

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Konsep Dasar Sistem**

Suatu sistem dapat diartikan sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan dan bekerja secara bersama-sama untuk melakukan suatu kegiatan guna mencapai suatu sasaran tertentu. Sistem sangat dibutuhkan dalam suatu perusahaan atau instansi pemerintahan, untuk meningkatkan kinerja dalam sebuah pekerjaan yang sedang berjalan.

##### **2.1.1. Pengertian Sistem**

Menurut Sutabri dalam Figran dkk (2015:1), pengertian dari sistem adalah “Sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu”.

Menurut Pratama (2014:7) mengemukakan bahwa, “Sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama”.

Dari kedua kutipan para ahli diatas, penulis dapat menyimpulkan bahwa sistem adalah sekumpulan kegiatan atau prosedur yang saling berhubungan dan bekerja sama untuk mencapai suatu tujuan tertentu dengan mengerjakan tugas pokok perusahaan secara harmonis.

### 2.1.2. Karakteristik Sistem

Menurut Hutahaean (2015:3) supaya sistem itu dikatakan sistem yang baik memiliki karakteristik yaitu:

#### 1. Komponen

Suatu sistem terdiri dari sejumlah komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem terdiri dari komponen yang berupa subsistem atau bagian-bagian dari sistem.

#### 2. Batasan Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

#### 3. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar sistem (*Environment*) adalah diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan yang harus tetap dijaga dan yang merugikan yang harus dijaga dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup sistem.

#### 4. Penghubung Sistem (*Interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem lainnya. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung.

5. Masukkan Sistem (*Input*)

Masukkan adalah energi yang dimasukkan ke dalam sistem, yang dapat berupa perawatan (*maintenance input*), dan masukkan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Contoh dalam sistem *computer program* adalah *maintenance input* sedangkan data adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Keluaran sistem adalah energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Contoh komputer menghasilkan panas yang merupakan sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

7. Pengolah Sistem

Suatu sistem menjadi bagian pengolah yang akan merubah masukkan menjadi keluaran. Sistem produksi akan mengolah bahan baku menjadi bahan jadi, sistem akuntansi akan mengolah data menjadi laporan-laporan keuangan.

8. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan input yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

### 2.1.3. Klasifikasi Sistem

Menurut Sutabri dalam Damayanti dan Riyanto (2013:19), klasifikasi sistem adalah suatu bentuk integrasi antara suatu komponen dengan komponen lainnya, karena sistem memiliki sarana yang berbeda untuk setiap kasus yang terjadi dalam sistem tersebut. Oleh karena itu sistem dapat diklasifikasikan sebagai berikut:

1. Sistem Abstrak

Sistem yang berupa pemikiran-pemikiran atau ide-ide yang tidak tampak secara fisik misalnya Sistem Teologia (Ketuhanan).

2. Sistem Fisik

Sistem yang ada secara fisik, seperti komputer, sistem produksi, dan sistem akuntansi.

3. Sistem Alamiyah

Sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi, terjadinya siang dan malam, dan pergantian musim.

4. Sistem Buatan Manusia

Sistem yang melibatkan manusia dengan mesin, yang disebut dengan *Human Machine System*. Contohnya sistem komputer.

5. Sistem Deterministik

Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem deterministik. Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan.

6. Sistem Probabilistik

Sistem yang kondisi masa depannya tidak dapat diprediksi, karena mengandung unsur probabilitas.

7. Sistem Terbuka

Sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya, yang menerima masukan dan menghasilkan keluaran untuk subsistem lainnya.

8. Sistem Tertutup

Sistem yang tidak berhubungan dan dipengaruhi oleh lingkungan luarnya.

Sistem ini bekerja secara otomatis tanpa ada campur tangan dari pihak luar.

#### 2.1.4. Pengertian Informasi

Menurut Pratama (2014:9) mengemukakan bahwa, “Informasi merupakan hasil pengolahan data dari satu atau berbagai sumber, yang kemudian diolah, sehingga memberikan nilai, arti dan manfaat”.

Menurut Darmawan dan Fauzi (2013:26), “Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat yang akan datang”.

