

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Konsep Dasar Program**

Menurut Kusdiawan (2010:2) “program adalah susunan instruksi dalam suatu bahasa pemrograman yang menjelaskan kepada mesin (komputer) bagaimana melakukan sesuatu proses terhadap data. Ada dua elemen utama sebuah program yaitu *data* dan *instruksi*. Data diwakili oleh variabel dan instruksi diwakili oleh *command* (perintah) atau *keyword* (kata kunci). Setiap variabel harus diberi nama yang dilengkapi dengan *tipe-data* dan melalui suatu instruksi diberikan nilai”.

Menurut Kadir (2015:2) “program adalah kumpulan instruksi yang ditujukan untuk komputer supaya peralatan tersebut dapat melakukan tindakan-tindakan yang dikehendaki oleh pemakai program (*user*)”.

Dari keterangan diatas dapat ditarik kesimpulan bahwa program komputer sebenarnya mempunyai peran yang sangat penting dalam menentukan kinerja komputer layak disebut hebat dan berguna bagi individu, institusi, dan industri-industri. Agar dapat berdaya guna maka suatu komputer harus mampu menyelesaikan pekerjaan yang dilakukan secara manual.

#### **2.2. Peralatan Pendukung (*Tools Program*)**

Adapun peralatan pendukung (*Tools Program*) yang digunakan dalam pembuatan program ini dijelaskan dalam sub-sub bab berikut:

### 2.2.1. Jasa

Menurut Sugiarto dan Herlambang (2007:12) "jasa merupakan layanan seseorang/instansi/barang yang akan memenuhi kebutuhan masyarakat".

Menurut Sudarso (2016:31) "jasa adalah sesuatu yang tidak berwujud dimana tindakan atau unjuk kerja yang ditawarkan oleh salah satu pihak ke pihak lain dan tidak menyebabkan perpindahan kepemilikan apapun".

Berdasarkan pendapat menurut para ahli di atas penulis dapat menarik kesimpulan bahwa jasa merupakan suatu layanan kepada masyarakat yang tidak berwujud namun dapat dirasakan manfaatnya bagi masyarakat tersebut.

### 2.2.2. ERD (*Entity Relationship Diagram*)

Menurut Fatta (2007:121) "ERD adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis. Entitas biasanya menggambarkan jenis informasi yang sama. Dalam entitas digunakan untuk menghubungkan antar entitas yang sekaligus menunjukkan hubungan antar data".

Menurut Pratama (2014:49) "ERD (*Entity Relationship Diagram*) adalah diagram yang menggambarkan keterkaitan antartabel beserta dengan *field-field* didalamnya pada suatu *database* sistem. Sebuah *database* memuat minimal sebuah tabel dengan sebuah atau beberapa buah *field* (kolom) didalamnya. Namun pada kenyataannya, database lebih sering memiliki lebih dari satu buah tabel (dengan beberapa *field* didalamnya). Setiap tabel umumnya memiliki keterkaitan hubungan. Keterkaitan antar tabel ini biasanya disebut dengan Relasi".

Berdasarkan pendapat menurut para ahli di atas penulis dapat menarik kesimpulan bahwa ERD merupakan diagram yang digunakan untuk menunjukkan aturan-aturan bisnis pada sistem informasi yang akan dibangun.

### **2.2.3.LRS *Logical Record Structure***

Menurut Frieyadie (2007:13) “LRS merupakan hasil dari pemodelan *Entity Relational Ship* (ER) beserta atributnya sehingga bisa terlihat hubungan-hubungan antar entitas”.

Menurut Hasugian (2012:608) “*Logical record structure* adalah sebuah model sistem yang menggambarkan dengan sebuah diagram ER akan mengikuti pola atau aturan pemodelan tertentu dalam kaitannya dengan konversi ke LRS, maka perubahan yang terjadi adalah mengikuti aturan-aturan berikut ini : setiap entitas akan diubah kebentuk kotak, sebuah atribut relasi disatukan dalam sebuah kotak bersama entitas jika hubungan yang terjadi pada diagram-ER 1:M (relasi berstu dengan *cardinality* M) atau tingkat hubungan 1:1 (relasi bersatu dengan *cardinality* yang paling membutuhkan referensi), sebuah relasi dipisah dalam sebuah kotak tersendiri (menjadi entitas baru) jika tingkat hubungan M:M (*many to many*) dan memiliki *foreign key* sebagai *primary key* yang diambil dari kedua entitas yang sebelumnya saling berhubungan”.

