

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Sistem**

Suatu sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu (Andri Kristanto, 2018).

##### **2.1.1 Konsep Dasar Sistem**

Sistem juga merupakan kumpulan elemen-elemen yang saling terkait dan bekerja sama untuk memproses masukan (input) yang ditunjukkan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran (output) yang diinginkan.

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

Suatu sistem mempunyai ciri-ciri karakteristik yang terdapat pada sekumpulan elemen yang harus dipahami dalam mengidentifikasi pembuatan sistem. Adapun karakteristik sistem (Hutahaean, 2015:3) yang dimaksud adalah sebagai berikut:

1. Komponen Sistem terdiri dari sejumlah komponen yang saling berinteraksi dan bekerja sama untuk membentuk satu kesatuan. Komponen sistem dapat berupa sub sistem atau bagian-bagian dari sistem.
2. Batasan sistem (boundary) Daerah yang membatasi antara suatu sistem dengan sistem lainnya atau dengan lingkungan luar dinamakan dengan batasan sistem. Batasan sistem ini memungkinkan sistem dipandang sebagai satu kesatuan dan juga menunjukkan ruang lingkup (scope) dari sistem tersebut.
3. Lingkungan luar sistem (environment) Apapun yang berada di luar batas dari sistem dan mempengaruhi sistem tersebut dinamakan dengan lingkungan luar sistem. Lingkungan luar yang bersifat menguntungkan wajib

dipelihara dan yang merugikan harus dikendalikan agar tidak mengganggu kelangsungan sistem.

4. Penghubung sistem (interface) Media penghubung diperlukan untuk mengalirkan sumber-sumber daya dari sub sistem ke sub sistem lainnya dinamakan dengan penghubung sistem.
5. Masukkan sistem (input) Energi yang dimasukkan ke dalam sistem dinamakan dengan masukan sistem (input) dapat berupa perawatan dan masukan sinyal. Perawatan ini berfungsi agar sistem dapat beroperasi dan masukan sinyal adalah energi yang diproses untuk menghasilkan keluaran (output).
6. Keluaran sistem (output) Hasil dari energi yang telah diolah dan diklasifikasikan menjadi keluaran yang berguna dinamakan dengan keluaran sistem (output). Informasi merupakan contoh keluaran sistem.
7. Pengolah sistem Untuk mengolah masukan menjadi keluaran diperlukan suatu pengolah yang dinamakan dengan pengolah sistem.
8. Sasaran sistem Sistem pasti memiliki tujuan atau sasaran yang sangat menentukan input yang dibutuhkan oleh sistem dan keluaran yang dihasilkan.

## **2.2 Informasi**

Informasi merupakan data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan (Andri Kristanto , 2018).

### **2.2.1 Konsep Dasar Informasi**

Menurut Sukanto dan Shalahuddin (2015:43) informasi merupakan proses lebih lanjut dari data yang sudah memiliki nilai tambah. Informasi dapat dikelompokkan menjadi 3 bagian, yaitu :

- a. Informasi Strategis

informasi ini digunakan untuk mengambil keputusan jangka panjang, yang mencakup informasi eksternal, rencana perluasan perusahaan, dan sebagainya.

b. Informasi Taktis

informasi ini dibutuhkan untuk mengambil keputusan jangka menengah, seperti informasi tren penjualan yang dapat dimanfaatkan untuk menyusun rencana penjualan.

c. Informasi Teknis

informasi ini dibutuhkan untuk keperluan operasional sehari-hari, seperti informasi persediaan stock, retur penjualan, dan laporan kas harian.

## 2.3 Sistem Informasi

Sebuah sistem informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer serta perangkat manusia yang akan mengolah data menggunakan perangkat keras dan perangkat lunak tersebut (Andri Kristanto , 2018).

### 2.3.1 Komponen Sistem Informasi

Menurut Sukanto dan Shalahuddin (2015:43) Terdapat 6 (enam) komponen dari sebuah sistem informasi. Sebagai suatu sistem ke enam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasaran. Blok-blok tersebut adalah:

a. Blok Masukan (*InputBlock*)

*Input* mewakili data yang masuk ke dalam sistem informasi. *Input* disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

b. Blok Model (*ModelBlock*)

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 tertentu untuk menghasilkan keluaran yang diinginkan.

c. Blok Keluaran (*OutputBlock*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

d. Blok Teknologi (*TechnologyBlock*)

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 keseluruhan.

e. Blok Basis data (*DatabaseBlock*)

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

f. Blok Kendali (*ControlsBlock*)

Banyak hal yang dapat merusak sistem informasi. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

## 2.4 Data

“Data adalah fakta mengenai objek, orang, dll. (data dapat dinyatakan dengan nilai, angka, deretan karakter, atau simbol)”. Abdul Kadir (2014).

