

## **BAB II**

# **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

### **2.1 Tinjauan Pustaka**

Dalam penelitian ini, diperoleh beberapa jurnal yang mengkaji mengenai mengenai sistem informasi donasi *online* berbasis *website* seperti berikut ini.

Penelitian sebelumnya telah dilakukan oleh Sri Mulandari, dkk (2021). pada penelitian tersebut membahas donasi online yang dibangun berbasis *website*. Sistem informasi ini dikembangkan dengan metode *waterfall*. Dengan permasalahan mewujudkan pembuatan *website* khusus yayasan atau lembaganya sendiri, yayasan atau lembaga yang menggunakan buku serta mengolah data dengan menggunakan *microsoft excel* yang mana kurang efektif dan efisien. Selain itu, penyampaian laporan penggunaan dana kepada para donatur juga sering terhambat sehingga menimbulkan ketidakpercayaan para donatur. Sehingga *website* donasi *online* memberikan solusi untuk kemudahan dan membantu para donatur untuk dapat melakukan donasi secara *online* kepada masyarakat yang membutuhkan[4].

Penelitian sebelumnya telah dilakukan oleh Syahwan Rinaldi, dkk (2022). Pada penelitian tersebut membahas mengenai donasi *online* yang di provokasi pemuda berbasis *website responsive* dengan permasalahan dalam pencatatan dan pendataan yang dilakukan organisasi dilakukan dengan cara proses tradisional berbasis kertas dan bolpoin, sehingga sangat rentan terjadinya kehilangan dan kerusakan karena manajemen arsip yang tidak dikelola dengan baik. Oleh karena itu melalui adanya Popay dapat membantu pengurus organisasi dalam pengelolaan data secara sistematis dan memudahkan donatur dalam mendapatkan informasi kegiatan dan laporan penggunaan dana[5].

Penelitian sebelumnya telah dilakukan oleh Hasanuddin, dkk (2022). Pada sistem ini membahas tentang donasi *online* dengan arsitektur *Rest Api*. Masalah yang di temukan dalam

melakukan penelitian ini adalah kegiatan donasi dilakukan melalui media *offline* atau langsung sehingga mempersulit masyarakat dalam melakukan transaksi donasi. Dari permasalahan tersebut, dikembangkan sebuah sistem berbasis android. Penelitian ini menghasilkan sistem berbasis API dengan arsitektur *REST* dalam hal *backend* untuk memudahkan masyarakat dalam melakukan transaksi donasi[6].

Penelitian sebelumnya telah dilakukan oleh Monanda Rio Meta, dkk (2022). Pada sistem ini membahas tentang donasi *online* dengan arsitektur *Rest Api*. Masalah yang ditemukan dalam melakukan penelitian ini kesulitan dalam pengolahan data dan penyebaran informasi tentang panti asuhan. Dari permasalahan tersebut, dikembangkan sebuah sistem berbasis *website* dengan metode *waterfall*. Hasil penelitian adalah *website* untuk pengolahan data dan sebagai media baru untuk menyebarkan informasi mengenai Panti Asuhan Bundo Saiyo kepada masyarakat[7].

Penelitian sebelumnya telah dilakukan oleh Wiwit Lestari, dkk (2022). Pada sistem ini membahas tentang penyaluran donasi untuk korban bencana alam. Masalah yang di temukan dalam melakukan penelitian ini adalah adanya rasa kekhawatiran yang ada yaitu penggalangan dana palsu (penggalangan dana ilegal). Dari permasalahan tersebut, dikembangkan sebuah sistem berbasis *website* dengan menggunakan metode *waterfall*. Hasil penelitian ini adalah mempermudah donatur dalam melakukan donasi berupa pakaian layak, bahan pangan maupun uang *cash* atau *E-Money* seperti *Gopay*, *Ovo* tanpa harus datang langsung ke lokasi penggalangan dana[8].

Pada penelitian kali ini, penulis membangun Sistem Informasi Donasi di Yayasan Gerak Sedekah Cilacap. Sistem ini bertujuan untuk memberikan informasi mengenai *campaign* Yayasan Gerak Sedekah Cilacap kepada masyarakat luas. Selain itu, mempermudah admin atau staff yayasan dalam mengakses dan mengelola data donasi, mengurangi resiko penipuan donasi yang menyebabkan kesalahan pencatatan, serta memudahkan Donatur dalam proses transaksi pembayaran yang bisa menggunakan platform pembayaran seperti *E-wallet*, *Virtual*

*Account (VA), Transfer, Kartu kredit, dan lain-lain yang didukung payment gateway.*

