

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Program

Konsep dasar program berisi tentang teori-teori yang berhubungan dengan definisi program, aplikasi, *database* dan pengertian sewa.

2.1.1. Pengertian Program

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

Sedangkan menurut Kusdiawan (2010:2) “program adalah instruksi yang terdiri dari sekumpulan kode yang diberikan kepada komputer, agar komputer dapat melaksanakan tugas-tugas tertentu”.

Dari pengertian diatas dapat diambil kesimpulan pengertian program adalah perintah yang diberikan kepada komputer agar komputer melakukan tugas yang dikehendaki oleh pemakai.

2.1.2. Pengertian Aplikasi

Menurut Wasito (2010:1) “aplikasi adalah suatu unit perangkat lunak yang dirancang untuk memenuhi kebutuhan aktivitas seperti sistem perniagaan, pelayanan masyarakat, informasi kependudukan atau semua proses yang hampir semua manusia lakukan”.

Sedangkan Hendrayudi (2009:143) menuturkan “aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (khusus)”.

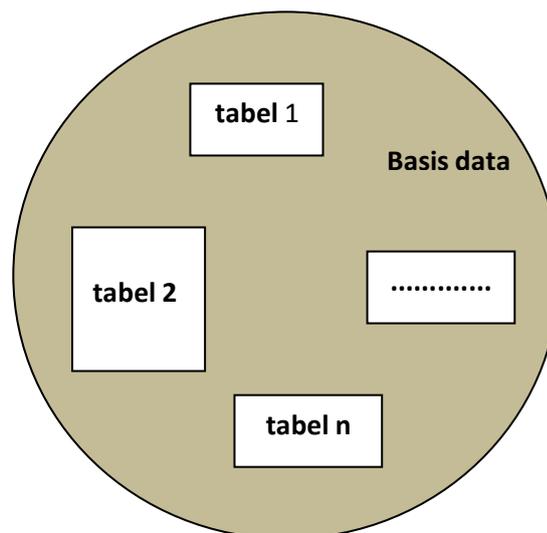
Dari pengertian tersebut dapat diambil kesimpulan bahwa aplikasi adalah sebuah perangkat lunak yang dibuat untuk memudahkan aktifitas sehari-hari manusia.

2.1.3. Pengertian Basis Data (*Database*)

Menurut Musliadi (2013:15) “*database* atau yang sering dikenal dengan basis data merupakan sekumpulan data atau informasi yang berhubungan dengan satu objek, topik, atau tujuan tertentu”.

Wasito (2010:2) menerangkan “secara sederhana *database* didefinisikan sebagai kumpulan informasi yang terintegritas, diorganisasikan dan disimpan dalam suatu cara yang memudahkan pengambilan kembali”. Sedangkan menurut Sukamto dan M. Shalahuddin (2014:43) “basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat”.

Berdasarkan penjelasan diatas penulis mengambil kesimpulan bahwa *database* adalah sekumpulan data atau informasi yang dapat diolah dan diakses dengan mudah.



Sumber: Sukamto dan M. Shalahuddin (2014:44)

Gambar II.1 Ilustrasi Basis Data

2.1.4. Pengertian Sewa

Pengertian sewa menurut PBI dalam Ikit (2015:92) “adalah transaksi sewa menyewa terhadap suatu barang dan atau upah mengupah atas suatu jasa dalam waktu tertentu melalui pembayaran sewa atau imbalan jasa”.

Pengertian sewa menurut Kamus Besar Bahasa Indonesia dalam As dan Leni Fitriani (2016:199) adalah “pemakaian sesuatu dengan membayar uang sewa, uang yang dibayarkan karena memakai atau meminjam sesuatu, yang boleh dipakai dengan membayar uang dengan uang”.

Berdasarkan pengertian diatas penulis membuat kesimpulan bahwa sewa adalah suatu transaksi dimana satu pihak meminjamkan suatu barang kepada pihak lain dengan syarat atau perjanjian yang harus dipenuhi oleh kedua belah pihak.

2.2. Peralatan Pendukung (*Tools Program*)

Pada bagian ini penulis menerangkan mengenai pengertian dari peralatan pendukung yang digunakan dalam pembuatan aplikasi penyewaan.

2.2.1. MySQL

Menurut Faizal dan Irnawati (2015: 4) “MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa inggris: *database management system*) atau DBMS yang *multithread, multiuser*, dengan sekitar 6(enam) juta instalasi diseluruh dunia”.

Sedangkan menurut Kustianingsih dan Devie Rosa Anamisa (2011:145) “MYSQL merupakan sebuah basis data yang mengandung satu atau sejumlah

tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom”

Berdasarkan pengertian dari beberapa ahli diatas penulis menarik kesimpulan bahwa MYSQL adalah suatu perangkat lunak untuk menampung basis data.

