox. Aplikasi *web browser* merupakan *software* yang berfungsi menampilkan dokumen-dokumen yang tersimpan di dalam *web server*. Untuk dapat mengakses informasi kita perlu melakukan instalasi aplikasi *web browser* pada komputer. URL (*Uniform Resource Locator*) merupakan rangkaian karakter yang disusun berdasarkan aturan tertentu, yang berfungsi untuk menunjukkan alamat suatu sumber misalnya dokumen atau gambar di internet. Kita perlu menuliskan URL untuk mengakses informasi melalui

*web browser*. Aturan penulisan URL tersusun mulai dari *protocol*, *hostname*, *path* dan *file*. [5]

### **2.2.2. Marketplace**

Marketing adalah kegiatan pemasaran yang saling menunjang antara satu dengan lainnya. Dalam dunia pemasaran keberhasilan perusahaan di dukung oleh empat kebijakan pemasaran yaitu produk (product), harga (price), saluran distribusi (place), dan promosi (promotion). [6] Dalam menghadapi persaingan bisnis di era globalisasi, Marketplace memberikan peluang besar bagi pemilik usaha bisnis di bidang apapun dalam menyampaikan informasi produk yang di hasilkan baik berupa barang atau jasa dengan menggunakan teknologi internet. Online market merupakan sebuah pasar online untuk segala usaha dalam melakukan pemasaran suatu produk dan jasa menggunakan media internet. Sedangkan place artinya tempat, jadi kesimpulannya marketplace adalah tempat atau wadah bagi para pemilik usaha untuk memasarkan produk-produk yang dihasilkan melalui media internet [7].

### **2.2.3. Jasa**

Jasa merupakan suatu aktivitas ekonomi yang ditawarkan oleh satu pihak ke pihak lain dan dilakukan dalam waktu tertentu (time-based). Suatu kegiatan (performance) yang membawa hasil yang diinginkan oleh penerima, obyek atau akses-akses lainnya yang menjadi tanggung jawab pembeli. Menjadi pertukaran uang, waktu, dan upaya, pelanggan jasa akan mendapatkan nilai (value) dari akses barang-barang, tenaga kerja, tenaga ahli, fasilitas, jejaring maupun sistem tertentu. Para pelanggan biasanya tidak mendapatkan hak milik dari unsur-unsur fisik yang ada dalam penyedia layanan jasa tertentu. [8]

### **2.2.4. PHP**

PHP merupakan Bahasa pemrograman yang digunakan oleh server web. PHP (*Personal Home Page*) atau FI (*Form Interface*) pertama kali dibuat oleh Rasmus Lerdoff. PPH awalnya adalah program CGI yang khusus untuk menerima input melalui form yang ditampilkan oleh browser web. CGI (*Common Gateway Interface*) merupakan suatu standar yang menghubungkan *interface* dengan server web. PHP telah dikembangkan menjadi Bahasa pemrograman script yang bisa dijalankan di atas *platform* sistem operasi secara langsung. PHP juga dapat

digunakan untuk membuat program desktop. Kemampuan PHP dapat signifikan karena adanya dukungan kepada banyak database. Untuk membuat halaman web sangat mudah dilakukan dengan menggunakan data dari database. Berikut daftar database yang didukung oleh PHP :

- Adabas D
- Hyperwave
- Interbase
- MSQL
- MySQL
- Oracle (OCI7 dan OC18)
- SQLite
- Unix DBM
- Dll

PHP juga menjadi Bahasa pemrograman untuk membuat aplikasi untuk pembuatan laporan. Fungsi yang disediakan untuk pembuatan laporan adalah dalam format PDF atau Excel dan dalam bentuk file teks. Hampir semua *server web* yang ada pada setiap sistem operasi komputer telah mendukung PHP. Untuk kemudahan pemasangan modul PHP maka sebaiknya pilih PHP versi stabil dalam format *file setup*. Jika menggunakan versi .zip bisa digunakan untuk mendownload versi paket lengkap yang disebut dengan stack, seperti XAMPP, versi yang terdapat *server web*, PHP, dan paket database MySQL. PHP bisa di download dari situs resmi PHP yaitu <http://php.net> di <http://windows.php.net/download>. [9]

#### **1.2.5. Sublime Text Editor**

Sublime Text adalah text editor berbasis python, text editor yang simple yang cukup terkenal di kalangan pengembang, penulis, dan desainer. Sublime text banyak digunakan programmer untuk mengetik source code untuk program yang sedang dilakukan. Sublime text menjadi favorit para pembuat website karena sifatnya yang ringan dan banyak fitur menarik. [10]

#### **1.2.6. MySQL**

MySql merupakan *database server* yang berfungsi untuk menangani *database* tujuannya adalah untuk mempermudah penyimpanan dan pengaksesan

data secara mudah dan cepat. MySQL merupakan *database* relasional yang datanya ada dua bentuk dimensi yang dinamakan tabel terdiri dari kolom dan baris. MySQL bisa diunduh di <http://www.php.net>, untuk mempermudah mengakses *database* melalui PHP kita bisa memanfaatkan WampServer. Kita bisa membuat aplikasi web dengan mudah melalui paket perangkat lunak gratis ini.