Tabel 2. 1 Perbandingan Penelitian Sebelumnya

No	Judul	Penulis (Tahun)	Persamaan	Perbedaan
1	Sistem Informasi Donasi <i>Online</i> Berbasis <i>Website</i>	Sri Mulandari, Yuni Fitriani, Sri Utami, Bambang Junadi (2021)	Metodologi pengujian yang digunakan yaitu <i>black-box testing</i> .	<p><b>Penelitian Sebelumnya:</b></p> <p>a. Validasi Pembayaran dengan cara upload <i>screenshoot</i> donasi yang masuk sebagai pembuatan laporan.</p> <p>b. Metode Pengembangan <i>Waterfall</i>.</p> <p><b>Penelitian yang akan Dirancang:</b></p> <p>a. Menggunakan <i>Payment Gateway (Midtrans)</i>.</p> <p>b. Metode Pengembangan <i>RAD</i>.</p>
2	Sistem Informasi Manajemen Organisasi Pemuda Peduli Anak Yatim ( <i>Popay</i> ) Berbasis <i>Website Responsive</i>	Syahwan Rinaldi, Diah Rahmawati, Asep Hardiyanto Nugroho (2022)	Menggunakan metode pengembangan <i>RAD</i> .	<p><b>Penelitian Sebelumnya:</b></p> <p>a. Hanya mencakup donasi untuk mengelola data anak yatim.</p> <p><b>Penelitian yang akan Dirancang:</b></p> <p>a. Mencakup banyak program <i>campaign</i> yang sudah di tentukan oleh yayasan.</p>

3	Rancang Bangun <i>Rest API</i> Aplikasi <i>Weshare</i> Sebagai Upaya Mempermudah Pelayanan Donasi Kemusiaan	Hasanuddin, Hari Asgar, Budi Hartono (2022)	Menggunakan Arsitektur <i>Rest API</i>	<p><b>Penelitian Sebelumnya:</b></p> <ol style="list-style-type: none"> <li>Berbasis <i>Android</i></li> <li>Metode Pengembangan <i>Agile</i>.</li> <li>Konfirmasi Pembayaran Donasi di <i>Update Manual (Paid</i> jika sukses, <i>Unpaid</i> jika belum dibayar).</li> <li>Menggunakan Teknologi <i>Backend framework Expressjs</i> dengan Bahasa <i>JavaScript</i> dan Frontend <i>React Native</i> dengan Bahasa <i>JavaScript</i>.</li> <li><i>Database Postgresql</i>.</li> </ol> <p><b>Penelitian yang akan Dirancang:</b></p> <ol style="list-style-type: none"> <li>Berbasis <i>Website</i></li> <li>Terintegrasi <i>Payment Gateway</i>.</li> <li>Metode Pengembangan <i>RAD</i>.</li> <li>Menggunakan Teknologi <i>Backend framework Laravel</i> dengan Bahasa <i>PHP</i> dan Frontend <i>Vuejs</i> dengan Bahasa <i>JavaScript</i>.</li> <li><i>Database Mysql</i>.</li> </ol>
---	---	---	--	---

4	Perancangan Sistem Informasi Pada Panti Asuhan Saiyo Berbasis Web	Monanda Rio Meta, Arman, Rajab, Hadi Sepri Yeni, Zainul Efendy (2022)	Menggunakan Bahasa Pemrograman <i>PHP</i> dan <i>Mysql</i> .	<p><b>Penelitian Sebelumnya:</b></p> <ol style="list-style-type: none"> <li>Metode Pengembangan <i>Waterfall</i>.</li> <li>Hanya mencakup pengolahan data dan penyebaran informasi tentang panti asuhan.</li> </ol> <p><b>Penelitian yang akan Dirancang:</b></p> <ol style="list-style-type: none"> <li>Metode Pengembangan <i>RAD</i>.</li> <li>Mencakup <i>Campaign</i> untuk kebutuhan masyarakat umum yang membutuhkan.</li> </ol>
5	Penyaluran Untuk Donasi Korban Alam Berbasis <i>Website</i>	Wiwit Lestari, Inge Wina Checathia, Juan Immanuel, Dwi Ade Handayani Capah (2022)	<ol style="list-style-type: none"> <li>Menggunakan Bahasa Pemrograman <i>PHP</i> dan <i>Mysql</i>.</li> <li>Metodologi pengujian yang digunakan yaitu <i>black-box testing</i>.</li> </ol>	<p><b>Penelitian Sebelumnya:</b></p> <ol style="list-style-type: none"> <li>Hanya Mencakup untuk kebencanaan.</li> <li>Metode Pengembangan <i>Waterfall</i>.</li> </ol> <p><b>Penelitian yang akan Dirancang:</b></p> <ol style="list-style-type: none"> <li>Mencakup <i>Campaign</i> untuk kebutuhan masyarakat umum yang membutuhkan dengan kategori yang sudah ditentukan yayasan.</li> <li>Metode Pengembangan <i>RAD</i>.</li> </ol>

## 2.2 Landasan Teori

### 2.2.1 Sistem, Informasi, dan Sistem Informasi

Sistem merupakan suatu kesatuan usaha yang terdiri dari bagian-bagian yang berkaitan satu sama lain yang berusaha mencapai suatu tujuan dalam suatu lingkungan kompleks[9].