Berdasarkan pengertian diatas maka dapat disimpulkan bahawa data adalah fakta-fakta pikiran atau pendapat yang belum memiliki makna yang dapat dimanfaatkan. Data ini merupakan bahan mentah dari informasi. Jadi untuk

memanfaatkan informasi yang dibutuhkan kita harus terlebih dahulu mendapatkan data

## 2.5 Basis Data

Dalam pembuatan sebuah aplikasi para pembuat aplikasi atau *programmer* menggunakan basis data atau *database* sebagai dasar dalam mengolah data atau mengelola *file-file*.

Pada umumnya basis data merupakan salah satu aspek yang sangat penting dalam sistem informasi, dimana basis data dijadikan sebagai gudang penyimpanan data yang akan diolah lebih lanjut. Basis data yaitu kumpulan data yang disusun secara sistematis didalam komputer menggunakan perangkat lunak untuk menghasilkan informasi.

Menurut Sukanto dan Shalahuddin (2015:43), “sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”.

Sedangkan, menurut Hirin dan Virgi (2011:29) memberikan batasan bahwa “*database* atau basis data yaitu sekumpulan informasi atau data secara sistematis sehingga dapat diperiksa oleh program komputer untuk memperoleh informasi dari basis data tersebut”.

Dapat ditarik kesimpulan bahwa basis data merupakan kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut.

## 2.6 MySQL

Menurut Sutanto dalam (Purba, 2015), *My Structured Query Language (MySQL)* adalah sebuah perangkat lunak sistem manajemen basis data yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi diseluruh dunia. MySQL AB membuat MySQL teredia sebagai perangkat lunak gratis dibawah lisensi GNU GPL (*General Public License*).

MySQL adalah salah satu dari sekian banyak sistem database yang merupakan terobosan solusi yang tepat dalam aplikasi database. MySQL merupakan turunan salah satu konsep utama dalam database sejak lama yaitu SQL (*Structured Query Language*).

MySQL dikembangkan pada tahun 1994 oleh sebuah perusahaan pengembang software dan konsultan database di Swedia bernama *TcX Data KonsultAB*. Tujuan awal dikembangkan MySQL adalah untuk mengembangkan aplikasi berbasis web pada client. Saat ini MySQL dapat di-download secara gratis di [www.mysql.com](http://www.mysql.com).

Sebagai database server yang memiliki konsep database modern, MySQL memiliki banyak sekali keistimewaan antara lain:

- a. *Portabilitas*, dapat berjalan stabil pada berbagai sistem operasi, seperti Windows, Linux, MacOS, dan lain-lain.
- b. *Open Source*, didistribusikan secara gratis dibawah lisensi GPL (*General Public License*).
- c. *Multiuser*, dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah.
- d. *Performance Tuning*, memiliki kecepatan yang menakjubkan dalam menangani query yang sederhana, dapat memproses lebih banyak SQL per satuan waktu.
- e. *Security*, memiliki beberapa lapisan sekuritas seperti level subnet mask, nama host, izin akses user dengan sistem perizinan yang mendetail serta password yang terenskripsi.
- f. *Scalability and Limits*, mampu menangani database dalam skala besar, dengan jumlah record lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
- g. *Connectivity*, dapat melakukan koneksi dengan client menggunakan protocol TCP/IP, Unix socket (Unix), atau Named pipes (NP).

- h. *Localisation*, dapat mendeteksi pesan kesalahan pada client dengan menggunakan lebih dari 20 bahasa.
- i. *Interface*, memiliki antarmuka (*interface*) terhadap beberapa aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
- j. *Clients and Tools*, dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi database, dan pada setiap tool yang ada disertakan petunjuk online.