Dari kedua kutipan diatas, penulis dapat menyimpulkan bahwa informasi adalah hasil pengolahan data yang hasilnya akan bermanfaat dan berguna bagi orang yang menerima informasi tersebut.

#### 2.1.5. Pengertian Sistem Informasi

Menurut Hutahaean (2015:13) mengatakan bahwa, Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan

kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan.

Menurut Pamungkas (2017:1), Sistem Informasi merupakan suatu sistem dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur, dan pengendalian untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan.

Dari kedua kutipan di atas, penulis dapat menyimpulkan bahwa sistem informasi adalah suatu sistem yang berada didalam suatu organisasi atau kelompok guna memberi jalur untuk berkomunikasi agar menghasilkan sebuah informasi yang dibutuhkan dan bermanfaat untuk pengambilan keputusan.

#### **2.1.6. Pengertian Pemrograman Berorientasi Objek**

Menurut Sukamto dan M. Shalahuddin (2018:100) mengemukakan bahwa, Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis.

#### **2.1.7. Bahasa Pemrograman Java**

Dalam buku *Rekayasa Perangkat Lunak* yang ditulis oleh Sukamto dan M. Shalahuddin (2018:103), Java dikembangkan oleh perusahaan *Sun Microsystem*. Java menurut definisi dari *Sun Microsystem* adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. Java 2 adalah generasi kedua dari *Java Platform*.

Java berdiri diatas sebuah mesin *interpreter* yang diberi nama *Java Virtual Machine* (JVM). JVM inilah yang akan membaca *bytecode* dalam *file .class* dari

suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu bahasa Java disebut sebagai bahasa pemrograman yang *portable* karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM.

Java merupakan bahasa pemrograman objek murni karena semua kode programnya dibungkus dalam kelas. Saat ini *Sun Microsystem* sudah diakuisisi *Oracle Corporation* sehingga pengembangan Java diteruskan oleh *Oracle Corporation*.

#### 2.1.8. *Jaspersoft Ireport*

Menurut Sugianur dan Nurcahyanti (2015:4), *iReport* adalah sebuah *tool* yang digunakan untuk membuat *design* laporan pada *Jaspersoft*. Artinya dengan fasilitas ini kita bisa membuat *report* dengan *drag* dan *drop*, bahkan secara otomatis (*wizard*). Ketentuan atau objek-objek laporan terbagi kedalam kelompok-kelompok area masing-masing. Hal ini agar *fleksibel* dan mudah dalam implementasi untuk berbagai kebutuhan yang seragam. Objek-objek *report* adalah sebagai berikut:

- a. *Band Title* adalah untuk meletakkan judul atau keterangan lain yang terkait judul.
- b. *Band PageHeader* untuk keterangan yang muncul disetiap halaman.
- c. *Band ColumnHeader* adalah untuk memberi judul pada *detail* laporan.
- d. *Band GroupHeader* adalah untuk meletakkan objek-objek. Bisa berupa judul atau data-data yang bersifat kelompok data yakni sebagai grup.

- e. *Band Detail* adalah untuk meletakkan objek yang merupakan representasi dari isi *detail* data dalam judul-judul kolom, baik pada *ColumnHeader* maupun *GroupHeader*.

*Jaspereport* file yang dibutuhkan adalah *jrxml* yaitu sebuah format *xml* yang dikenalnya, dengan aturan penulisan yang sudah ditetapkan mulai dari definisi header, kolom, isi detil laporan, grup dan sebagainya. *Tool ireport* yang secara otomatis menghasilkan *file – file jrxml* melalui *design* secara langsung *drag and drop*, dan bisa melakukan *preview*, sehingga pembuatan laporan menjadi sangat mudah.

### 2.1.9. Basis Data

Menurut Pamungkas (2017:2), “Basis Data merupakan suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, dan dengan *software* untuk melakukan manipulasi untuk kegunaan tertentu”.