Sistem juga merupakan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran tertentu. Supaya sistem dapat dikatakan sistem yang baik, maka perlu memiliki karakteristik sebagai berikut:

1. **Komponen Sistem**  
Suatu sistem terdiri dari komponen-komponen yang saling bekerja sama untuk membentuk suatu kesatuan. Komponen sistem terdiri dari beberapa subsistem.
2. **Batasan Sistem**  
Batasan sistem berarti daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya.
3. **Lingkungan Luar Sistem**  
Lingkungan Luar Sistem berada diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan dan merugikan. Untuk yang bersifat menguntungkan harus tetap dijaga, sedangkan yang bersifat merugikan harus dijaga dan dikendalikan supaya tidak mengganggu kelangsungan hidup dari suatu sistem.
4. **Penghubung Sistem**  
Penghubung Sistem adalah media penghubung antara satu subsistem dengan subsistem lainnya.
5. **Masukan Sistem**  
Masukan sistem merupakan energi yang dimasukkan kedalam sistem, dapat berupa perawatan (*maintenance input*), dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* merupakan energi yang diproses untuk didapatkan keluaran.
6. **Keluaran Sistem**  
Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna.

## 7. Pengolahan Sistem

Suatu sistem menjadi bagian pengolahan yang akan merubah masukan menjadi keluaran.

## 8. Sasaran Sistem

Suatu sistem pasti memiliki tujuan. Tujuan dari sistem sangat menentukan *input* yang dibutuhkan dan *output* yang akan dihasilkan oleh sistem.

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya. Sumber dari informasi adalah data. Menurut Gordon B. Davis, informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi si penerima dan mempunyai nilai nyata atau yang dapat dirasakan dalam keputusan-keputusan yang sekarang atau keputusan-keputusan yang akan datang[9].

Jadi dapat disimpulkan bahwa sistem informasi merupakan suatu sistem di dalam organisasi yang berkaitan satu sama lain dengan menggunakan data sebagai informasi untuk mencapai suatu tujuan tertentu.

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*) yaitu:

1. Blok Masukan (*Input Block*) : Blok masukan merupakan data yang masuk ke dalam sistem informasi.
2. Blok Model (*Model Block*) : Blok model terdiri dari kombinasi prosedur, logika, dan metode matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara tertentu untuk menghasilkan keluaran yang diinginkan.
3. Blok Keluaran (*Output Block*) : Blok keluaran merupakan produk dari sistem informasi yang berupa informasi berkualitas dan dokumentasi yang berguna bagi semua tingkatan manajemen serta semua pemakai sistem.
4. Blok Teknologi (*Technology Block*) : Blok Teknologi berfungsi untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian diri secara keseluruhan. Blok Teknologi terdiri dari 3 unsur utama yaitu: teknisi, perangkat lunak, dan perangkat keras.
5. Blok Basis Data (*Database Block*) : Blok basis data merupakan kumpulan dari data yang saling terhubung satu



dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Blok Kendali (*Control Block*) : Blok kendali berfungsi untuk mengendalikan sistem supaya sistem dapat dicegah dari kerusakan.

### **2.2.2 Filantropi**

Dasar landasan dari lembaga filantropi dijelaskan dalam PP Nomor 29 Tahun 1980 bahwa pengumpulan sumbangan untuk pembangunan dalam bidang kesejahteraan sosial sebagaimana dimaksud dalam Undang-undang Nomor 9 Tahun 1961 tentang Pengumpulan Uang atau Barang[10].

### **2.2.3 Yayasan**

Yayasan yaitu suatu lembaga badan hukum yang memiliki sifat sosial, kemanusiaan, serta keagamaan yang didirikan dengan persyaratan formal sesuai ketentuan undang – undang. Dasar pendirian yayasan diatur dalam undang – undang nomor 16 Tahun 2001[11]. Selain itu, dasar lain dijelaskan pada UU Republik Indonesia Nomor 28 Tahun 2004 Tentang Yayasan[12].

### **2.2.4 Donasi**

Donasi atau derma merupakan pemberian sesuatu seperti pada umumnya yang bersifat fisik secara sukarela pada perorangan atau badan hukum[6]. Dalam jurnal lain donasi adalah memberikan sesuatu pada orang yang membutuhkan[13].

### **2.2.5 Fundraising**

Proses pengumpulan barang atau jasa secara sukarela dalam bentuk uang atau sumberdaya lain[14]. Dalam jurnal lain juga dijelaskan *fundraising* adalah suatu kegiatan menghimpun dana atau mengumpulkan dana[15].

### **2.2.6 Platform Digital**

Suatu program yang dapat menunjang suatu kegiatan[16]. Dalam jurnal lain platform digital adalah penunjang suatu kegiatan[15].

### **2.2.7 Crowdfunding**

Platform penggalangan dana secara online seperti kitabisa.com, amalsholeh.com[17]. Dalam jurnal lain *Crowdfunding* yaitu platform yang dapat mengumpulkan dana tanpa harus melakukan tatap muka[15].