Proses pengaksesan *database* MySQL menggunakan program mysql yang berbasis DOS atau melalui phpMyAdmin. Langkah pertama untuk mempraktikkan *database* adalah dengan menciptakan *database* lalu menciptakan tabel-tabel yang ada di *database* dan selanjutnya adalah memasukkan data, menampilkan data, mengubah dan menghapus data. Pemrograman MySQL menggunakan SQL untuk mengakses *database* serta dapat menambahkan, mengubah dan menghapus data pada tabel. SQL (*Structured Query Language*) merupakan bahasa standar yang digunakan untuk mengakses *database* yang bersifat relasional. SQL mendukung perintah untuk membuat *database*, membuat tabel, menambahkan, mengubah, menghapus, mengambil data dan mengatur pengaksesan data. [11]

### 1.2.7. CodeIgniter

*CodeIgniter* (CI) merupakan framework pengembangan aplikasi (*Application Development Framework*) menggunakan PHP. Pembuat program tidak perlu membuat program dari awal (*from scratch*) karena CI mempunyai library yang diperlukan untuk menyelesaikan pekerjaan umum. *Framework* CI memiliki dokumentasi yang lengkap dan jelas. CI dirancang dan dikembangkan dengan acuan Apache contohnya adalah penggunaan file .htaccess milik Apache.

*CodeIgniter* memberikan dan menyediakan :

- Kerangka kerja dengan jejak yang sedikit (*small footprint*)
- Kinerja yang bagus
- Akun hosting standar yang dengan perbedaan versi dan konfigurasi
- Kerangka kerja yang tidak memerlukan baris perintah (*command line*)
- Kerangka kerja yang tidak mengharuskan menggunakan bahasa *templating*, walaupun CI menyediakan perser untuk *template*
- Menyediakan solusi sederhana untuk menghindari kompleksitas dalam pembuatan program
- Dokumentasi kerja yang menyeluruh dan jelas

Fitur dari *CodeIgniter* antara lain:

- Sistem berbasis MVC (*Model View Controller*)
- Merupakan *framework* yang ringan
- Memiliki fitur *class database* yang mendukung beberapa *platform*
- Class untuk pengiriman email yang mendukung lampiran (*attachment*), email dengan HTML/Teks, menggunakan berbagai protocol (sendemail, SMTP, dan Mail)
- Library manipulasi *image* (menggunting, mengubah ukuran, memutar, dan lainnya), mendukung SD, ImageMagick, dan NetPBM
- *Class* untuk *upload file*
- Class FTP
- Lokalisasi
- Paginasi
- Enkripsi data
- Logging error
- Class kalender
- Class user agent
- Class encoding zip
- Class engine template
- Class *trackback*
- Class unit pengujian
- URL yang bersahabat dengan search-engine
- Library fungsi helper yang banyak
- Dll

Dengan menggunakan *framework CodeIgniter* kita tidak perlu membuat program dari awal karena sudah diberikan library yang fungsi-fungsinya sudah terorganiser untuk menyelesaikan suatu program. *CI* memiliki kerangka kerja yang fleksibel, karena hanya dengan menentukan titik awal dari pembuat programnya yaitu bergerak dari *controller*. Paket *framework CodeIgniter* dapat di download dari situsnya : <http://codeigniter.com/download>. [12]

### 1.2.8. XAMPP

Xampp merupakan paket server web PHP dan MySQL berbasis *open source* yang digunakan sebagai *tool* pengembangan aplikasi yang berbasis PHP. XAMPP mengkombinasi beberapa perangkat lunak yang berbeda ke dalam satu paket. Sampai versi 1.7.4 telah dibundel beberapa paket antara lain adalah *Apache HTTPD*, *mod\_autoindex\_colormodule*, *Filezilla FTP server*, *Mercury Mail Transport Agent*, *OpenSSL*, *SQLite*, *The Webalizer*, *msmtp( a sendmail compatible SMTP Client)*, *MySQL*, *PrimeBase XT Storage Engine for MySQL*, *PHP*, *eAccelerator extension*, *Xdebug extension*, *Ming extension*, *PDFlib extension*, *PEAR*, *phpMyadmin*, *FPDF Library*, *ADODB*, *Perl*, *CPAN*, *PPM*, *mod\_pearl*, *Apache::ASP*. Kita bisa mendapatkan file XAMPP di <http://apachefriend.org> dan dapat memilih macam-macam versi sesuai kebutuhan dan pastinya harus disesuaikan dengan OS yang kita gunakan. [13]

Apache merupakan aplikasi *web server* yang dapat diperoleh di situs resmi Apache: <http://httpd.apache.org/>. Agar Apache bisa digunakan sebagai *web server* dari program PHP yang dibuat dengan melakukan sedikit konfigurasi terhadap Apache. Untuk menjalankan Apache, dilakukan dengan cara klik (*double click*) pada *icon* Apache yang berada di *task bar*. [14]

### 1.2.9. UML (*Unified Modeling Language*)

UML adalah bahasa visual untuk permodelan sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML berfungsi untuk melakukan permodelan dan penggunaannya tidak terbatas pada metodologi tertentu, meskipun kenyataannya UML banyak digunakan pada metodologi berorientasi objek. [15]

