

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Sistem memiliki fungsi untuk menangani fungsi yang secara terus-menerus dan berulang-ulang atau yang rutin terjadi. Umumnya sistem dibuat sebagai suatu alat untuk mempermudah dalam pekerjaan manusia.

Konsep dasar sistem merupakan kumpulan dari elemen, aspek dan sudut pandang yang berbeda-beda sesuai dengan keterangan dalam hal-hal yang berkaitan dengan sistem yang memiliki ciri dan karakteristik tertentu.

2.1.1. Pengertian Sistem

Pendekatan yang lebih menekankan pada elemen atau komponennya, mendefinisikan suatu sistem sebagai kumpulan dari elemen-elemen yang berinteraksi guna mencapai tujuan tertentu. Secara umum suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen yang terorganisir, saling berinteraksi, saling ketergantungan satu sama lain dan terpadu untuk mencapai tujuan tertentu.

Seperti dikatakan oleh Davis dalam Ladjamudin (2013:3), menyatakan “sistem sebagai bagian-bagian yang saling berkaitan yang beroperasi bersama-sama untuk mencapai beberapa saran dan maksud”. Menurut Fathansyah (2015:11), mengemukakan “Sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional (dengan satuan fungsi dan tugas khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses tertentu”. Sistem merupakan satu bentuk komponen

dengan komponen yang lainnya. Karena sistem memiliki sasaran yang berbeda untuk setiap kasus terjadi yang ada didalam sistem tersebut (Ladjamudin, 2013:6).

Berdasarkan pengertian di atas dapat di simpulkan bahwa sistem merupakan suatu jaringan kerja dari kumpulan dari unsur-unsur, komponen-komponen, prosedur-prosedur atau variabel-variabel yang terorganisir, dan saling berinteraksi atau saling ketergantungan antara satu dan lainnya dimana setiap sistem dibuat agar bersama-sama dapat menangani dan menyelesaikan sesuatu masalah tertentu yang secara rutin terjadi dan terus berulang.

Unsur, komponen atau variabel yang dimaksud bukan hanya bagian yang tampak secara fisik tetapi juga hal-hal yang bersifat abstrak atau konseptual seperti misi, pekerjaan yang kegiatan kelompok yang informal dan sebagainya.

A. Karakteristik Sistem

Di dalam suatu sistem, terdapat beberapa karakteristik yang harus di pahami terlebih dahulu sebagai dasar pertimbangan dalam pembuatan sistem tersebut.

Menurut Mulyanto (2009:2), mengemukakan bahwa sebuah “sistem mempunyai karakteristik atau sifat-sifat tertentu yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem”. Adapun karakteristik menurut Mulyanto (2009:2), yang dimaksud adalah sebagai berikut :

1. Komponen Sistem (*Component*)

Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung (*Interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian, dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

6. Keluaran Sistem (*Output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini dapat menjadi masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi.

Informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang menjadi input bagi subsistem lain.

7. Pengolah Sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran (*Objectives*)

Suatu sistem mempunyai tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

9. Tujuan (*Goal*)

Suatu sistem dikatakan berhasil apabila mengenai sasaran atau tujuannya. Jika suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

B. Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, adapun klasifikasi terhadap sistem tersebut menurut Mulyanto (2009:8), adalah sebagai berikut :

1. Sistem Abstrak dan Sistem Fisik

- a. Sistem abstrak (*abstract system*) merupakan sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik (sistem teologia yang merupakan suatu sistem yang menggambarkan hubungan antara Tuhan dengan manusia).
- b. Sistem Fisik (*Physical System*) diartikan sebagai sistem yang nampak

secara fisik sehingga setiap makhluk dapat melihatnya, misalnya sistem komputer, sistem akuntansi, sistem produksi, dan lain-lain.

2. Sistem Alamiah dan Sistem Buatan Manusia

- a. Sistem Alamiah (*Natural System*) merupakan sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem tata surya, sistem galaksi, sistem reproduksi dan lain-lain.
- b. Sistem Buatan Manusia (*Human Made System*) merupakan sistem yang dirancang oleh manusia. Sistem buatan yang melibatkan interaksi manusia, misalnya sistem akuntansi, sistem informasi, dan lain-lain.