### **2.2.8 Website**

Sekumpulan halaman *web* yang saling berhubungan yang berisikan suatu informasi yang disediakan oleh perorangan, sekelompok, atau organisasi[4]. Dalam jurnal lain *website* merupakan sekumpulan halaman web yang saling terhubung dan terkait yang memuat informasi berbagai berbentuk media seperti teks, gambar, animasi, suara, film, dan lain sebagainya[18].

### **2.2.9 Payment Gateway**

Transaksi online yang menjembatani proses pembayaran semua platform baik kartu kredit, *transfer bank*, dompet digital seperti (Gopay, OVO, Dana), direct debit (BCA KlikPay, Mandiri Clickpay, CIMB Clicks, dan e-Pay BRI) dan e-wallet (TCASH dan XL Tunai), Virtual Account, Alfamart, dan sebagainya[19]. Dalam jurnal lain dijelaskan *payment gateway* adalah pembayaran *online* yang menguraikan data informasi pada sebuah transaksi yang sudah diatur oleh *provider*[20].

### **2.2.10 Restful API**

Arsitektur komunikasi standar yang digunakan untuk menghubungkan semua platform sehingga dapat saling berinteraksi dan di akses secara terprogram(Hasanuddin et al., 2022). Dalam jurnal lain dijelaskan *REST Api* adalah arsitektur penggabungan web platform dimana *client* membuat permintaan terhadap *server* dan diproses *server* untuk mengembalikan *response*[21].

### **2.2.11 Basis Data**

Basis data (*database*) adalah penampung berbagai yang saling terkait satu sama lain, sehingga memudahkan pengelolaan data menjadi informasi yang bermanfaat. Dalam jurnal lain basis data adalah wadah pengelolaan data yang ada pada sistem[22].

Basis Data atau *Database* adalah sistem terkomputerisasi yang memiliki tujuan utamanya untuk memelihara data yang sudah diolah atau informasi, serta membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa *file* teks ataupun *Database Management System* (DBMS)[23].

*Database Management System* (DBMS) atau dalam bahasa Indonesia sering disebut sebagai Sistem Manajemen Basis Data merupakan suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data[23]. Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut:

1. Menyediakan fasilitas untuk mengelola akses data.
2. Mampu menangani integritas data.
3. Mampu menangani backup data[23].

DBMS dibagi menjadi dua versi, yaitu versi komersial dan versi *open source*. Untuk versi komersial yang paling banyak digunakan di dunia saat ini adalah:

1. Oracle.
2. Microsoft SQL Server.
3. IBM DB2.
4. Microsoft Access[23].

Sedangkan DBMS versi *open source* yang paling banyak digunakan saat ini adalah:

1. MySQL.
2. PostgresSQL.
3. Firebird.
4. SQLite[23].


Pada DBMS terdapat bahasa standar yang digunakan untuk mengakses data yang disebut *Structured Query Language* (SQL). *Structured Query Language* (SQL) merupakan bahasa yang tidak berpengaruh terhadap penggunaan huruf kapital maupun huruf kecil atau yang biasa disebut tidak menganut *case sensitive*[24]. Berikut ini adalah contoh pengaksesan data pada DBMS dengan SQL yang secara umum terdiri dari empat hal sebagai berikut:




1. Memasukkan data (*insert*);
2. Mengubah data (*update*);
3. Menghapus data (*delete*);
4. Menampilkan data (*select*)[23].

Selain penggunaan bahasa SQL, pada DBMS juga dibutuhkan pemodelan awal untuk mempermudah pembuatan database. Salah satu pemodelan awal yang paling sering digunakan adalah *Entity Relationship Diagram* (ERD).

*Entity Relationship Diagram* (ERD) digunakan untuk pemodelan relasional. Tujuan ERD adalah untuk menunjukkan objek data dan *relationship* yang ada pada objek tersebut[23]. Di samping itu, ERD merupakan salah satu alat untuk perancangan dalam basis data. Dalam ERD memiliki simbol-simbol yaitu sebagai berikut:

**Tabel 2. 2** Simbol ERD

Nama	Gambar	Deskripsi
Entitas ( <i>Entity</i> )		Entitas merupakan data inti yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer. Penamaan entitas biasanya lebih ke arah kata benda dan belum merupakan nama tabel.
Relasi ( <i>Relationship</i> )		Relasi yang menghubungkan

		antar entitas biasanya diawali dengan kata kerja.
Atribut		Atribut adalah karakteristik dari entitas atau dari relasi yang menyediakan penjelasan detail tentang entitas atau relasi tersebut. Atribut berupa <i>field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
Penghubung ( <i>Line / Connector</i> )		Penghubung antara relasi dan entitas yang di kedua ujungnya memiliki kemungkinan jumlah pemakaian. Kemungkinan jumlah pemakaian maksimum keterhubungan antara entitas satu dengan yang lainnya dapat disebut kardinalitas. 1 : M

		atau sering disebut <i>one to many</i> menghubungkan entitas A dengan entitas B.
--	--	--

### 2.2.12 Rekayasa Perangkat Lunak (RPL)

Perangkat lunak (*software*) merupakan program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*). Sedangkan rekayasa perangkat lunak (*software engineering*) merupakan pembangunan dengan menggunakan prinsip rekayasa yang bertujuan untuk menghasilkan perangkat lunak sehingga memiliki nilai ekonomi yang dipercaya dan dapat bekerja secara efisien menggunakan mesin[23].