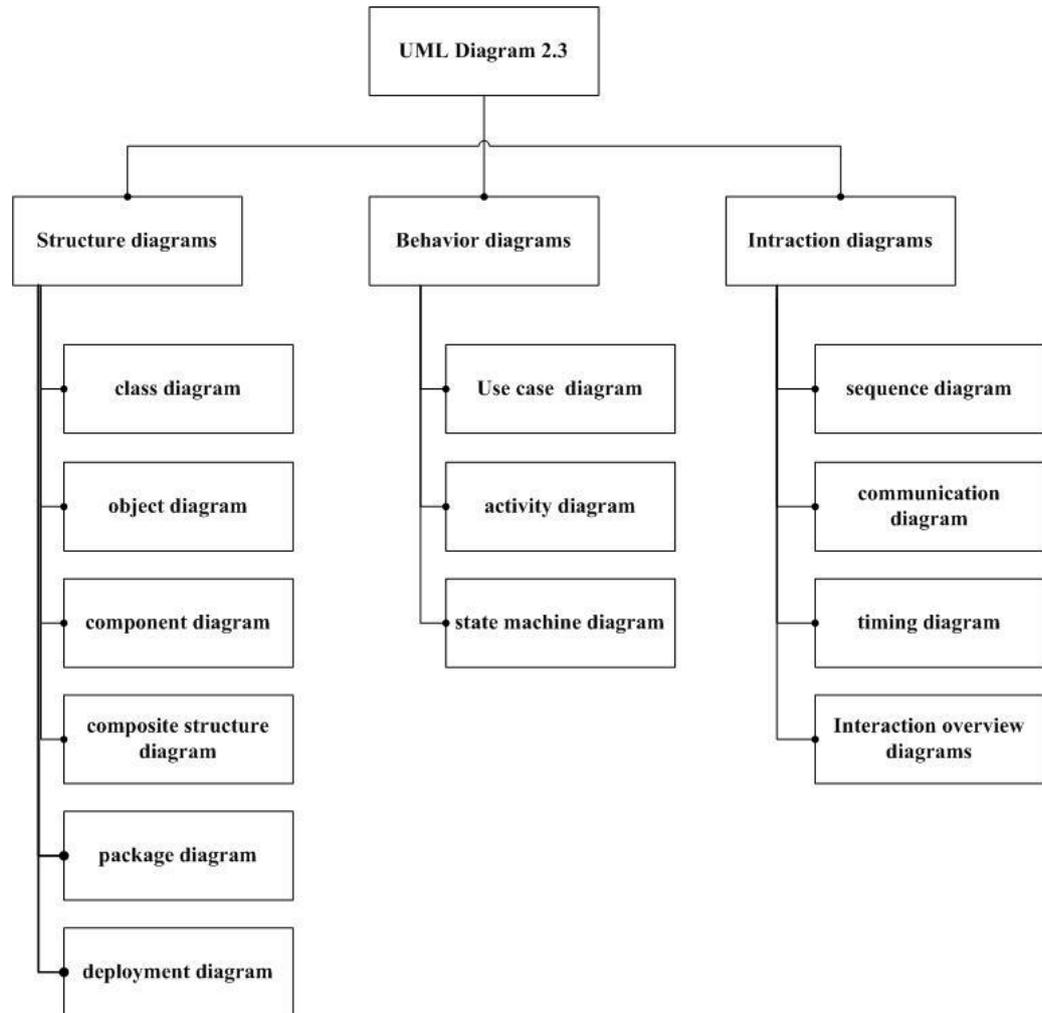
#### 1.2.9.1. Sejarah UML

Bahasa pemrograman yang pertama kali dikenal dengan nama Simula-67 pada tahun 1967, namun bahasa pemrograman ini kurang berkembang dan dikembangkan lebih lanjut. Pemrograman berorientasi objek mulai menggeliat ketika pada tahun 1980an berkembang bahasa pemrograman Smalltalk kemudian diikuti dengan berkembangnya bahasa pemrograman seperti C objek, C++, Eiffel, dan CLOS. Sekitar lima tahun setelah Smalltalk berkembang, muncul sebuah metode pertama yang diperkenalkan oleh Sally Shlaer dan Stephen Mellor 1988, Peter Coad dan Edward Yourdon 1991, diikuti oleh Grady Booch 1991, James R.

Rumbaugh, Michael R. Blaha, William Lorensen, Frederick Eddy, William Premerlani 1991 dan masih banyak lagi. Karena pada saat itu banyak metodologi yang berkembang pesat, maka muncullah ide untuk membuat bahasa yang mudah dimengerti oleh semua orang. Dibuatlah bahasa gabungan dari beberapa konsep seperti *Object Modelling Technique* (OMT), *The Classes, Responsibilities, Collaborators*, dan beberapa konsep dari James R. Rumbaugh, Grady Booch, dan Ivar Jacobson yang bergabung dalam perusahaan bernama Rational Software Corporation dan menghasilkan bahasa yaitu *Unified Modeling Language* (UML). Pada bulan September 1997 UML diakomodasi oleh OMG dan sampai saat ini UML telah berkontribusi cukup besar dalam metodologi berorientasi objek dan hal-hal yang terkait didalamnya.

#### **1.2.9.2. Diagram UML**

Dalam UML 2.3 terdiri dari 13 macam diagram dan dikelompokkan dalam 3 kategori. Pembagian kategori diagram dapat dilihat pada gambar berikut ini.



Gambar 2. 1 Diagram UML

Berikut penjelasan dari pembagian kategori :

- *Structure diagrams* adalah kumpulan diagram untuk menggambarkan struktur statis dari permodelan sistem
- *Behavior diagrams* adalah kumpulan diagram untuk menggambarkan kelakuan sistem atau serangkaian perubahan yang terjadi pada sistem
- *Interaction diagrams* adalah kumpulan diagram untuk menggambarkan interaksi sistem dengan sistem lain atau interaksi antara subsistem dengan suatu sistem

### 1.2.9.3. Class Diagram

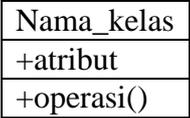
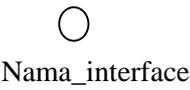
Diagram kelas merupakan struktur sistem dari pendefinisian kelas-kelas untuk membangun sistem. Kelas memiliki atribut dan metode atau operasi.