2.2.2. Language Record Structure (LRS)

Hasugian dan Ahmad Nur Shidiq (2012:608) mendefinisikan: *Language Record Structure* (LRS) adalah sebuah model sistem yang digambarkan dengan sebuah *diagram-ER* akan mengikuti pola / 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 bersatu 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 hubungannya M:M (*many to many*) dan memiliki *foreign key* sebagai *primary key* yang diambil dari kedua entitas yang sebelumnya saling berhubungan.

Sedangkan menurut Frieyadi dalam Rahmayu (2015:162) “LRS merupakan hasil dari pemodelan *Entity Relationship* (ER) beserta atributnya sehingga bisa terlihat hubungan-hubungan antar entitas”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa LRS adalah sebuah pemodelan untuk menggambarkan relasi dan atribut dengan aturan-aturan yang harus diikuti.

2.2.3. Entity Relationship Diagram (ERD)

Menurut Fathansyah dalam Faizal dan Irnawati (2015:15) “*Entity Relationship Diagram* adalah diagram yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta yang ditinjau”.

Kusrini dan Andri Koniyo (2007:99) menerangkan “ERD merupakan notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antarpenyimpanan. ERD digunakan untuk memodelkan struktur data dan hubungan antar data, karena hal ini relatif kompleks”.

Dari pendapat para ahli tersebut dapat disimpulkan bahwa pengertian ERD adalah diagram yang mendeskripsikan relasi antar entitas dilengkapi dengan atribut yang mempresentasikan seluruh fakta.

ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antardata. Pada dasarnya ada 3 macam simbol yang digunakan, yaitu (Kusrini dan Andri Koniyo, 2007:99) :

1. *Entity*.

Entity adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat. Sebagai contoh adalah barang, pemasok, pekerja dan lain-lain.

Seandainya A adalah barang maka A adalah isi dari barang, sedangkan jika B adalah seorang pelanggan maka B adalah isi dari pelanggan. Karena itu harus dibedakan antara entitas seperti A dan B dalam contoh kasus diatas. Entitas digambarkan dalam bentuk persegi empat.

2. *Attribut*.

Entitas mempunyai elemen yang disebut atribut dan berfungsi mendeskripsikan karakter entitas, misalnya atribut nama barang dari entitas barang. Setiap ERD bisa berisi lebih dari satu atribut. Atribut digambarkan dalam bentuk elips.

3. Hubungan - *relationship*.

Sebagaimana halnya entitas, hubunganpun harus dibedakan antara hubungan atau bentuk hubungan antarentitas dengan isi dari hubungan itu sendiri.

Misalnya dalam kasus hubungan antar entitas barang dan entitas pelanggan adalah menjual barang, sedangkan isi hubungannya dapat berupa tanggal jual atau yang lainnya. *Relationship* digambarkan dalam bentuk intan (*diamond*).

Jenis-jenis hubungan (Kusrini dan Andri Koniyo, 2007:101) :

1. Satu ke satu, misalnya suatu perusahaan mempunyai aturan satu sopir hanya boleh menangani suatu kendaraan karena alasan tertentu.



Sumber: Kusrini dan Andri Koniyo (2007:1014)

Gambar II.2 Relational 1 to 1

2. Satu ke banyak, atau banyak ke satu, misalnya suatu perusahaan selalu berasumsi bahwa satu pelanggan dapat membeli banyak barang.



Sumber: Kusrini dan Andri Koniyo (2007:101)