#### 1. XAMPP

Menurut Yudhanto dan Purbaya dalam Supriyanta dan Nisa (2015:36), “XAMPP merupakan program paket PHP dan MySQL berbasis *opensource* yang saat ini merupakan andalan para programmer PHP dalam melakukan programming dan melakukan testing hasil programnya”.

#### 2. Apache

Menurut Sadeli dalam Supriyanta dan Nisa (2015:37), *Apache* (*Server HTTP Apache* atau *Server Web/WWW Apache* ) adalah *webserver* yang dapat dijalankan di banyak sistem operasi (*Unix, BSD, Linux, Microsoft Windows dan Novell Netware* serta *platform* lainnya) yang berguna untuk melayani dan memfungsikan situs *web*.

### 3. *MySQL*

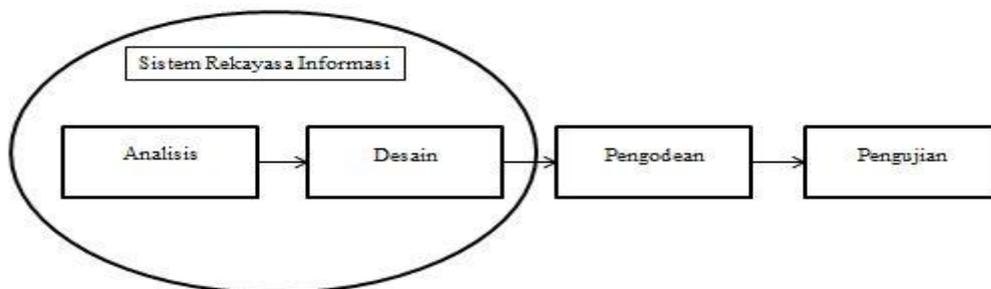
Menurut Sibero dalam Supriyanta dan Nisa (2015:37), *MySQL* atau dibaca “*My Sekuel*” dengan suatu RDBMS (*Relational Database Management System*) merupakan aplikasi sistem yang menjalankan fungsi pengolahan data. *MySQL* sendiri pertama dikembangkan oleh *MySQL AB* yang kemudian diakuisisi oleh *Sun Microsystem* dan terakhir dikelola oleh *Oracle Corporation*.

### 4. *PhpMyAdmin*

Menurut Rahman dalam Supriyanta dan Nisa (2015:37), *PHPMyadmin* adalah sebuah *software* berbasis pemrograman PHP yang dipergunakan sebagai administrator MySQL melalui *browser (web)* yang digunakan untuk *management database*.

#### 2.1.10. Model Perangkat Lunak

Menurut Sukamto dan M. Shalahuddin (2018:28), Model SDLC 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 perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).



Sumber : Sukamto dan M. Shalahuddin (2018:29)

**Gambar II.1. Ilustrasi Model *Waterfall***

1. Analisa Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminilisir kesalahan (*Error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

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

### 2.1.11. Pengertian Data

Menurut Pamungkas (2017:1), “Data merupakan nilai yang merepresentasikan deskripsi dari suatu objek atau kejadian”.

Menurut Darmawan dan Fauzi (2013:1), “Data adalah fakta atau apa pun yang dapat digunakan sebagai input dalam menghasilkan informasi”.

Dari kedua kutipan diatas, penulis dapat menyimpulkan bahwa data adalah sesuatu yang akan digunakan sebagai masukan untuk menghasilkan keluaran yang dinilai sebagai informasi.

### 2.1.12. Pengertian Penduduk

Menurut keputusan Menteri Dalam Negeri No 54 tentang kependudukan dalam Fujiyati (2015:2), Penduduk adalah Warga Negara Indonesia (WNI) dan Warga Negara Asing (WNA) pemegang ijin tinggal tetap di wilayah Negara kesatuan republik Indonesia atau semua orang yang berdomisili di desatersebut selama enam bulan lebih atau mereka yang berdomisili kurang dari enam bulan tetapi bertujuan menetap.

Menurut Kusumawati dkk (2017:28), “Penduduk adalah warga negara Indonesia dan orang asing yang bertempat tinggal di Indonesia”.