- Atribut adalah variable milik suatu kelas.
- Operasi atau metode merupakan fungsi-fungsi milik suatu kelas.

Diagram kelas dibuat supaya pembuat program membuat kelas dengan melihat rancangan di dalam kelas agar dokumentasi perancangan dengan perangkat lunak sinkron. Kelas-kelas pada struktur sistem harus bisa melakukan fungsi-fungsi sesuai kebutuhan sistem agar pembuat program dapat membuat kelas-kelas program perangkat lunak yang sesuai dengan rancangan diagram kelas. Struktur kelas pada diagram kelas yang baik adalah sebagai berikut :

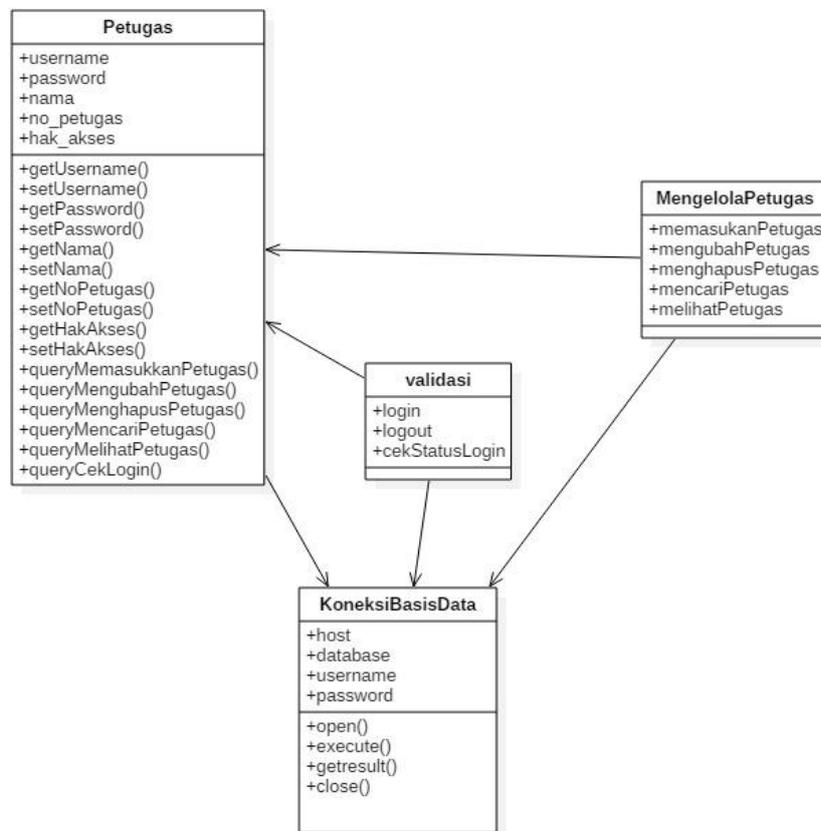
- Kelas main, merupakan fungsi awal yang dieksekusi ketika sistem berjalan
  - Kelas tampilam sistem (*view*), merupakan fungsi yang mengatur tampilan ke pemakai
  - Kelas dari pendefinisian *use case* (*Controller*), merupakan fungsi yang diambil dari pendefinisian *use case*, biasanya disebut dengan kelas proses pada perangkat lunak
  - Kelas dari pendefinisian data (*model*), merupakan kelas untuk membungkus data menjadi satu yang akan diambil maupun disimpan ke basis data
- Simbol-simbol diagram kelas adalah sebagai berikut :

Tabel 2. 1 Simbol Class Diagram

Simbol	Deskripsi
Kelas 	Kelas struktur sistem
Antarmuka/ <i>interface</i> 	Konsepnya sama dengan <i>interface</i> di pemrograman berorientasi objek

Asosiasi / <i>association</i> _____	Relasi antarkelas bermakna umum yang biasanya disertai <i>multiplicity</i>
Asosiasi berarah / <i>directed association</i> _____→	Relasi antarkelas bermakna kelas yang satu digunakan kelas lain yang biasanya disertai dengan <i>multiplicity</i>
Generalisasi _____▷	Relasi antarkelas bermakna generalisasi-spesialisasi (umum khusus)
Kebergantungan / <i>dependency</i> _____→	Relasi antarkelas bermakna kebergantungan antarkelas
Agregasi / <i>aggregation</i> _____◊	Relasi antarkelas bermakna semua bagian ( <i>whole-part</i> )

Berikut ini adalah contoh dari *Class Diagram*



Gambar 2. 2 Contoh *Class Diagram*

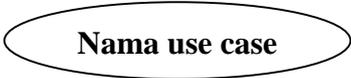
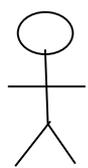
#### 2.2.9.4. Use Case Diagram