3. Sistem Deterministik dan Sistem Probabilistik

- a. Sistem deterministik (*deterministic system*) merupakan sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan, misalnya sistem computer, adalah contoh sistem yang tingkah lakunya dapat dipastikan program-program computer yang dijalankan.
- b. Sistem Probabilistik (*Probabilistic System*) merupakan sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas, misalnya sistem manusia.

4. Sistem Terbuka dan Sistem Tertutup

- a. Sistem Tertutup (*Open System*) merupakan sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih spesifik dikenal juga dengan sistem terotomasi, yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh satu atau lebih

komputer sebagai bagian dari sistem yang dipergunakan dalam masyarakat modern. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya, misalnya sistem kebudayaan manusia.

- b. Sistem Tertutup (*Closed System*) merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya campur tangan dari pihak luar. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relative tertutup, tidak benar-benar tertutup).

2.1.2. Bahasa Pemrograman

Sebuah intruksi standar untuk memerintah komputer agar menjalankan fungsi tertentu. Bahasa pemrograman ini merupakan suatu himpunan dari aturan *sintaks* dan semantik yang dipakai untuk mendefinisikan program Komputer.

A. Microsoft Visual Basic 6.0

Menurut Kusriani dan Andri, (2007:127), menyatakan bahwa “Visual Basic adalah salah satu bahasa pemrograman komputer”. *Microsoft Visual Basic 6.0* menyediakan fasilitas yang memungkinkan kita untuk menyusun sebuah program dengan memasang objek-objek grafis dalam sebuah *form*. Selain itu Visual Basic juga menawarkan berbagai kemudahan dalam mengelola sebuah *database*, kemudahan itu ditambah lagi dengan tersedianya sarana dan peranti yang lengkap . maka dari itu *Visual Basic* cukup populer dan mudah untuk dipelajari (Madcorms, 2010:2)

Berdasarkan definisi diatas, dapat disimpulkan bahwa *Microsoft Visual Basic 6.0* adalah bahasa pemrograman yang mudah digunakan sebagai sarana atau alat untuk membuat program aplikasi berbasis *desktop*.

B. Crystal Report

Menurut Mesran (2010:173), menyatakan bahwa “*Crystal Report* merupakan sebuah produk dari Seagate yang didesain khusus untuk membuat suatu laporan”. Membuat Laporan dengan menggunakan *Crystal Report* hasilnya lebih baik dan lebih mudah, karena banyak tersedia objek-objek maupun komponen yang mudah digunakan. *Crystal Report* merupakan program yang terpisah dengan *Microsoft Visual Basic 6.0*, tetapi keduanya dapat dihubungkan (Madcoms, 2010:234).

Dari definisi para ahli diatas dapat disimpulkan bahwa *Crystal Report* adalah program yang mempermudah kita untuk membuat laporan pada program *Visual Basic*.

2.1.3. Basis Data

Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data adalah representasi dari fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep, keadaan, dan sebagainya yang diwujudkan dalam bentuk angka, huruf, symbol, teks, gambar, bunyi, atau kombinasinya (Fathansyah, 2015:2). Rosa dan Shalahuddin (2015:43) mengemukakan bahwa “sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”.

Berdasarkan kesimpulan dari para ahli, maka basis data merupakan

kumpulan informasi atau data yang bertujuan untuk mengelola data yang sudah diolah atau informasi dan dapat digunakan kembali.

Aplikasi basis data yang penulis paparkan dalam Tugas Akhir ini yaitu:

1. PhpMyAdmin

PhpMyAdmin merupakan salah satu perangkat lunak yang digunakan untuk melakukan pengolahan terhadap basis data, tabel-tabel (*entity*), atribut (*field*) dari *entity*, hubungan (*relation*) antar *entity*, indeks, dan lain-lain.