Dari kedua definisi diatas, penulis dapat menyimpulkan bahwa penduduk adalah seluruh masyarakat baik yang berstatus warga negara indonesia dan warga negara asing yang memiliki ijin untuk tinggal dan menetap di wilayah Negara Kesatuan Republik Indonesia.

### **2.1.13. Pengertian Administrasi Kependudukan**

Menurut Undang-Undang Dasar nomor 23 tahun 2006 tentang Administrasi Kependudukan dalam Lestari (2016:2622) menyatakan bahwa, Berdasarkan Undang-undang Republik Indonesia Nomor 23 tahun 2006 tentang Administrasi Kependudukan, Administrasi Kependudukan adalah rangkaian kegiatan penataan dan penertiban dalam penerbitan dokumen dan data kependudukan melalui pendaftaran penduduk, pencatatan sipil, pengelolaan informasi administrasi kependudukan serta pendayagunaan hasilnya untuk pelayanan publik dan pembangunan sektor lain.

### **2.1.14. Sistem Informasi Administrasi Kependudukan**

Menurut Undang-Undang Nomor 23 Tahun 2006 tentang administrasi kependudukan dalam Marbun dkk (2015:5) menyebutkan bahwa, Sistem Informasi Administrasi Kependudukan, selanjutnya disingkat SIAK, adalah sistem informasi yang memanfaatkan teknologi informasi dan komunikasi untuk memfasilitasi pengelolaan informasi administrasi kependudukan di tingkat Penyelenggara dan Instansi Pelaksana sebagai satu kesatuan.

## **2.2 Teori Pendukung**

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

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

Menurut Pratama (2014:49), “ERD adalah diagram yang menggambarkan keterkaitan antar tabel beserta dengan *field-field* di dalamnya pada suatu database”.

Menurut Yanto (2016:32), ERD adalah suatu diagram untuk menggambarkan desain konseptual dari model konseptual suatu basis data relasional. ERD juga merupakan gambaran yang merelasikan antara objek yang satu dengan objek yang lain dari objek di dunia nyata yang sering dikenal dengan hubungan antar entitas.

## 2. Komponen *Entity Relationship Diagram* (ERD)

Menurut Yanto (2016:32), ERD terdiri dari 3 komponen utama yaitu Entitas (*Entity*), Atribut (*Attribute*) dan Relasi (*Relationship*).

### a. Entitas (*Entity*)

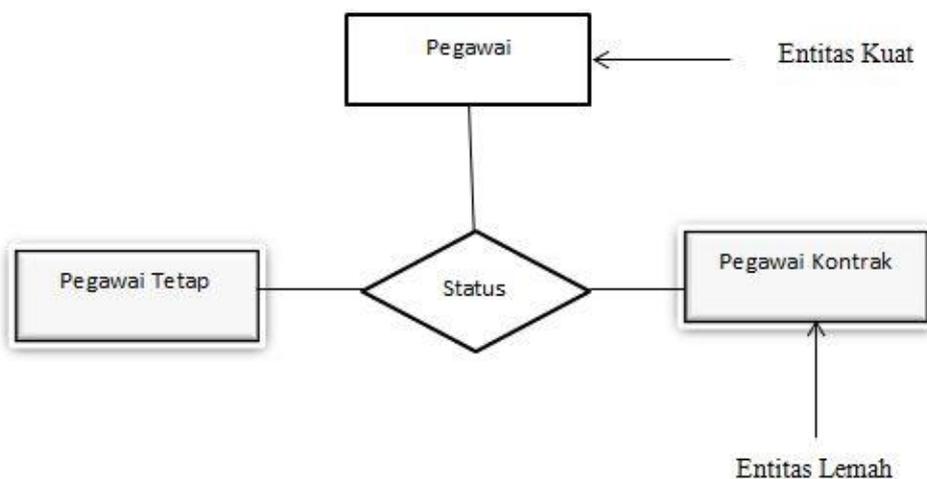
Entitas adalah suatu objek di dunia nyata yang dapat dibedakan dengan objek lainnya. Objek tersebut dapat berupa orang, benda ataupun hal lainnya. Entitas terbagi menjadi dua yaitu:

#### 1) Entitas Kuat (*Strong Entity*)

Entitas kuat adalah entitas yang dapat berdiri sendiri tidak bergantung pada entitas lainnya, entitas kuat memiliki atribut *key* dan entitas kuat digambarkan sebagai kotak persegi panjang bergaris tunggal. Contoh entitas kuat adalah entitas pegawai.