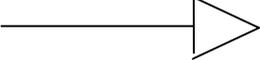
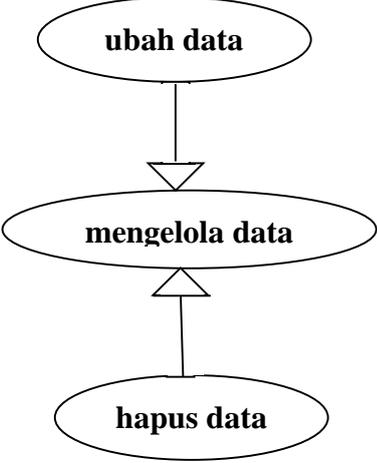
Diagram *use case* adalah model untuk kelakuan (*behavior*) suatu sistem yang akan dibuat. *Use case* mendeskripsikan interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use case* berguna untuk mengetahui apa saja fungsi yang ada di dalam sistem dan siapa saja yang berhak menggunakan fungsi-fungsi sistem tersebut. Terdapat dua hal pada *use case* yaitu pendefinisian aktor dan *use case*.

- Aktor adalah orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat diluar sistem yang akan dibuat itu sendiri, walaupun symbol dari aktor adalah gambar orang, akan tetapi aktor belum tentu orang.
- *Use case* adalah fungsionalitas yang disediakan oleh sistem sebagai unit-unit yang saling bertukar pesan dengan actor.

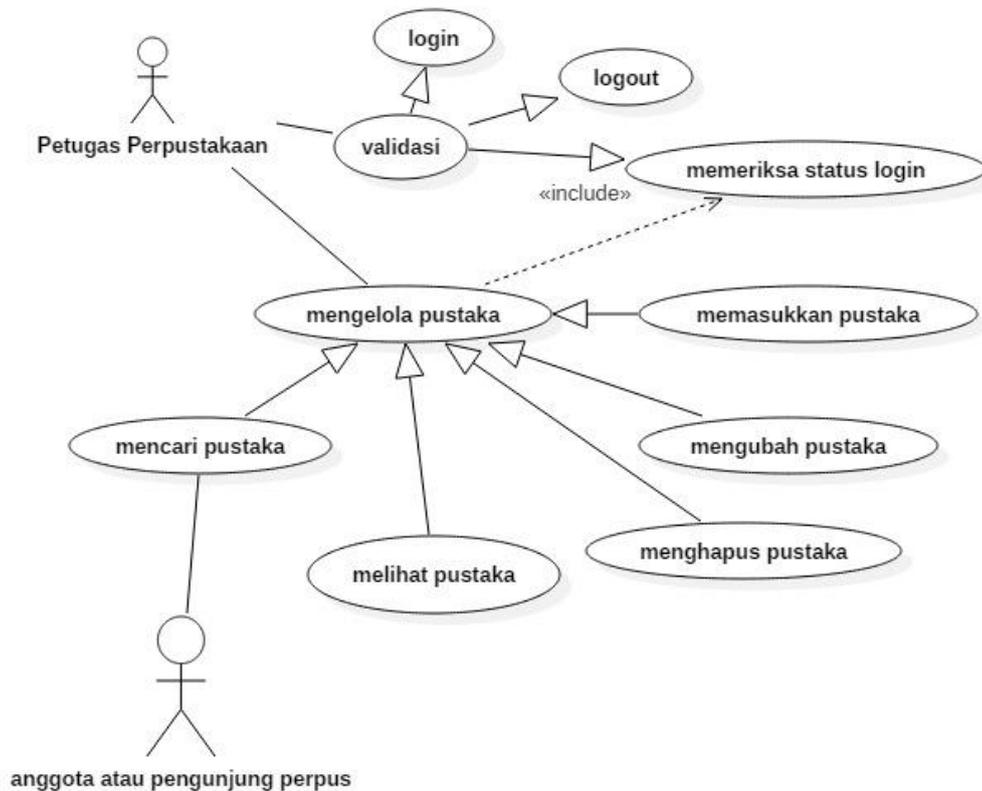
Berikut symbol-symbol diagram *use case*:

Tabel 2. 2 Simbol Use Case Diagram

Simbol	Deskripsi
<p><i>Use case</i></p>  <p><b>Nama use case</b></p>	<p>Fungsionalitas yang disediakan sistem untuk unit-unit yang bertukar pesan antara unit atau aktor biasanya menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p>Aktor/actor</p>  <p><b>Nama aktor</b></p>	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat diluar sistem yang akan dibuat itu sendiri, walaupun symbol dari aktor adalah gambar orang, akan tetapi aktor belum tentu orang. Dinyatakan menggunakan kata benda di awal frase nama aktor</p>
<p>Asosiasi/association</p> 	<p>Komunikasi antara aktor dan <i>use case</i> memiliki interaksi dengan aktor atau <i>use case</i> berpartisipasi pada <i>use case</i></p>

<p>Ekstensi/extend</p> <p style="text-align: center;">       &lt;&lt; extend &gt;&gt;        .....→     </p>	<p>Relasi <i>use case</i> tambahan yang dapat berdiri sendiri walau tanpa <i>use case</i> tambahan tersebut. Biasanya <i>use case</i> tambahan mempunyai nama depan yang sama dengan <i>use case</i> yang ditambahkan</p>
<p>Generalisasi/generalization</p> <p style="text-align: center;">       →  </p>	<p>Hubungan umum dan khusus antar dua <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya</p> <div style="text-align: center;">  <pre> graph TD     A(ubah data) --&gt; B(mengelola data)     C(hapus data) --&gt; B           </pre> </div> <p>arah panah mengarah pada <i>use case</i> yang menjadi umum</p>
<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p style="text-align: center;">       &lt;&lt;include&gt;&gt;        .....→     </p> <p style="text-align: center;">uses</p> <p style="text-align: center;">       →  </p>	<p>Relasi <i>use case</i> tambahan dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. Ada dua sudut pandang yang besar mengenai <i>include</i> di <i>use case</i>:</p> <ul style="list-style-type: none"> <li>• <i>Include</i> artinya <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan</li> <li>• <i>Include</i> artinya <i>use case</i> yang ditambahkan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan berjalan sebelum <i>use case</i> tambahan berjalan</li> </ul>