Menurut siberio (2013:376), menyatakan pendapatnya bahwa PhpMyAdmin digunakan untuk administrasi *database*, mengubah *database*, pembuatan tabel, mengubah tabel, menghapus tabel, menambah data, menampilkan data, mengubah data, menghapus data membuat *view*, menghapus *view*, membuat *index* kolom dan menghapus *index* kolom.

Menurut Riyanto (2015:17), “phpMyAdmin merupakan aplikasi *web* berbasis PHP yang telah banyak digunakan untuk administrasi *database* MySQL”.

Dapat ditarik kesimpulan bahwa, phpMyAdmin merupakan sebuah aplikasi pengelola basis data yang digunakan sebagai *administrator* dalam MySQL melalui *browser* yang digunakan untuk *manage database*.

2. MySQL

Selain PhpMyAdmin, MySQL juga merupakan perangkat lunak yang berfungsi sebagai manajemen basis data dan dapat melakukan rekayasa struktur data sesuai dengan SQL (*Structured Query Language*).

Menurut Anhar (2010: 21) “MySQL (*My Structure Language*) adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database management*

system) atau DBMS, seperti Oracle, MS SQL, Postagre SQL dan lain-lain” .
Sedangkan menurut Winarno dkk (2014:102), menyatakan “MySQL merupakan tipe data relasional yang artinya MySQL menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan”.

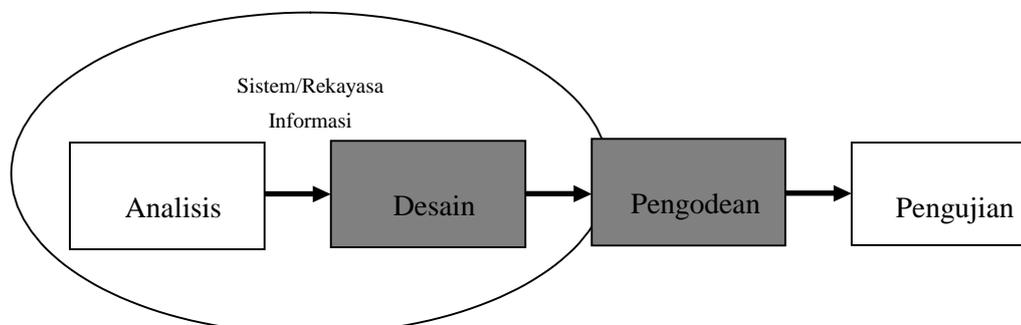
Berdasarkan pengertian para ahli, maka *MySQL* merupakan *software* manajemen basis data yang digunakan untuk mengelola *database*.

2.1.4. Model Pengembangan Perangkat Lunak

Salah satu dari tahap perancangan sistem rinci/detail dari siklus hidup pengembangan sistem. Secara garis besar, perangkat lunak merupakan sekumpulan kode dan program untuk menjalankan eksekusi yang diperintahkan oleh user.

Dalam pengembangan perangkat lunak, disiplin ilmu yang sering digunakan sebagai dasar pemodelan pengembangan perangkat lunak menggunakan model air terjun (*waterfall*).

Menurut Rosa dan Shalahuddin (2015:28), mengemukakan bahwa, “model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain pengodean, pengujian dan tahap pendukung (*support*)”.



Sumber : Rosa dan Shalahuddin (2015:29)

Gambar II.1. Ilustrasi Model Waterfall

Adapun metode air terjun menurut Rosa dan Shalahuddin (2015:29) adalah sebagai berikut:

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara insentif 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 pengkodean. 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 ad-hoc dari segi *logic* dan fungsional serta memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir 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 bisa 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.

2.2. Teori Pendukung

Teori pendukung merupakan teori tambahan yang digunakan penulis dalam mendukung penulisan Tugas Akhir yang berlandaskan pendapat para ahli. Adapun teori pendukung yang digunakan yaitu:

2.2.1. Perancangan Sistem

Dalam tahapan perancangan ini, proses multi langkah-langkah yang fokus pada perancangan pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pemodelan.