#### 2) Entitas Lemah (*Weak Entity*)

Entitas lemah adalah entitas yang tidak dapat berdiri sendiri. Entitas lemah merupakan hasil dari pembentukan entitas kuat, entitas lemah tidak memiliki atribut *key* dan entitas lemah digambarkan sebagai kotak persegi panjang bergaris ganda. Jika entitas kuat yang membentuk entitas lemah dihapus maka secara otomatis entitas lemah akan terhapus. Contoh entitas lemah adalah entitas pegawai kontrak, pegawai tetap.



Sumber : Yanto (2016:34)

**Gambar II.2. Jenis Entitas**

b. Atribut (*attribute*)

Atribut merupakan semua informasi yang berkaitan dengan entitas. Atribut sering dikenal dengan *property* dari suatu entitas atau objek. Atribut digambarkan dalam bentuk lingkaran *clips*. Macam-macam atribut:

1) Atribut Sederhana (*Simple Attribute*)

Atribut sederhana adalah atribut yang dinilainya tidak dapat dibagi lagi menjadi banyak atribut yang lebih kecil.

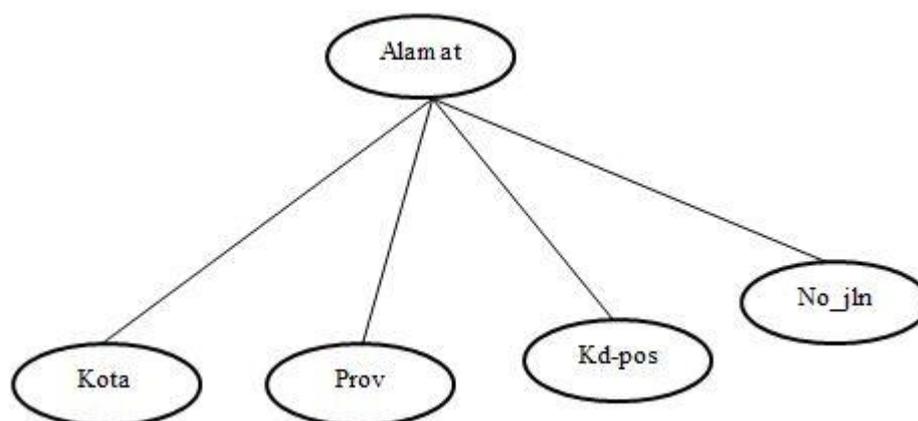


Sumber : Yanto (2016:34)

**Gambar II.3. Atribut Sederhana**

## 2) Atribut Komposit (*Composite Attribute*)

Atribut komposit adalah atribut gabungan yang nilainya dapat dipecah menjadi bagian yang lebih kecil. Atau sering disebut atribut yang terdiri dari beberapa atribut kecil di dalamnya.



Sumber : Yanto (2016:35)

**Gambar II.4. Atribut Komposit**

## 3) Atribut Bernilai Tunggal (*Single Values Attribute*)

Atribut bernilai tunggal adalah jenis atribut yang nilainya hanya satu dari suatu entitas. Contoh atribut bernilai tunggal adalah tanggal\_lahir dari entitas mahasiswa. Telah bisa dipastikan bahwa setiap mahasiswa mempunyai satu tanggal\_lahir.

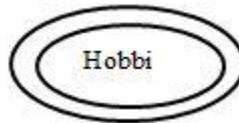


Sumber : Yanto (2016:35)

**Gambar II.5. Atribut Bernilai Tunggal**

#### 4) Atribut Bernilai Banyak

Atribut bernilai banyak adalah jenis atribut yang nilainya lebih dari satu dalam suatu entitas tertentu. Contoh atribut bernilai banyak adalah hobi dimungkinkan bahwa mahasiswa memiliki lebih dari satu hobi.