Berikut ini adalah contoh dari *Use Case Diagram*



Gambar 2. 3 Contoh *Use Case Diagram*

### 2.2.9.5. Activity Diagram

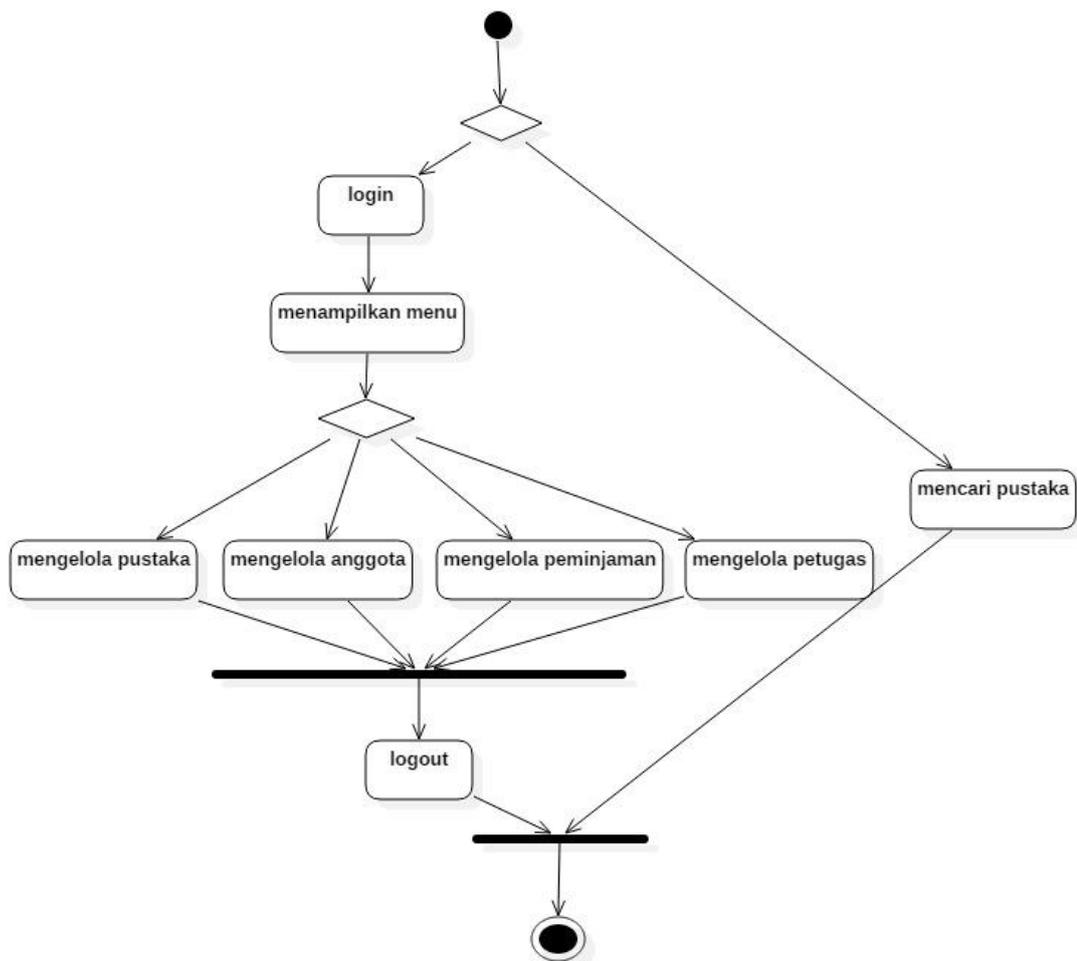
Diagram aktivitas yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sistem dan menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan aktivitas aktor, artinya aktivitas yang dilakukan oleh sistem. Berikut symbol-simbol diagram aktivitas :

Tabel 2. 3 Simbol Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal sistem, diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem dan biasanya diawali dengan kata kerja

Percabangan/ <i>decision</i>	Asosiasi percabangan, jika ada pilihan aktivitas lebih dari satu
Penggabungan/ <i>join</i>	Asosiasi penggabungan yang lebih dari satu aktivitas digabungkan menjadi satu
Status akhir	Status akhir yang dilakukan sistem
Swimlane	Pemisah dari organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Berikut ini adalah contoh dari *Activity Diagram*

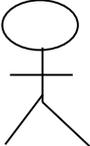
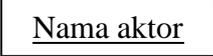
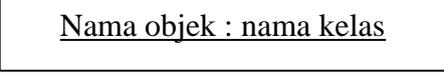