Menurut Nafisah (2017:1), menyatakan “Perancangan adalah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi ”. Menurut Satzinger dkk (2017:1) “perancangan sistem adalah sekumpulan aktivitas yang menggambarkan secara rinci bagaimana sistem akan berjalan ”.

Berdasarkan pengertian di atas, penulis dapat menyimpulkan bahwa perancangan sistem merupakan suatu proses untuk membuat dan mendesain

sistem berjalan yang baru.

2.2.2. Sistem Informasi

Sistem yang baik tentunya akan menghasilkan suatu informasi dimana informasi tersebut akan digunakan sebagai bahan dasar pertimbangan suatu organisasi untuk menentukan langkah kedepannya.

Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan atau untuk mengendalikan sesuatu (Ladjamudin, 2013:13). Menurut Mulyanto (2009:29), “Sistem informasi adalah suatu komponen yang terdiri dari manusia, teknologi informasi, dan prosedur kerja yang memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk mencapai suatu tujuan”.

Dari pendapat para ahli di atas dapat disimpulkan bahwa sistem informasi merupakan gabungan dari manusia, *hardware*, *software*, jaringan komunikasi dan data yang saling berinteraksi untuk menyimpan, mengumpulkan, memproses, dan mendistribusikan informasi untuk mendukung pengambilan keputusan dalam suatu organisasi.

2.2.3. Pengarsipan Data Kependudukan

Penulis melakukan analisa dan perancangan sistem terhadap pengarsipan data kependudukan pada Kelurahan Siantan Hulu Kota Pontianak.

A. Pengarsipan

Menurut Kamus Besar Bahasa Indonesia, “Arsip adalah dokumen tertulis, lisan, atau gambar dari waktu yang lampau, disimpan dalam media tulis(kertas), elektronik(kaset, disket komputer dan sebagainya), biasanya dikeluarkan oleh instansi resmi, disimpan dan dipelihara ditempat khusus untuk referensi”.

Berdasarkan pengertian diatas dapat disimpulkan pengarsipan adalah dokumen yang disimpan dalam media kertas, elektronik yang digunakan untuk penyimpanan data.

B. Data

Data merupakan bentuk jamak dari bentuk tunggal datum. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata (Sutabri,2012:1). Sedangkan menurut Ladjamudin (2013:20) “data merupakan komponen dasar dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi”.

Berdasarkan teori para ahli diatas dapat disimpulkan bahwa, data adalah deskripsi dasar dari benda, peristiwa, aktivitas dan transaksi yang direkam, dikelompokkan, dan disimpan dalam jumlah yang besar untuk menghasilkan informasi.

C. Kependudukan

Menurut Kamus Besar Bahasa Indonesia, “Penduduk adalah orang atau orang-orang yang mendiami suatu tempat atau orang-orang yang turun temurun tinggal disuatu daerah”. Sedangkan menurut Pasal 26, ayat 2 Undang-Undang Dasar 1945, “Penduduk ialah warga warga Negara Indonesia dan orang asing yang bertempat tinggal di Indonesia”.

Jadi dapat diartikan bahwa penduduk adalah orang-orang atau warga Negara Indonesia dan orang asing yang tinggal diIndonesia apakah itu menetap atau tidak menetap.

2.2.4. Diagram Alir Data (DAD)

Alat pembuatan model yang memungkinkan profesional sistem untuk

menggambarkan sistem sebagai suatu jaringan proses fungsional yang dibutuhkan satu sama lain dengan alur data baik secara manual ataupun terkomputerisasi.

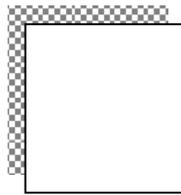
Menurut Ladjmudin (2013:63) "DAD menampilkan kegiatan sistem lengkap dengan komponen-komponen yang menunjukkan secara tegas file-file yang dipakai, unsur sumber atau tujuan data serta aliran data dari satu proses ke proses lainnya". Menurut Rosa dan Shalahuddin (2015:70), "DFD tidak sesuai untuk memodelkan sistem yang menggunakan pemrograman berorientasi objek".