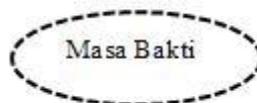


Sumber : Yanto (2016:36)

**Gambar II.6. Atribut Bernilai Banyak**

#### 5) Atribut Turunan (*Derived Attribute*)

Atribut turunan adalah jenis atribut yang nilainya diperoleh dari atribut yang lain. Contoh atribut turunan adalah masa\_bakti dari entitas pegawai. Atribut masa\_bakti akan muncul nilainya ketika atribut tanggal\_masuk\_kerja sudah ada nilainya. Pada dasarnya atribut masa bakti tidak akan dijadikan suatu kolom. Atribut masa\_bakti akan muncul dengan bantuan query.



Sumber : Yanto (2016:36)

**Gambar II.7. Atribut Turunan**

#### 6) Atribut Identitas (*Key Attribute*)

Atribut identitas adalah atribut yang dijadikan sebagai kunci pada suatu table. Sifat atribut identitas ini unik, tidak ada yang menyamai, atribut identitas terdiri dari beberapa jenis yaitu:

a) *Super Key*

*Super key* adalah satu atribut atau kumpulan atribut yang secara unik mengidentifikasi sebuah baris di dalam relasi atau himpunan dari satu atau lebih entitas yang dapat digunakan untuk mengidentifikasi secara unik sebuah entitas dalam set entitas.

b) *Candidate Key*

*Candidate key* adalah atribut yang menjadi determinan yang dapat dijadikan identitas baru pada sebuah relasi biasanya *super key* minimum.

c) *Primary Key*

*Primary key* adalah kandidat *key* yang dipilih untuk mengidentifikasi baris data secara unik dalam relasi.

d) *Alternative Key*

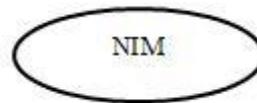
*Alternative key* adalah kandidat *key* yang tidak terpilih sebagai *primary key* atau atribut untuk menggantikan kunci utama.

e) *Foreign Key*

*Foreign key* adalah atribut dengan domain yang sama yang menjadi kunci utama sebuah relasi, tetapi pada relasi lain atribut adalah sebagai atribut biasa.

f) *Composite Key*

*Composite key* adalah kunci yang terdiri dari dua atribut atau lebih. Atribut-atribut tersebut jika berdiri sendiri tidak menjadi identitas baris, tetapi bila dirangkaikan menjadi satu kesatuan akan dapat mengidentifikasi secara unik.

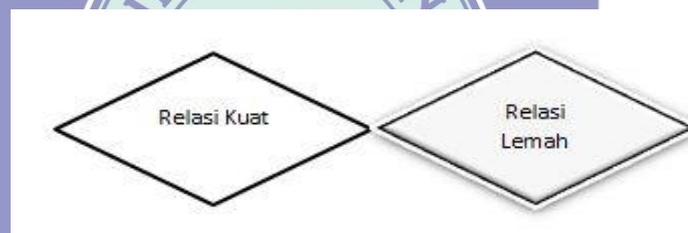


Sumber : Yanto (2016:38)

**Gambar II.8. Atribut Identitas**

c. Tipe Relasi

Gambar belah ketupat merupakan perlambangan relasi antar entitas atau sering disebut kerelasian. Ada 2 macam penggambaran relasi yaitu relasi kuat dan relasi lemah. Relasi kuat adalah untuk menghubungkan antar entitas kuat sedangkan relasi lemah untuk menghubungkan antar entitas kuat dengan entitas lemah.