Rekayasa perangkat lunak lebih fokus pada bagaimana membuat perangkat lunak yang memenuhi kriteria sebagai berikut:

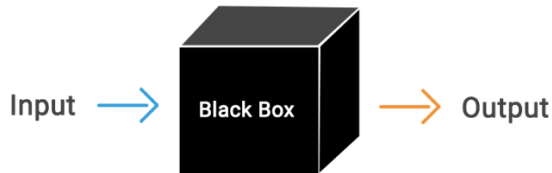
- a. Dapat terus dipelihara setelah perangkat lunak selesai dibuat seiring perkembangan teknologi dan lingkungan.
- b. Dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi.
- c. Efisien dari segi sumber daya dan penggunaan.
- d. Kemampuan untuk dipakai sesuai dengan kebutuhan[23].

Untuk mengembangkan rekayasa perangkat lunak maka dibutuhkan metode pengembangan sistem, metode pengujian sistem, dan *tools* (alat bantu).

### 2.2.13 Metode Pengujian Sistem *Black-Box*

Pengujian sistem merupakan pengujian perangkat lunak atau *software* yang lengkap dan terintegrasi. Salah satu metode pengujian sistem pada rekayasa perangkat lunak adalah metode pengujian sistem *black-box*[25].

## Black Box Testing



**Gambar 2. 1** Metode Pengujian Sistem *Black Box*

Metode pengujian sistem *black-box* atau dalam bahasa Inggris disebut *black-box testing* merupakan metode pengujian perangkat lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur program atau *internal code*[25]. Penguji akan menyadari apa yang harus dilakukan oleh program, tapi tidak mempunyai pengetahuan bagaimana melakukannya[26]. Metode pengujian ini dilakukan untuk mengetahui *error-error* di dalam kategori berikut :

1. *Error* desain antarmuka;
2. *Error* di dalam struktur data atau *database*;
3. *Error* di bagian performanya;
4. *Error* di bagian terminasi dan inisialisasi;
5. Tidak benar atau hilangnya fungsionalitas[26].

*Black box testing* ini memiliki beberapa kelebihan dan kekurangan. Berikut ini adalah beberapa kelebihan dan kekurangan yang terdapat pada *black box testing*[25].

Kelebihan *black box testing* yaitu:

1. Tidak memerlukan kode akses;
2. Pemisahan antara perspektif pengguna dan pengembang;
3. Efisien untuk segmen kode besar[25].

Kekurangan *black-box testing* yaitu:

1. Proses pengujian tidak efisien, karena keberuntungan penguji diukur dari pengetahuan tentang perangkat lunak internal.
2. Memiliki cakupan yang terbatas, karena hanya sebagian kecil dari skenario pengujian yang dilakukan[25].

Selain memerlukan metode pengembangan dan metode pengujian sistem, untuk mengembangkan perangkat lunak, seorang *software engineer* juga membutuhkan aplikasi yang dapat membantu pembuatan sistem perangkat lunak atau yang biasa di sebut alat bantu (*tools*) rekayasa perangkat lunak[27].

#### **2.2.14 Alat Bantu (Tools)**

Alat bantu (*tools*) pada perangkat lunak adalah aplikasi atau perangkat lunak yang membantu pembuatan sistem perangkat lunak. *Tools* muncul karena para pelaku rekayasa perangkat lunak membutuhkan perangkat yang dapat mempermudah pekerjaan mereka. Berikut adalah beberapa alat bantu untuk mempermudah *software engineer* dalam mengembangkan rekayasa perangkat lunak[23].

#### **2.2.15 Unified Modeling Language (UML)**

*Unified Modeling Language* (UML) adalah bahasa visual untuk komunikasi dan pemodelan mengenai sebuah sistem menggunakan diagram dan teks-teks pendukung. Penggunaan UML tidak terbatas pada metodologi tertentu karena UML hanya berfungsi untuk melakukan pemodelan. Chonoles (2003) mengatakan bahwa UML sebagai bahasa memiliki arti sintaks dan semantik. Ketika membuat model dengan UML, kita juga tidak sembarangan dalam menggambarkan diagram-diagramnya. Karena di dalam diagram tersebut juga menceritakan konteks dari sistem yang kita buat[28].

Ada beberapa diagram dalam UML yaitu:

##### **a) Use Case Diagram**

Diagram *use case* merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Penamaan pada *use case* didefinisikan sesederhana mungkin. Terdapat dua hal utama pada *use case* yaitu pendefinisian aktor dan *use case*[23].

##### **1) Aktor**

Aktor adalah orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.





Jadi, meskipun simbol dari aktor adalah gambar orang, tetapi aktor tersebut belum tentu merupakan orang[23].




2) *Use Case Diagram*

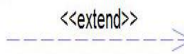

*Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antar satu unit atau aktor dengan unit atau aktor lain[23].