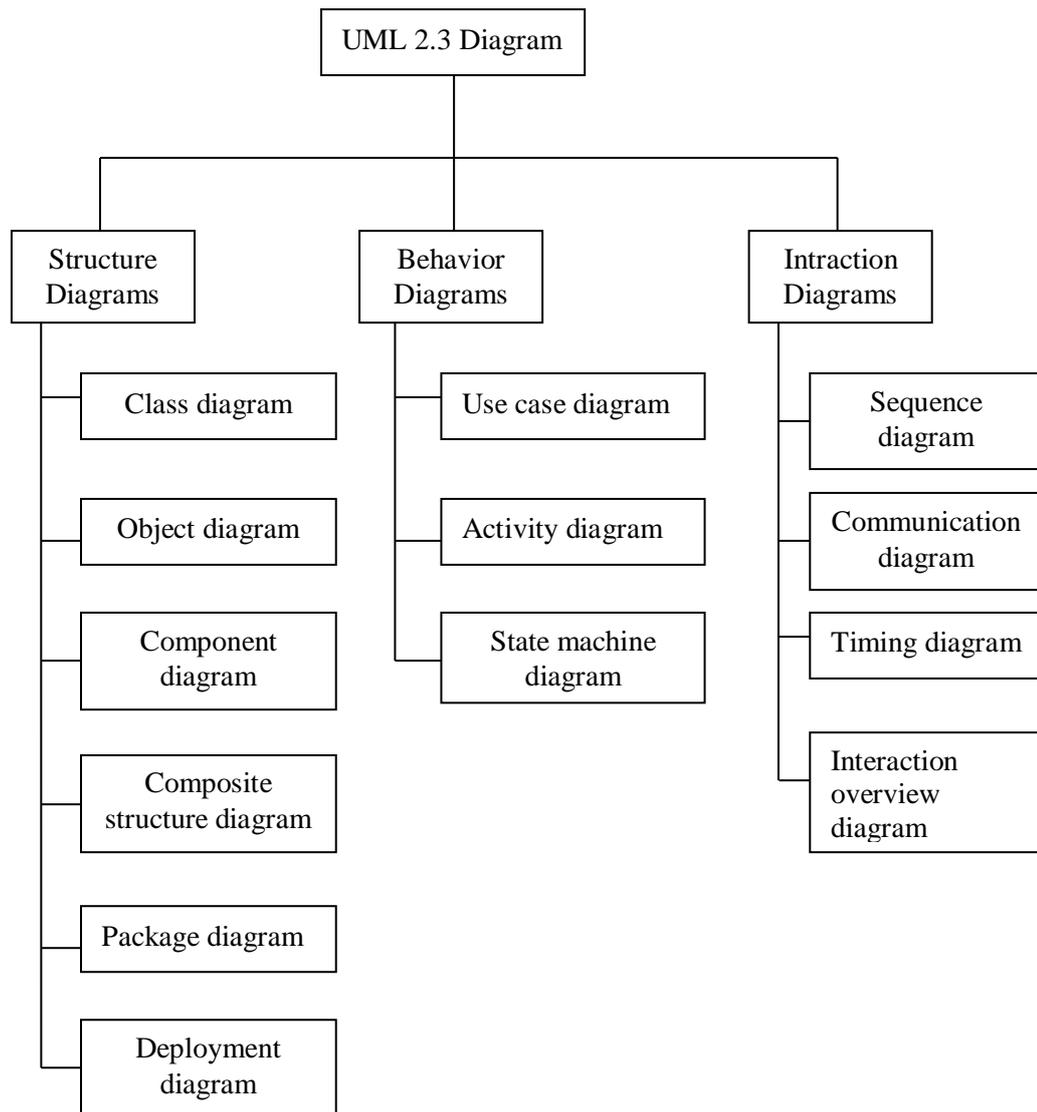
Gambar II.3 Relational 1 to M

2.2.4. *Unified Modeling Language (UML)*

Menurut Sukamto dan M. Shalahuddin (2014:137) “*Unified Modeling Language (UML)* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”.

Sugiarti (2013:34) mendefinisikan “*Unified Modelling Language (UML)* adalah sebuah ‘bahasa’ yang menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak”.

Dari beberapa pendapat para ahli diatas penulis menyimpulkan bahwa UML adalah bahasa yang digunakan untuk pemodelan sebuah sistem.



Sumber: Sukamto dan M. Shalahuddin (2014:140)

Gambar II.4 Diagram UML

Berikut ini penjelasan singkat dari pembagian model tersebut (Sukamto dan M. Shalahuddin, 2014:141):

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

2.2.5. Use Case Diagram

Menurut Sukamto dan M. Shalahuddin (2014:155) “*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat”.

Pengertian *Use case diagram* menurut Sugiarti (2013:41) adalah “pemodelan untuk menggambarkan kelakuan (*behaviour*) sistem yang akan dibuat”.

Berdasarkan pendapat para ahli diatas, penulis menyimpulkan bahwa *Use case diagram* adalah sebuah pemodelan yang dilihat dari segi kelakuan sistem yang akan dibuat.

Setiap *use case* harus dijelaskan alur prosesnya melalui sebuah deskripsi *usecase (use case description)* atau *scenario use case*. Deskripsi *use case* berisi (Sugiarti, 2013:44) sebagai berikut:

1. Nama *use case* yaitu penamaan *use case* yang menggunakan kata kerja.

2. Deskripsi yaitu penjelasan mengenai tujuan *use case* dan nilai yang akan didapatkan oleh aktor.
3. Kondisi sebelum (*pre-condition*) yaitu kondisi-kondisi yang perlu ada sebelum *use case* dilakukan.
4. Kondisi sesudah (*post-condition*) yaitu kondisi-kondisi yang sudah dipenuhi ketika *use case* sudah dilaksanakan.
5. Alur dasar (*basic flow*) yaitu alur yang menceritakan jika semua aksi yang dilakukan adalah benar atau proses yang harusnya terjadi.
6. Alur alternatif (*alternatif flow*) yaitu alur yang menceritakan aksi alternatif, yang berbeda dari alur dasar.