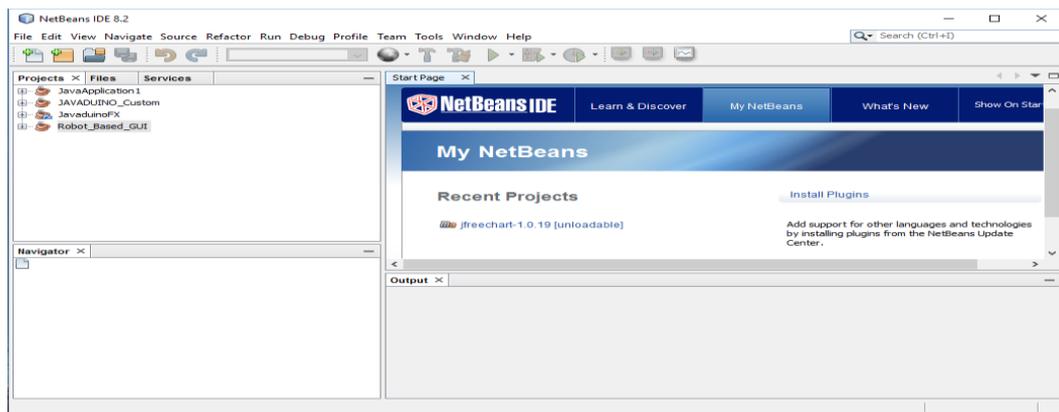
## 2.7 Netbeans

NetBeans adalah *Integrated Development Environment* (IDE) berbasis Java dari Sun Microsystems yang berjalan di atas *Swing*. *Swing* adalah sebuah teknologi Java untuk pengembangan aplikasi *desktop* yang dapat berjalan di berbagai macam *platforms* seperti *Windows*, *Linux*, *Mac OS X* dan *Solaris* (Heiko Bock, 2009).

Suatu IDE adalah lingkup pemrograman yang diintegrasikan kedalam suatu aplikasi perangkat lunak yang menyediakan pembangun *Graphic User Interface* (GUI), suatu *text* atau kode *editor*, suatu *compiler* atau *interpreter* dan suatu *debugger*. NetBeans merupakan *software development* yang *Open Source*, dengan kata lain *software* ini di bawah pengembangan bersama dan bebas biaya. NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra. Sun Microsystems mendirikan proyek kode terbuka NetBeans pada bulan Juni 2000 dan terus menjadi sponsor utama. Saat ini terdapat dua produk : NetBeans IDE dan Netbeans *Platform* (Heiko Bock, 2009).

NetBeans IDE adalah sebuah lingkungan pengembangan sebuah *tools* untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program. NetBeans IDE ditulis dalam Java, namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas Netbeans IDE (Heiko Bock, 2009).

Menurut (Heiko Bock, 2009) NetBeans IDE adalah sebuah produk bebas dengan tanpa batasan bagaimana digunakan. Tersedia juga Netbeans *Platform* sebuah fondasi yang modular dan dapat diperluas yang dapat digunakan sebagai perangkat lunak dasar untuk membuat aplikasi *desktop* yang besar. Mitra ISV menyediakan *plug-in* bernilai tambah yang dapat dengan mudah diintegrasikan ke dalam *platform* dan dapat juga digunakan untuk membuat *tools* dan solusi sendiri. Produk ini merupakan kode terbuka (*open source*) dan bebas (*free*) untuk penggunaan komersial dan non komersial. Kode sumber tersedia untuk guna ulang dengan lisensi *Common Development and Distribution License* (CDDL).



**Gambar 2.1.**Tampilan NetBeans IDE

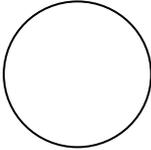
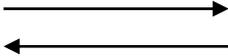
## 2.8 Alat-Alat Pengembangan Sistem

Pada sub bab ini akan diuraikan mengenai alat-alat yang digunakan dalam pengembangan sistem antara lain bagan alir dokumen, *data flow diagram*, dan *Entity Relationship Diagram*.

### 2.8.1 Pengertian *Data Flow Diagram* (DFD)

Sutabri (2012:116), DFD adalah suatu *network* yang menggambarkan suatu sistem komputerisasi, manualisasi, atau gabungan dari keduanya, yang penggambarannya disusun dalam bentuk kumpulan komponen sistem yang saling berhubungan sesuai dengan aturan mainnya.