Berikut ini adalah simbol-simbol yang ada pada diagram *use case*:

**Tabel 2. 3** Simbol *Use Case Diagram*

Nama	Gambar	Deskripsi
<i>Use case</i>		<p><i>Use case</i> merupakan fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antar satu unit atau aktor dengan unit atau aktor lain.</p> <p><i>Use case</i> biasanya dinyatakan menggunakan kata kerja di awal frase nama <i>use case</i>.</p>
Aktor / Actor		<p>Aktor adalah orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang</p>

		akan dibuat itu sendiri. Jadi, meskipun simbol dari aktor adalah gambar orang, tetapi aktor tersebut belum tentu merupakan orang.
Asosiasi		Asosiasi antara aktor dan <i>use case</i> digambarkan dengan garis tanpa panah yang dapat diartikan sebagai siapa atau apa yang meminta interaksi secara langsung.
Asosiasi		Asosiasi antara aktor dan <i>use case</i> dengan garis panah terbuka dapat mengidentifikasikan bila aktor berinteraksi secara pasif dengan sistem.
<i>Include</i>		<i>Include</i> , terdapat di dalam <i>use case</i> lain atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan

		sebuah fungsi sistem.
<i>Extend</i>		<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain apabila kondisi atau syarat terpenuhi.
<i>Boundary</i>		<i>Boundary</i> , merupakan batasan dari sebuah sistem.



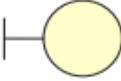


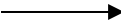
#### b) *Sequence Diagram*

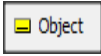

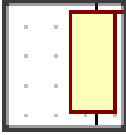
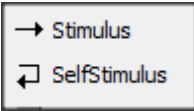
*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Jumlah banyaknya diagram yang harus digambar minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau semua *use case* yang didefinisikan interaksi jalannya pesan sudah dicakup pada *sequence diagram*[23]. Sehingga dapat disimpulkan bahwa semakin banyaknya *use case* yang didefinisikan maka *sequence diagram* yang dibuat juga semakin banyak.

Berikut adalah simbol-simbol yang ada pada diagram *sequence*:

**Tabel 2. 4** Simbol *Sequence Diagram*

Nama	Gambar	Deskripsi
------	--------	-----------

Aktor / Actor		Aktor adalah orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Jadi, meskipun simbol dari aktor adalah gambar orang, tetapi aktor tersebut belum tentu merupakan orang.
<i>Entity Class</i>		Menggambarkan hubungan kegiatan yang berinteraksi dengan sistem.
<i>Boundary Class</i>		Menggambarkan sebuah penggambaran dari form.
<i>Control Class</i>		Menggunakan penghubung antara <i>boundary</i> dengan tabel.
<i>Lifeline</i>		Menggambarkan tempat mulai dan berakhirnya sebuah pesan.
<i>Line Message</i>	<p data-bbox="434 1283 555 1310">A Message</p> 	Menggambarkan pengiriman pesan.








Objek		Menyatakan yang berisikan pesan.
Garis Hidup ( <i>Lifeline</i> )		Simbol ini menyatakan kehidupan suatu objek.
Waktu Aktif		Simbol ini menyatakan bahwa objek dalam keadaan aktif dan berinteraksi. Semua yang berhubungan dengan waktu aktif adalah sebuah tahapan yang dilakukan di dalamnya aktor tidak memiliki waktu aktif.
Stimulus		Menggambarkan suatu objek mengirimkan pesan untuk menjalankan operasi yang ada pada objek.

### c) *Activity Diagram*

Menurut Rosa dan Shalahuddin, (2014) mengemukakan: *Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Perlu diperhatikan bahwa diagram

aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas.

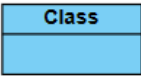

**Tabel 2. 5** Simbol *Activity Diagram*

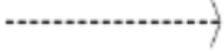
<b>Notasi</b>	<b>Keterangan</b>	<b>Simbol</b>
<i>Initial State</i>	Titik awal untuk memulai suatu aktivitas	
<i>Final State</i>	Titik akhir untuk mengakhiri suatu aktivitas	
<i>Activity</i>	Menandakan sebuah aktivitas	
<i>Decision Node</i>	Pilihan untuk pengambilan keputusan	
<i>Control Flow</i>	Arus Aktivitas	
<i>Fork Join</i>	Menunjukkan kegiatan kegiatan menggabungkan dua <i>panel activity</i> menjadi satu atau satu <i>panel activity</i> menjadi dua	
<i>Note</i>	Catatan khusus untuk sebuah aktivitas	

**d) Class diagram**

Menurut Rosa dan Shalahuddin (2013) mengemukakan: Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Berikut adalah simbol-simbol yang ada pada diagram Kelas.