Menurut pendapat para ahli diatas penulis dapat menarik kesimpulan bahwa LRS merupakan sebuah model sistem yang menghubungkan antara entitas satu ke entitas yang lain.

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

Menurut Pratama (2014:48) ”UML (*Unified modeling language*) adalah standarisasi internasional untuk notasi dalam bentuk grafik, yang menjelaskan tentang analisis dan desain perangkat lunak yang dikembangkan dengan pemrograman berorientasi objek”.

Menurut Mulyani (2016:) “*Unified modeling language* adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem”.

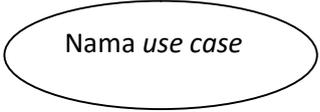
Menurut pendapat para ahli diatas penulis dapat menarik kesimpulan bahwa UML merupakan sebuah pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan mendokumentasikan dari sistem perangkat lunak.

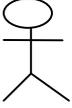
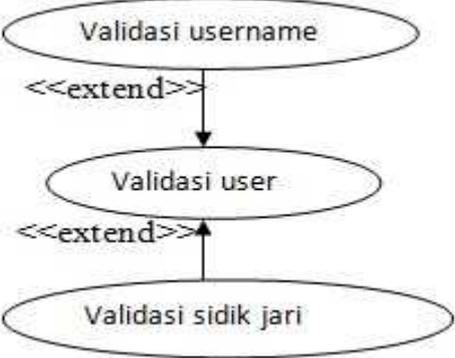
### 2.2.5. Use Case

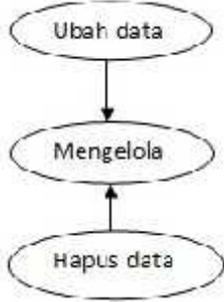
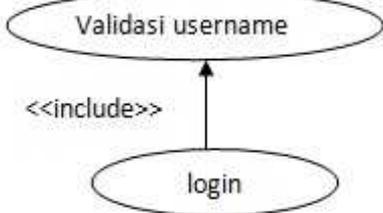
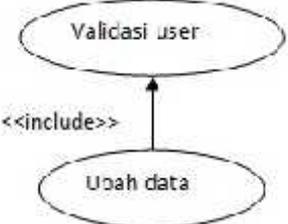
Menurut Fatta (2007:91) “*Use case* adalah metode berbasis teks untuk menggambarkan dan mendokumentasikan proses yang kompleks. *Use case* menambahkan detail untuk kebutuhan yang telah dituliskan pada definisi sistem kebutuhan”.

Menurut Rosa dan Shalahudin (2015:155) “*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat”.

Adapun simbol yang digunakan dalam *use case* menurut Rosa dan Shalahuddin (2015:156), yaitu:

Simbol	Deskripsi
<p data-bbox="316 1697 432 1731"><i>Use case</i></p> 	<p data-bbox="774 1697 1359 1877">Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>.</p>

<p>Aktor / actor</p>  <p>Nama aktor</p>	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda dari awal frase nama aktor.</p>
<p>Asosiasi /association</p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p>Ekstensi /<i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan kesebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inhertance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.</p>  <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
<p>Generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p>

	 <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p><code>&lt;&lt;include&gt;&gt;</code>  .....&gt;</p> <p><code>&lt;&lt;uses&gt;</code>  —————&gt;</p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> 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.</p> <p>Ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>:</p> <p>) <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</p>  <p>) <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</p>  <p>Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber : Rosa dan Shalahuddin(2015:27)

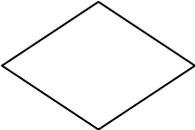
**Gambar II.I Simbol *use case***

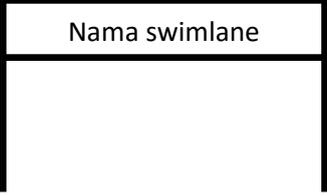
Menurut pendapat para ahli diatas penulis dapat menarik kesimpulan bahwa *use case* merupakan pemodelan untuk mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

### 2.2.6. Activity Diagram

Menurut Rosa dan Shalahuddin (2015:161) “Diagram aktivitas atau *activity diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. *Activity diagram* juga bisa diartikan rancangan proses bisnis dimana setiap urutan aktifitas yang digambarkan merupakan proses bisnis sistem. Dengan demikian, *activity diagram* merupakan gambaran aliran yang melalui program dari titik mulai sampai ke titik akhir menurut Tantra (2012:150)”.