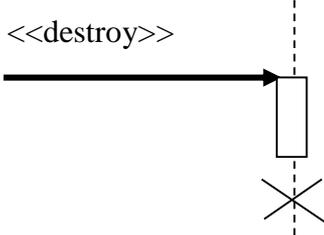
Gambar 2. 4 Contoh *Activity Diagram*

### 2.2.9.6. Sequence diagram

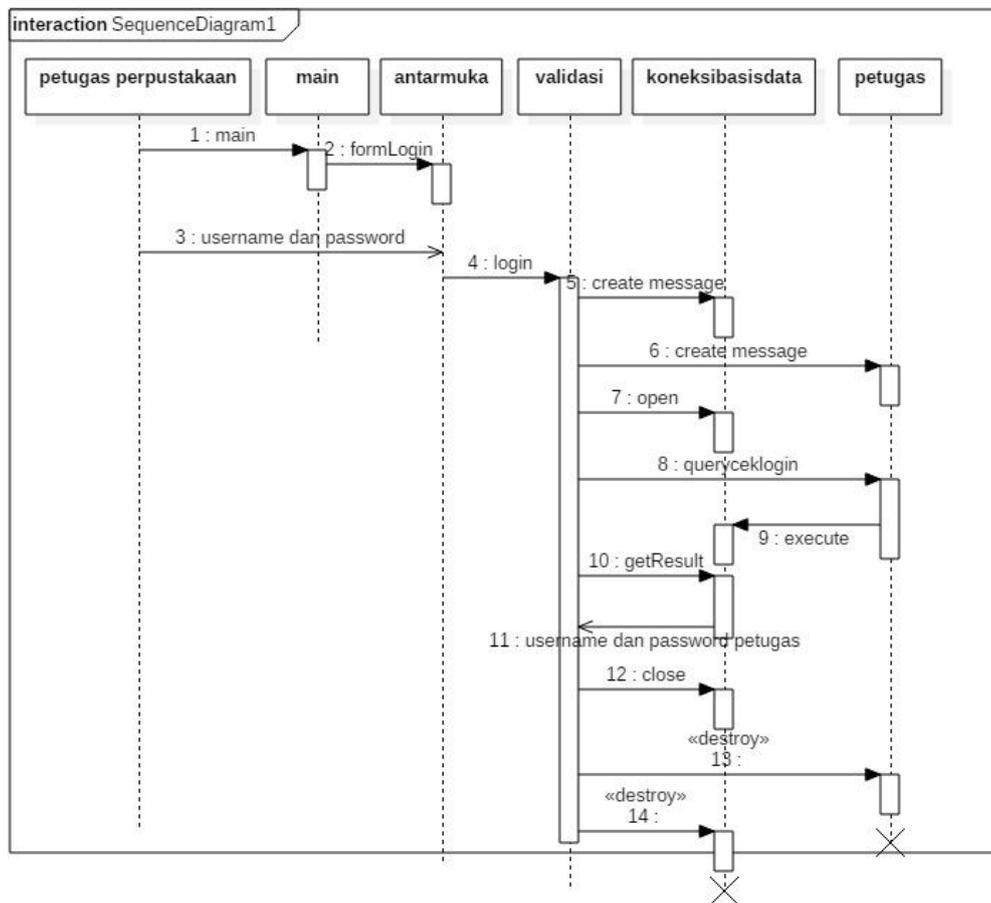
Diagram sekuen yang menggambarkan kelakuan objek terhadap *use case* dengan mengirimkan message lalu diterima antar objek. Untuk menggambar diagram sekuen harus diketahui objek-objek yang terlibat di dalam use case dan diagram sekuen dibutuhkan untuk melihat skenario dari *use case*. Banyaknya diagram sekuen tergantung berapa banyak pendefinisian *use case*, semakin banyak definisi *use case* maka semakin banyak pula diagram sekuen yang harus dibuat. Berikut simbol-simbol dari diagram sekuen :

Tabel 2. 4 Simbol Sequence Diagram

Simbol	Deskripsi
<p>Aktor</p>  <p><b>Nama aktor</b></p> <p>Atau</p>  <p>Tanpa waktu aktif</p>	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat diluar sistem yang akan dibuat itu sendiri, walaupun symbol dari aktor adalah gambar orang, akan tetapi aktor belum tentu orang. Dinyatakan menggunakan kata benda di awal frase nama aktor</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif berinteraksi yang artinya semua yang terhubung kedalam waktu aktif adalah sebuah tahapan yang dilakukan didalamnya</p>

<p>Pesan tipe create</p> <p style="text-align: center;">&lt;&lt;create&gt;&gt;</p> 	<p>Suatu objek membuat objek yang lain dan arah panah mengarah pada objek yang dibuat</p>
<p>Pesan tipe call</p> <p style="text-align: center;">1 : nama_metode()</p> 	<p>Suatu objek memanggil operasi atau metode yang ada pada objek lain ataupun dirinya sendiri</p>
<p>Pesan tipe send</p> <p style="text-align: center;">1 : masukan</p> 	<p>Suatu objek mengirimkan data atau masukan maupun informasi pada objek lain dan arah panah menuju ke objek yang dikirim</p>
<p>Pesan tipe return</p> <p style="text-align: center;">1 : keluaran</p> 	<p>Suatu objek yang telah menjalankan suatu operasi atau metode dan menghasilkan suatu kembalian ke objek tertentu. Arah panah menuju ke objek yang menerima kembalian</p>
<p>Pesan tipe destroy</p> <p style="text-align: center;">&lt;&lt;destroy&gt;&gt;</p> 	<p>Suatu objek mengakhiri hidup objek lain dan arah panah menuju objek yang diakhiri, jika ada create maka sebaiknya ada destroy</p>