**Tabel 2. 6** Simbol *Class Diagram*

<b>Notasi</b>	<b>Keterangan</b>	<b>Simbol</b>
<i>Object</i>	Blok-blok pembangun program. Bagian atas <i>class</i> menunjukkan namadari <i>class</i> , bagian tengah mengindikasikan atribut dari <i>class</i> , dan bagian bawah mendefinisikan <i>method</i> dari sebuah <i>class</i>	
<i>Activatio n</i>	Menunjukkan <i>relationship</i> atau hubungan antar <i>class</i>	

<i>Lifeline</i>	Menunjukkan ketergantungan antara satu <i>class</i> dengan <i>class</i> yang lain	
-----------------	---	---

e) **Flowchart**


*Flowchart* merupakan gambaran urutan logika dari suatu prosedur dalam memecahkan masalah. *Flowchart* juga merupakan simbol-simbol tertentu yang digambarkan sebagai langkah-langkah unruk menyelesaikan suatu masalah. *Flowchart* ini akan menunjukkan alur di dalam program secara logika[29]. Dalam perancangan *flowchart* terdapat aturan-aturan yaitu:

1. *Flowchart* digambarkan dengan orientasi dari atas ke bawah dan dari kiri ke kanan.
2. Setiap proses atau kegiatan dalam *flowchart* harus dinyatakan dengan eksplisit/jelas.
3. Pada proses *flowchart* harus dimulai dari satu *start state* dan berakhir pada satu atau lebih terminal akhir.
4. Menggunakan *connector* dengan label yang sama untuk menunjukkan keterhubungan antar *path* algoritma yang terpotong, sebagai contoh, akibat pindah atau ganti halaman[29].



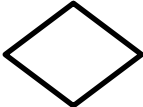
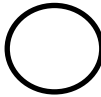
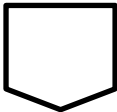
Tujuan dari *flowchart* yaitu untuk memberikan gambaran suatu tahap penyelesaian masalah secara sederhana, rapi, jelas, dan terurai menggunakan simbol-simbol standar[29].




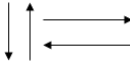
Berikut ini adalah simbol-simbol pada *flowchart*:

**Tabel 2. 7** Simbol *Flowchart*

Nama	Gambar	Deskripsi
Terminal		Terminal menyatakan permulaan atau



		akhir dari suatu program.
<i>Input / Output</i>		<i>Input / Output</i> menyatakan proses <i>input</i> dan <i>output</i> tidak tergantung dengan jenis peralatannya.
<i>Process</i>		<i>Process</i> menyatakan suatu proses yang dilakukan oleh komputer.
<i>Decision</i>		<i>Decision</i> menyatakan suatu kondisi tertentu yang menghasilkan dua kemungkinan jawabannya: ya / tidak.
<i>Connector</i>		<i>Connector</i> merupakan sambungan dari proses ke proses lain dalam halaman yang sama.
<i>Offline Connector</i>		<i>Offline Connector</i> merupakan sambungan dari proses ke proses lain

		dalam halaman yang berbeda.
<i>Predefined Process</i>		<i>Predefined Process</i> merupakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan harga awal.
<i>Punched Card</i>		<i>Punched Card</i> dapat dikatakan input yang berasal dari kartu atau <i>output</i> yang ditulis ke kartu.
<i>Document</i>		<i>Document</i> berfungsi untuk mencetak jalannya arus suatu proses.
<i>Flow</i>		<i>Flow</i> dapat menggambarkan jalannya suatu proses.

### 2.2.16 Rekayasa Web

Rekayasa *web* adalah sebuah aplikasi yang menggunakan pendekatan sistematis, disiplin, dan terukur untuk pengembangan, operasi dan pemeliharaan aplikasi berbasis *web*. Pada definisi lain disebutkan rekayasa *web* adalah sub disiplin dari rekayasa perangkat lunak yang membantu menyediakan metodologi untuk merancang, mengembangkan, memelihara, dan melibatkan aplikasi *web*[27]. Munculnya disiplin rekayasa *web* ini menunjukkan suatu kebutuhan yang dipusatkan pada

kesuksesan pengembangan aplikasi dan sistem berbasis *web*. Rekayasa *web* harus menjadi bidang yang multidisipliner dengan masukan berbagai bidang ilmu seperti:

1. Interaksi manusia dengan komputer;
2. Antarmuka pengguna;
3. Perancangan dan analisis sistem;
4. Rekayasa Perangkat Lunak;
5. Rekayasa kebutuhan;
6. Rekayasa hipermedia;
7. Struktur informasi;
8. Pengujian;
9. Pemodelan dan simulasi;
10. Manajemen proyek[27].

Prinsip yang mendasari aplikasi internet dan web diciptakan dengan skala besar, memiliki beberapa disiplin sebagai berikut :

#### 1. Rekayasa Jaringan (*Network Engineering*)

Rekayasa jaringan memiliki keterkaitan dengan analisis, teori, dan konstruksi jaringan komunikasi komputer. Seorang perancang *web* memerlukan ilmu dasar tentang rekayasa jaringan supaya dapat memahami interaksi aplikasi *web* dengan internet yang lebih luas. Ilmu dasar tentang rekayasa jaringan seperti lapisan fisik, lapisan internet, lapisan *transport*, lapisan aplikasi, dan kinerja[27].