Adapun simbol yang digunakan dalam *activity diagram* menurut Rosa dan Shalahuddin (2015:162), yaitu:

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

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

Sumber : Rosa dan Shalahuddin

### Gambar II.II Simbol *activity diagram*

Berdasarkan pendapat para ahli diatas penulis dapat menarik kesimpulan bahwa *Activity Diagram* merupakan diagram yang menggambarkan suatu aliran kerja dari sebuah sistem.

### 2.2.7. MySQL

Menurut Zakiyudin (2008:94) mengatakan bahwa “MySQL merupakan tipe data relasional yang artinya MySQL menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan”.

Menurut Anhar (2010:45) “MySQL (*My Structure Query Language*) adalah salah satu *DataBase Management System* (DBMS) dari sekian banyak DBMS seperti Oracle, MS SQL, Postagre SQL, dan lainnya”.

Dari definisi para ahli di atas, dapat disimpulkan bahwa MySQL merupakan tipe data yang bersifat *Open Source* sehingga umum digunakan di dunia.

### 2.2.8. Visual Basic

Menurut Kursini (2007:171) *Visual Basic* adalah salah satu bahasa pemrograman komputer. Bahasa pemrograman adalah perintah-perintah yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Bahasa pemrograman *Visual Basic*, yang dikembangkan oleh *Microsoft* sejak tahun 1991, merupakan pengembangan dari pendahulunya, yaitu bahasa pemrograman BASIC (*Benginner's All-purpose Symbolic Instruction Code*) yang dikembangkan pada era 1950an. *Visual Basic* merupakan salah satu *development tool*, yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi *Windows*.

Menurut Madcoms (2008:2) “*Microsoft Visual Basic* merupakan salah satu aplikasi pemrograman visual yang cukup populer dan mudah untuk dipelajari. Basic bahasa pemrograman yang digunakan dalam *Visual Basic* adalah bahasa

BASIC (*Beginners All-purpose Symbolic Instruction Code*) yang merupakan salah satu bahasa pemrograman tingkat tinggi yang sederhana dan mudah dipelajari”.

Menurut pendapat para ahli diatas penulis dapat menyimpulkan bahwa *Microsoft Visual Basic* ialah bahasa pemrograman tingkat tinggi yang sederhana dan mudah untuk dipelajari sehingga pengguna dapat membuat program aplikasi dalam sebuah form baik aplikasi kecil, sederhana hingga ke aplikasi pengolahan database.

### **2.2.9. Crystal Report**

Menurut Kursini (2007:264) “*Crystal Report* merupakan program dapat digunakan untuk membuat, menganalisis dan menerjemahkan informasi yang terkandung dalam *database* atau program ke dalam berbagai jenis laporan yang sangat fleksibel”.

Menurut Madcoms (2008:234) “*Crystal report* merupakan program yang terpisah dengan dengan program *Microsoft Visual Basic* tetapi keduanya dapat dihubungkan (*linkage*). Membuat laporan dengan *crystal report* hasilnya lebih baik dan lebih mudah, karena pada *crystal report* banyak tersedia objek-objek maupun komponen yang mudah digunakan”.

Berdasarkan pendapat para ahli diatas penulis dapat menarik kesimpulan bahwa *Crystal Report* merupakan suatu program yang terpisah dengan program *Microsoft Visual Basic* yang fungsinya untuk mencetak laporan.

### 2.2.10. XAMPP

Menurut Aryanto (2016:2) “XAMPP merupakan sebuah aplikasi perangkat lunak pemrograman dan database yang didalamnya terdapat berbagai macam pemrograman seperti; Apache HTTP Server, MySQL Database, bahasa pemrograman PHP dan Perl”.

Menurut Pratama (2014:440) “XAMPP adalah aplikasi *web server* bersifat instan (siap saji) yang dapat digunakan baik di sistem operasi Linux maupun di sistem operasi Windows”.

Berdasarkan pendapat para ahli diatas penulis dapat menarik kesimpulan bahwa XAMPP merupakan aplikasi *web server* yang fungsinya sebagai penghubung antara *database* ke bahasa pemrograman.