2.2.6. Activity Diagram

Menurut Sukamto dan M. Shalahuddin (2014:161) “diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada dalam perangkat lunak”.

Sedangkan menurut Sugiarti (2013:75):

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Dari pendapat para ahli tersebut dapat disimpulkan bahwa diagram aktivitas adalah sebuah pemodelan dilihat dari aktivitas yang dapat dilakukan oleh sistem bukan apa yang dilakukan oleh aktor.

Sugiarti (2013:77) dalam bukunya memaparkan langkah-langkah membuat diagram aktivitas adalah:

1. Buat simbol status awal ketika mengawali diagram

2. Gambarkan aksi pertama dan seterusnya sesuai aliran kegiatan sistem .
gunakan sebuah *fork* ketika berbagai aktivitas terjadi secara bersamaan.
Setelah penggabungan seluruh kegiatan paralel, harus digabungkan dengan simbol join.
3. Cabang keputusan digunakan untuk menunjukkan suatu kegiatan yang memenuhi kondisi tertentu. Seluruh pancabangan diakhiri tanda penggabungan (menggunakan tanda *decision*) sebagai akhir perilaku tersebut.
4. Akhiri diagram dengan simbol status akhir.

2.2.7. Visual Basic.NET

Menurut Kusriani dan Andri Koniyo (2007: 171) “Visual Basic adalah salah satu bahasa pemrograman komputer”.

Sedangkan menurut Hidayatullah (2014:5) memaparkan bahwa: Visual Basic.NET adalah Visual Basic yang direkayasa kembali untuk digunakan pada *platform* .NET sehingga aplikasi yang dibuat menggunakan Visual Basic.NET dapat berjalan pada sistem komputer apapun, dan dapat mengambil data dari *server* dengan tipe apapun asalkan terinstal .NET Framework.

Berikut ini perkembangan Visual Basic.NET (Hidayatullah, 2014:5):

1. Visual Basic.NET 2002 (VB 7.0)
2. Visual Basic.NET 2003 (VB 7.1)
3. Visual Basic.NET 2005 (VB 8.0)
4. Visual Basic.NET 2008 (VB 9.0)
5. Visual Basic.NET 2010 (VB 10.0)
6. Visual Basic.NET 2012 (VB 11.0)
7. Visual Basic.NET 2013

Dari pengertian diatas dapat disimpulkan bahwa Visual Basic. Net adalah visual Basic yang telah dikembangkan sehingga lebih memudahkan pemakai karena dapat mengambil data dari *server* dengan tipe apapun.

2.2.8. Crystal Report

Dalam bukunya Madcoms (2010:234) menerangkan:

Crystal Report merupakan program yang terpisah dengan Microsoft Visual Basic 6.0, 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.

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

Dari pengertian diatas dapat ditarik kesimpulan bahwa pengertian *Crystal Report* adalah sebuah program yang digunakan untuk membuat laporan.

Beberapa kelebihan dari *Crystal Report* adalah (Kusriani dan Andri Koniyo, 2007:264)

1. Pembuatan laporannya tidak terlalu rumit sehingga memungkinkan pemrogram pemula sekalipun untuk membuat laporan tanpa harus melibatkan banyak kode pemrograman.
2. Terintegrasi dengan berbagai bahasa pemrograman lain sehingga memungkinkan pemrogram memanfaatkannya dengan keahliannya sendiri-sendiri.
3. Fasilitas impor hasil laporan yang mendukung format yang populer seperti Microsoft Word, Excell, Acces, Adobe Acrobat Reader, HTML dan lain sebagainya.

2.2.9. XAMPP

Menurut Riyanto (2015:1) dalam bukunya menjelaskan “XAMPP merupakan paket PHP dan MySQL berbasis *open source*, yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis PHP, XAMPP mengombinasikan beberapa paket perangkat lunak berbeda kedalam satu paket”.

Sedangkan 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 pengertian diatas dapat ditarik kesimpulan bahwa XAMPP adalah sebuah alat yang dipakai untuk membantu mengembangkan sebuah aplikasi yang mengombinasikan beberapa paket perangkat lunak dalam satu paket.