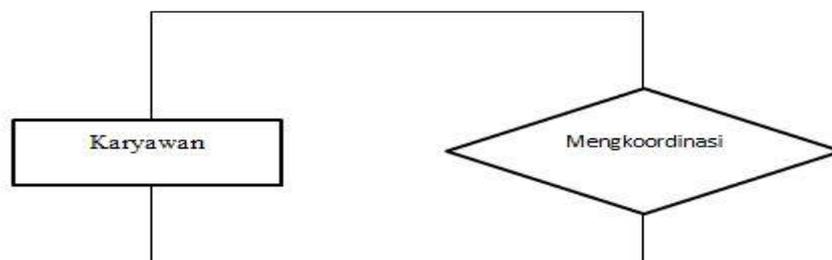


Sumber : Yanto (2016:38)

**Gambar II.9. Tipe Relasi**

Ada tiga macam relasi menurut derajatnya, yaitu:

- 1) *Unary* adalah relasi yang menghubungkan entitas yang sejenis



Sumber : Yanto (2016:39)

**Gambar II.10. Derajat Relasi Unary**

Derajat hubungan unary entitas dosen hanya bekerja sama dengan entitas dosen yang entitasnya sejenis, begitu juga dengan entitas karyawan (pimpinan) mengoordinasi entitas karyawan (pekerja) yang entitasnya sejenis.

2) *Binary* adalah relasi yang menghubungkan entitas yang tidak sejenis

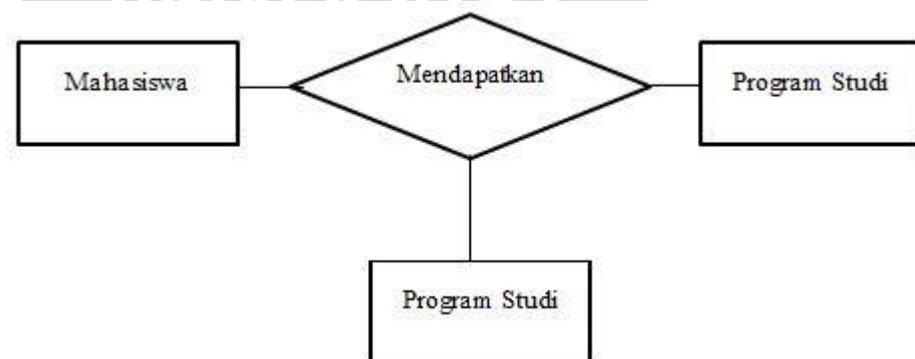


Sumber : Yanto (2016:39)

**Gambar II.11. Derajat Relasi Binary**

Derajat relasi binary pada gambar II.12 adalah entitas kepala program studi berelasi dengan entitas program studi.

3) *Ternary* adalah relasi yang menghubungkan lebih dari dua entitas yang tidak sejenis.



Sumber : Yanto (2016:40)

**Gambar II.12. Derajat Relasi Ternary**

Derajat relasi ternary pada gambar II.13 adalah entitas mahasiswa berelasi dengan dua entitas yang berbeda yaitu matakuliah nilai.

#### d. Derajat Kardinalitas

Derajat kardinalitas merupakan penjabaran dari hubungan antar entitas.

Derajat kardinalitas dibagi menjadi 3 bagian yaitu:

##### 1) Derajat Kardinalitas *One to One*

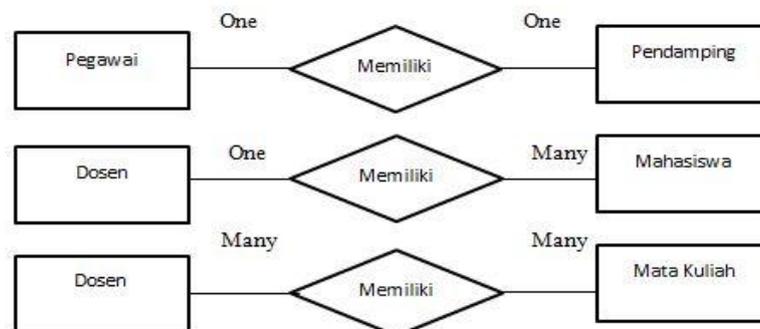
Derajat kardinalitas *one to one* terjadi jika satu entitas X hanya berelasi dengan satu entitas Y, ataupun sebaliknya. Sebagai contoh satu pegawai studi hanya memiliki satu pendamping.

##### 2) Derajat Kardinalitas *One to Many*

Derajat kardinalitas *one to many* terjadi jika satu entitas X berelasi dengan banyak entitas Y, ataupun sebaliknya. Sebagai contoh satu dosen mengampu banyak mahasiswa.