Berdasarkan pendapat dari para ahli di atas, dapat ditarik kesimpulan bahwa *Data Flow Diagram* atau Diagram Arus Data merupakan suatu gambaran grafis dan suatu sistem yang menggunakan sejumlah bentuk-bentuk simbol untuk menggambarkan bagaimana data mengalir melalui suatu proses yang berkaitan.

Simbol atau lambang yang digunakan dalam membuat diagram alir data yang lazim digunakan, menurut Rosa dan Shalahuddin (2015:71), DAD terdiri dari empat buah simbol yaitu :

1. Entitas/lingkungan Luar (*External Entity*)

Simbol ini digunakan untuk menggambarkan asal atau tujuan data, menunjukkan entitas atau kesatuan yang berhubungan dengan sistem, dapat berupa orang, organisasi, atau sistem lainnya yang akan memberikan input atau menerima input dari sistem atau keduanya digunakan dengan simbol empat persegi panjang.

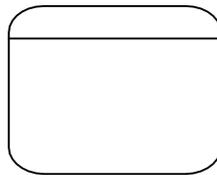


Sumber: Rosa dan Shalahuddin (2015:72)

Gambar II.2. External Entity

2. Proses (*Process*)

Simbol ini digunakan untuk proses pengolahan atau transformasi data, menunjukkan kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dan hasil suatu data yang masuk kedalam proses untuk menghasilkan arus data yang akan keluar dari proses, digambarkan dengan simbol lingkaran.

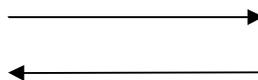


Sumber: Rosa dan Shalahuddin (2015:71)

Gambar II.3. Proses

3. Arus Data (*Data Flow*)

Simbol ini digunakan untuk menggambarkan aliran data yang berjalan, menunjukkan arus data yang berupa masukan untuk sistem atau hasil dari proses sistem yang mengalir diantara proses (*process*), simpanan data (*data store*) dan entitas (*external entity*) digambarkan dengan arah panah.

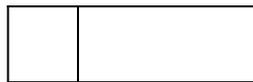


Sumber: Rosa dan Shalahuddin (2015:71)

Gambar II.4. Arus Data

4. Simpanan Data (*Data Store*)

Simbol ini digunakan untuk menggambarkan *data flow* yang sudah disimpan, menunjukkan suatu tempat penyimpanan data yang dapat berupa suatu file di sistem komputer, arsip atau catatan manual, tabel acuan dan lain-lain digambarkan dengan sepasang garis horizontal.



Sumber: Rosa dan Shalahuddin (2015:69)

Gambar II.5. *Data Store*

Tahap pembuatan Diagram Alir Data (DAD) menurut Kristanto (2008:30)

dibagi menjadi tiga tingkatan konstruksi Diagram Alir Data yaitu:

1. Diagram Konteks

Diagram ini dibuat untuk menggambarkan sumber serta tujuan data yang akan diproses atau dengan kata lain diagram tersebut untuk menggambarkan sistem secara global dari keseluruhan sistem yang ada.

2. Diagram Nol

Diagram dibuat untuk menggambarkan tahap-tahap proses yang akan ada didalam konteks atau penjabaran secara rinci.

3. Diagram Detail

Diagram ini dibuat untuk menggambarkan arus data secara lebih detail dan terperinci dari tahapan proses yang ada dalam diagram.

Berikut ini adalah aturan main Diagram Alir Data yang baku dan berlaku dalam penggunaan diagram alir data untuk membuat model sistem Rosa dan Shalahuddin (2015:72) yaitu :

1. Didalam diagram alir data tidak boleh menghubungkan antara *eksternal entity* dengan *eksternal entity* lain secara langsung.
2. Didalam diagram alir data tidak boleh menghubungkan *data store* dengan *data store* yang lain secara langsung.
3. Didalam diagram alir data tidak diperkenankan menghubungkan *data store* dengan *eksternal entity* secara langsung.
4. Setiap proses harus ada *data flow* yang masuk dan ada juga *data flow* yang keluar.