**Tabel 2.1** Simbol DFD

No	Simbol	Keterangan
1.		<p><i>EXTERNAL ENTITY</i></p> <p>Simbol ini digunakan untuk menggambarkan asal atau tujuan data.</p>
2.		<p>PROSES</p> <p>Simbol ini digunakan untuk proses pengolahan data atau transformasi data.</p>
3.		<p><i>DATA FLOW</i></p> <p>Simbol ini digunakan untuk menggambarkan aliran data yang berjalan</p>
4.		<p><i>DATA STORE</i></p> <p>Simbol ini digunakan untuk menggambarkan data <i>flow</i> yang sudah disimpan atau diarsipkan.</p>

Sumber: Sutabri (2012:117)

Sukanto dan Shalahuddin (2013:72) menjelaskan tentang tahapan-tahapan perancangan dengan menggunakan *Data Flow Diagram* yaitu:

1. Membuat DFD Level 0 atau sering disebut juga *Context Diagram*

DFD Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD Level 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

## 2. Membuat DFD Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. DFD Level 1 merupakan hasil *breakdown* DFD Level 0 yang sebelumnya sudah dibuat.

## 3. Membuat DFD Level 2

Modul-modul pada DFD Level 1 dapat di-*breakdown* menjadi DFD Level 2. Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu di-*breakdown* lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 yang di-*breakdown*.

## 4. Membuat DFD Level 3 dan seterusnya

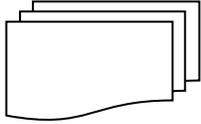
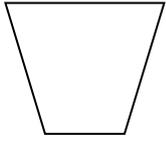
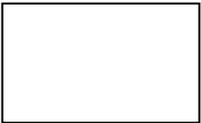
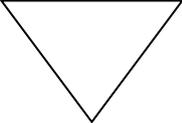
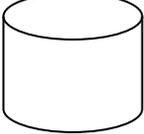
DFD Level 3,4,5, dan seterusnya merupakan *breakdown* dari modul pada DFD Level di-atasnya. *Breakdown* pada level 3,4,5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau 2.

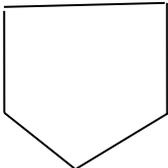
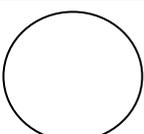
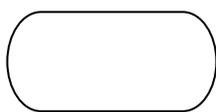
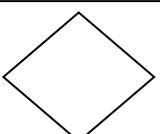
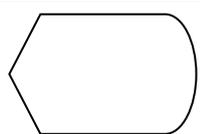
### **2.8.2 Pengertian *Block Chart***

Kristanto (2018), *blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *Blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.

Kristanto (2018), simbol-simbol yang sering digunakan dalam *blockchart* dapat dilihat pada tabel berikut ini:

**Tabel 2.2** Simbol *Blokchart*

No	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan
2.		Multi dokumen
3.		Proses Manual
4.		Proses dilakukan oleh komputer.
5.		Menandakan dokumen yang diarsipkan (arsip manual)
6.		Data penyimpanan (Storage)
7.		Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik.

8.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain.
9.		Terminal yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.
10.		Terminal yang menandakan awal dan akhir dari suatu aliran.
11.		Pengambilan keputusan (Decision).
12.		Layar peraga (monitor).
13.		Pemasukkan data secara manual.

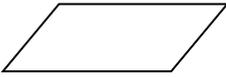
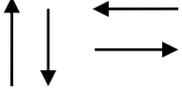
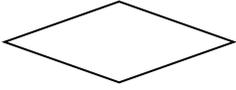
*Sumber: Kristanto (2018)*

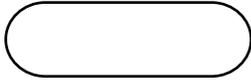
### 2.8.3 Pengertian *Flowchart*

Supardi (2013:58), *flowchart* merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program.

Supardi (2013:58), simbol-simbol yang digunakan dalam *Flowchart*, yaitu:

**Tabel 2.3** Simbol-simbol dalam *Flowchart*

No.	Simbol	Keterangan
1.	Simbol <i>input/output</i> 	Simbol <i>input</i> atau <i>output</i> ( <i>input/output symbol</i> ) digunakan untuk mewakili data <i>input/output</i>
2.	Simbol proses 	Simbol proses digunakan untuk mewakili suatu proses
3.	Simbol garis alir 	Simbol garis alir ( <i>flow lines symbol</i> ) digunakan untuk menunjukkan arus dari proses
4.	Simbol penghubung 	Simbol penghubung ( <i>connector symbol</i> ) digunakan untuk menunjukkan sambungan dari bagan alir yang terputus di halaman yang masih sama atau di halaman lainnya
5.	Simbol keputusan 	Simbol keputusan ( <i>decision symbol</i> ) digunakan untuk suatu penyeleksian kondisi di dalam program
6.	Simbol proses terdefinisi 	Simbol proses terdefinisi ( <i>predefined process symbol</i> ) digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain.
7.	Simbol persiapan 	Simbol persiapan ( <i>preparation symbol</i> ) digunakan untuk memberi nilai awal suatu

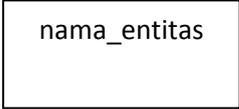
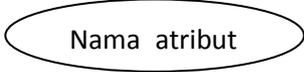
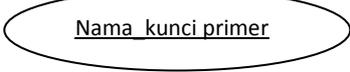
		besaran
8.	Simbol titik terminal 	Simbol titik terminal (terminal point symbol) digunakan untuk menunjukkan awal dan akhir dari suatu proses

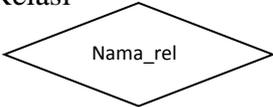
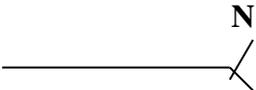
Sumber: Supardi (2013:58)

### 2.2.4 Pengertian *Entity Relationship Diagram* (ERD)

Menurut Sukamto dan Shalahuddin (2013:50), “ERD (*Entity Relationship Diagram*) digunakan untuk pemodelan basis data relasional”.

**Tabel 2.4** Simbol ERD

No	Simbol	Deskripsi
1	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 kata benda dan belum merupakan nama tabel.
2	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
3	Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam sebuah entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu

		kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).
4	Atribut multinilai/ <i>multivalued</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
6	Asosiasi / <i>association</i> 	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakai. Kemungkinan jumlah maksimum keterhubungan antara entitas yang lain disebut kardinalitas.

Sumber: Sukamto dan Shalahuddin (2013:50)

### 2.2.5 Kamus Data

Sukamto dan Shalahuddin (2013:73), kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan).

**Table 2.5** Simbol-simbol dalam Kamus Data

No.	Simbol	Arti
1.	=	disusun atau terdiri atas
2.	+	Dan
3.	[]	baik ...atau...
4.	{ } <sup>n</sup>	n kali diulang/ bernilai banyak
5.	( )	data operasional
6.	*...*	batas komentar

*Sumber: Sukamto dan Shalahuddin (2013:73)*

## 2.9 Metodologi pengembangan sistem

Shalahudin (2014:28) menjelaskan tentang metode pengembangan sistem yaitu metode air terjun. Metode air terjun (waterfall) sering juga disebut model sekuensial linier (sequential linear) atau alur hidup klasik (classic life cycle). Model air terjun menyediakan pendekatan alur hidup terurut mulai dari analisis, desain, pengodean, pengujian, dan pemeliharaan.

Tahapan pengembangan pada metode Waterfall Shalahudin (2014:28).

### 2.9.1 Analisis

Analisis Tahap analisis dilakukan secara intensif untuk menspesifikasikan kebutuhan sistem agar dapat dipahami sistem seperti apa yang dibutuhkan oleh user.

### 2.9.2 Design

Tahap desain adalah proses multi langkah yang fokus pada desain pembuatan program sistem termasuk struktur data, arsitektur sistem, representasi antarmuka, dan prosedur pengkodean (Coding). Tahap ini mentranslasi kebutuhan sistem dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

### 2.9.3 Coding (Pengkodean)

Pada tahap pengkodean, desain harus ditranslasikan ke dalam program sistem. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

### 2.9.4 Testing (Pengujian)

Tahap pengujian fokus pada sistem dari segi logika 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. Pada tahap pengujian digunakan alat pengujian aplikasi black box testing. Menurut Iskandaria (2012), Pengujian blackbox (blackbox testing) adalah salah satu metode pengujian perangkat lunak yang berfokus pada sisi fungsionalitas, khususnya pada input dan output aplikasi (apakah sudah sesuai dengan apa yang diharapkan atau belum). Tahap pengujian merupakan salah satu tahap yang harus ada dalam sebuah siklus pengembangan perangkat lunak.

### 2.9.5 Maintenance

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