##### 3) Derajat Kardinalitas *Many to Many*

Derajat kardinalitas *many to many* terjadi jika banyak entitas X berelasi dengan banyak entitas Y, atau pun sebaliknya. Sebagai contoh banyak mahasiswa belajar banyak mata kuliah.



Sumber : Yanto (2016:41)

**Gambar II.13. Contoh Derajat Kardinalitas**

### 3. *Logical Record Structure (LRS)*

Menurut Tabrani (2014:35) menyimpulkan bahwa:

*Logical Record Structure* dibentuk dengan nomor dari tipe *record*. Beberapa tipe *record* digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik. Perbedaan LRS dengan E-R diagram adalah nama tipe *record* berada diluar kotak *field* tipe *record* ditempatkan. *Logical Record Structure* terdiri dari *link-link* diantara tipe *record*. Link ini menunjukkan arah dari satu tipe *record* lainnya. Banyak *link* dari LRS yang diberi tanda *field-field* yang kelihatan pada kedua *link* tipe *record*. Penggambaran LRS mulai dengan menggunakan model yang dimengerti. Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonverensikan ke LRS, metode yang lain dimulai dengan ER-diagram dan langsung dikonversikan ke LRS.

Menurut Fridayanthie dan Mahdiati (2016:132), “*Logical Record Structured (LRS)* adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil relasi antar himpunan entitas. Menentukan kardinalitas, jumlah tabel, dan *ForeignKey (FK)*”.

Berikut adalah cara membentuk skema *database* atau LRS (*Logical Record Structured*) berdasarkan *Entity Relationship Diagram*:

- a. Jika relasinya satu-ke-satu, maka *foreign key* diletakan pada salah satu dari dua entitas yang ada tau menyatukan kedua entitas tersebut.
- b. Jika relasinya satu-ke-banyak, maka *foreign key* diletakan pada entitas *Many*.
- c. Jika relasinya banyak-ke-banyak, maka dibua “*file konektor*” yang berisi dua *foreign key* yang berasal dari kedua entitas.

#### 2.2.2. *Unified Modelling Language (UML)*

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

Menurut Fadallah dan Rosyida (2018:62) , “UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung” [5]. “*Unified Modeling Language (UML)* adalah bahasa untuk menspesifikasi, memvisualisasikan, serta mengontruksi bangunan dasar sistem perangkat lunak, termasuk melibatkan pemodelan aturan-aturan bisnis” [2]. UML menyediakan beberapa diagram visual yang menunjukkan berbagai aspek dalam sistem, ada beberapa diagram yang disediakan dalam UML, antara lain:

1. *Use case diagram* menyajikan interaksi antara *use case* dan aktor. Dimana, aktor dapat berupa orang, peralatan atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use case* menggambarkan fungsional sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai.
2. *Activity diagram* menggambarkan aliran fungsional sistem. Pada tahap pemodelan bisnis, diagram aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis. Dapat juga digunakan untuk menggambarkan aliran kejadian dalam *use case*.
3. *Sequence diagram* digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Setiap objek yang terlibat dalam diagram *use case* digambarkan dengan garis putus-putus *vertikal*, kemudian *message* yang dikirim oleh objek digambarkan dengan garis *horizontal* secara kronologis dari atas ke bawah.
4. *Class diagram* menunjukkan interaksi antar kelas dalam sistem. Diagram kelas dibangun berdasarkan diagram *use case* dan diagram *sekuensial* yang

telah dibuat sebelumnya. Kelas memiliki tiga area pokok yaitu : Nama (*stereotype*), *Atribut* dan *Metode*. *Atribut* dan *Metode* dapat memiliki salah satu sifat berikut:

- a. *Private*, tidak dapat dipanggil dari luar kelas yang bersangkutan.
- b. *Protected*, hanya dapat dipanggil oleh kelas yang bersangkutan dan anak-anak yang mewarisinya.
- c. *Public*, dapat dipanggil oleh siapa saja.