Berikut ini adalah contoh dari *Sequence Diagram*



Gambar 2. 5 Contoh *Sequence Diagram*

### 2.2.10. Black Box Testing

Black box testing merupakan pengujian yang hanya mengamati hasilnya melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi, dengan menggunakan pengujian *black box* kita hanya bisa melihat dan mengetahui tampilan dan output dari data yang telah diinputkan tanpa mengetahui proses didalamnya. Pengujian ini menjadi fungsional atau non fungsional, walaupun biasanya fungsional. Dalam pengujian ini perancangan harus memilih input yang valid dan tidak valid untuk menentukan output yang benar. [16]

Beberapa klasifikasi black box testing yang digunakan dalam penelitian :

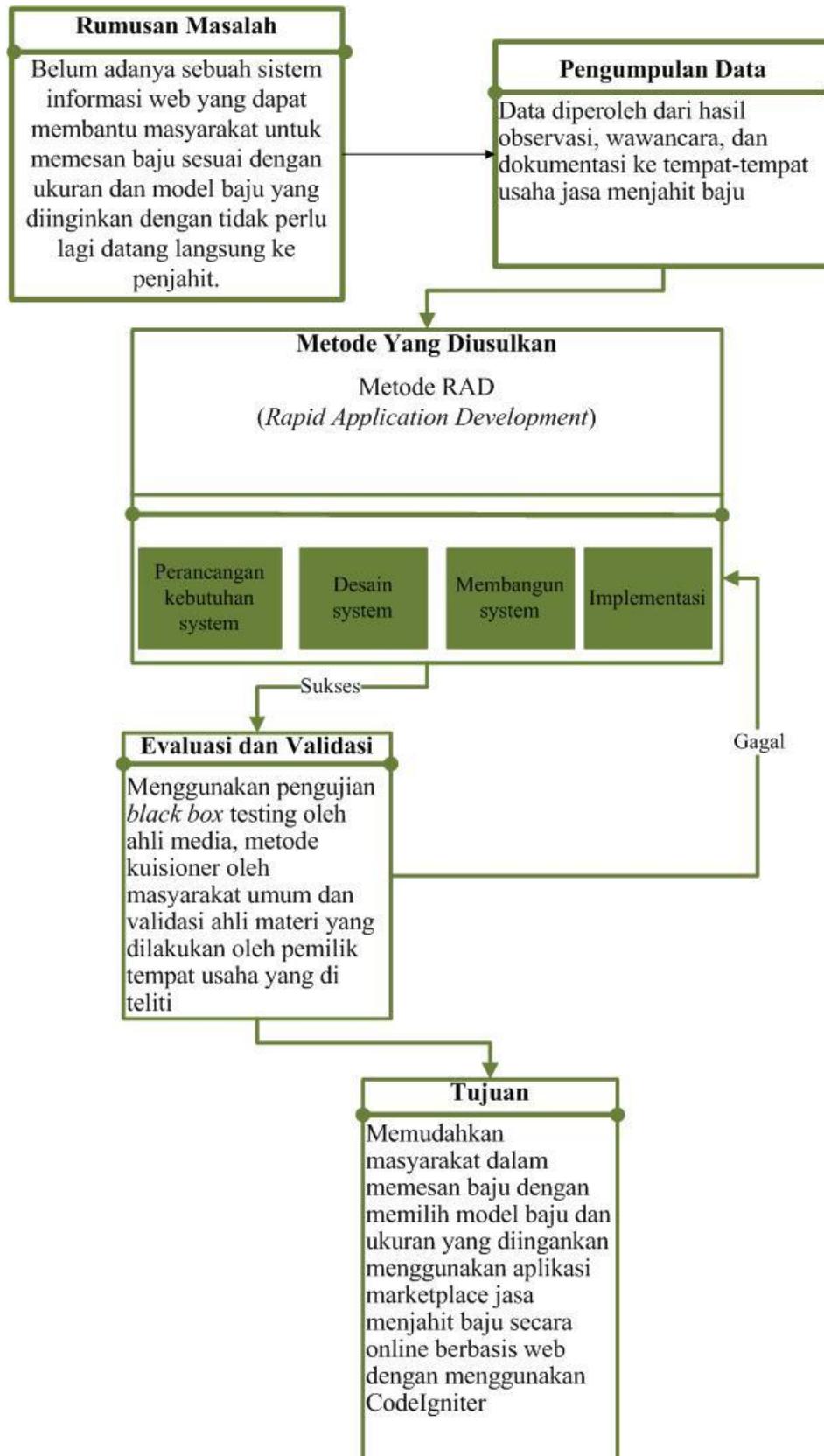
1. Pengujian fungsional (*functional testing*), adalah pengujian operasional untuk memastikan apakah produk mereka sesuai dengan spesifikasinya atau

tidak. Pengujian ini hanya fokus pada keluaran yang dihasilkan dari input yang dipilih.

2. Pengujian alfa (*alfa testing*), adalah pengujian operasional oleh pengguna atau tim uji independen dan pengujian ini biasanya digunakan sebelum perangkat lunak dilanjutkan ke pengujian beta.

Pengujian beta (*beta testing*), adalah pengujian yang dilakukan setelah pengujian alfa. Pengujian ini merupakan pengujian dari rilisnya produk perangkat lunak yang dilakukan pelanggan dan di luar tim pemrograman. [17]

### 2.3. Kerangka Berfikir



Gambar 2. 6 Kerangka Berfikir