#### 2. Rekayasa Perangkat Lunak (*Software Engineering*)

Proses yang akan digunakan dalam penciptaan perangkat lunak merupakan fokus yang harus dimiliki oleh rekayasa perangkat lunak. Aplikasi *web* yang besar adalah aktivitas pengembangan perangkat lunak yang kompleks. Rekayasa perangkat lunak mencakup hal-hal seperti proses, kebutuhan, arsitektur, desain, pengujian, dan lingkungan[27].

#### 3. Basis Data dan Sistem Penyimpanan

Basis data dan sistem penyimpanan adalah dasar dari aplikasi *web*. pengalaman dari prinsip-prinsip dan aplikasi basis data serta sistem penyimpanan mengizinkan seorang perancang *web* untuk mendesain data dalam basis data. Prinsip basis data dan penyimpanan mencakup hal-hal seperti desain data, bahasa *query* basis data, teori mesin pencari internet, dan fisik alat penyimpanan[27].

#### 4. Hipermedia

Hipermedia sangat berhubungan dengan objek informasi yang terhubung. Area hipermedia meliputi desain, visualisasi, usability, dan *tool* kolaboratif[27].

Rekayasa *web* memiliki 5 tahapan untuk dapat mengembangkan suatu perangkat lunak yaitu *customer communication*, *planning*, *modeling*, *construction*, dan *deployment*. Dalam pandangan lain, rekayasa *web* merupakan aplikasi sistematis dan pendekatan kuantitatif untuk biaya efektif analisis kebutuhan, desain, implementasi, *testing*, operasi, dan perawatan aplikasi *web*[30].

### 2.2.17 Pemrograman Berorientasi Objek (PBO)

Pemrograman Berorientasi Objek merupakan sebuah istilah yang diberikan kepada bahasa pemrograman yang menggunakan 2 teknik berorientasi atau berbasis pada sebuah objek dalam pembangunan program aplikasi tersebut. Maksudnya bahwa orientasi pembuatan program tidak lagi menggunakan orientasi linear melainkan berorientasi pada objek-objek yang terpisah-pisah. Suatu perintah dalam bahasa ini diwakili oleh sebuah objek yang didalamnya berisi beberapa perintah-perintah standar sederhana[31].

Dalam pemrograman berorientasi objek terdapat konsep dasar yang penting untuk diperhatikan dan dipahami jika kita mempelajari pemrograman :

#### A. *Class*

Di dalam sebuah objek kelas merupakan *prototype* atau rancangan yang menafsirkan variabel dan berbagai *method* pada sebuah objek tertentu. Fungsinya adalah untuk menampung konten atau isi dari sebuah program yang akan dijalankan. Dengan kata lain, kelas adalah rancangan atau *prototype* dari sebuah objek dalam pemrograman berorientasi objek. Untuk mempermudah kita memahami tentang class, salah satu contoh dari class adalah *class person*, *class animal*, *vehicle*, *fruit* dan lain-lain. Dalam bidang pemrograman, sebuah *class* haruslah dapat dikenali bahkan oleh orang bukan *programmer*, karena class merupakan dasar dari sebuah program[31].

#### B. *Object*

Yang dimaksud dengan *object* dalam pemrograman berorientasi objek adalah kumpulan perangkat lunak yang

terdiri dari metode-metode dan variabel yang saling berkaitan. Objek juga merupakan wujud nyata dari *prototype* atau rancangan yang disebutkan dalam *class* tadi[31].

### **C. Method**

*Method* atau metode merupakan suatu operasi berupa fungsi-fungsi yang dapat dikerjakan oleh suatu objek. *Method* didefinisikan pada *class*, namun dipanggil melalui objek. Contoh, pada objek buah pear: terdapat ‘metode ambil rasa’, ‘kupas kulit’ dan lain-lain[31].

### **D. Abstraksi**

Abstraksi merupakan kemampuan dari sebuah program untuk melewati aspek informasi yang diproses olehnya, atau bisa juga disebut sebagai kemampuan untuk memfokuskan pada intinya. Setiap objek dalam sistem PBO merupakan model dari “pelaku” abstrak yang dapat melakukan suatu pekerjaan, pelaporan dan perubahan pada keadaannya, dan melakukan komunikasi dengan objek lainnya yang ada di dalam sistem suatu program, tanpa memperlihatkan bagaimana hal ini diterapkan[31].

### **E. Enkapsulasi**

Ini adalah suatu bentuk dimana pengguna dari sebuah objek tidak dapat mengganti atau mengubah (memodifikasi) suatu keadaan dari sebuah objek dengan cara yang tidak layak, oleh karena itu keadaan ini hanya dapat dimodifikasi sesuai dengan ketentuan yang terdapat di dalam objek tersebut. Begitupun ketika kita ingin mengganti interfacenya haruslah memenuhi aturan dari objek yang bersangkutan. Objek yang lainnya tidak akan tergantung dengan objek tersebut[31].

*~Halaman Ini Sengaja Dikosongkan~*