2.2.5. Kamus Data (*Data Dictionary*)

Kamus data berfungsi membantu pelaku atau pengguna sistem untuk mengartikan aplikasi secara detail dan mengorganisasi elemen-elemen data yang digunakan dalam sistem secara persis sehingga pemakai dan penganalisa sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses (Ladjamudin, 2013:70). Keterangan lebih lanjut tentang struktur dari suatu arus data di DAD secara lebih terperinci menurut Ladjamudin (2013:70) dapat dilihat dikamus data. Kamus data memuat hal-hal sebagai berikut:

a. Nama arus data

Nama arus data harus dicatat dikamus data, sehingga mereka yang membaca DAD memerlukan penjelasan lebih lanjut tentang suatu arus data tertentu dan dapat langsung mencarinya dengan lebih mudah.

b. Alias

Alias atau nama lain dari data dapat ditulis bila ada. Untuk menyatakan nama lain dari suatu data element atau *data store* yang sebenarnya sama dengan data *element data store* yang telah ada.

c. Bentuk data

Bentuk data perlu dicatat dikamus data, karena dapat dipergunakan untuk mengelompokkan kamus data kedalam kegunaannya sewaktu perancangan sistem.

d. Arus data

Arus data menunjukkan dari mana data mengalir dan kemana data menuju. Keterangan arus data ini perlu dicatat di kamus data untuk memudahkan mencari arus data di DAD.

e. Penjelasan

Untuk memperjelas tentang makna dari arus data yang dicatat dikamus data, maka sebagai penjelasan dapat diisi dengan keterangan-keterangan tentang arus data tersebut.

Berikut simbol yang menjelaskan informasi tambahan pada kamus data sebagai berikut:

Tabel II.1 Notasi Tipe Data

Notasi	Keterangan
X	Setiap Karakter
9	Angka <i>Numeric</i>
A	Karakter Alphabet
Z	Angka Nol Ditampilkan sebagai spasi kosong
.	Titik, sebagai pemisah ribuan
,	Koma, sebagai pemisah pecahan
-	Hypen, sebagai tanda penghubung
/	Slash, sebagai tanda pembagi

Sumber : Rosa dan Shalahuddin (2015:74)

1. Notasi Struktur Data

Notasi ini digunakan untuk membuat spesifikasi elemen data. Dimana notasi yang umum digunakan adalah sebagai berikut :

Tabel II.2 Notasi Struktur Data

Notasi	Keterangan
=	Terdiri
+	Dan(and)
()	Pilih (boleh Ya atau Tidak)
{}	Pengulangan proses (Iterasi)
[]	Pilih salah satu
I	Pemisah pilihan didalam tanda []
*	Keterangan atau catatan
@	Petunjuk (Key field)

Sumber : Rosa dan Shalahuddin (2015:74)

2.2.6. *Entity Relationship Diagram (ERD)*

Dalam perancangan basis data, ERD merupakan teknik pemodelan yang sering digunakan dalam menggambarkan basis data yang sedang dirancang.

1. Pengertian *Entity Relationship Diagram (ERD)*

ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak

Menurut Ladjamudin (2013:142), “*Entity Relationship Diagram (ERD)* adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak”. Menurut Rosa dan Shalahuddin (2015:53) “ERD adalah bentuk paling awal dalam melakukan perancangan basis data relasional. Jika menggunakan OODMBS maka perancangan ERD tidak perlu dilakukan”.

Dapat ditarik kesimpulan bawa ERD merupakan bentuk awal perancangan basis data yang menggunakan simbol-simbol yang terdiri dari entitas dan hubungannya.

2. Komponen *Entity Relationship Diagram* (ERD)

Simbol-simbol atau komponen yang digunakan pada *Entity Relationship Diagram* (ERD) adalah sebagai berikut:

a. Entitas (*entity*)

Menurut Rosa dan Shalahuddin (2015:53), “Entity menunjukkan obyek-obyek dasar yang terkait didalam sistem obyek dasar dapat berupa orang, benda atau hal lain yang keterangannya perlu disimpan dalam basis data”.



Sumber: Rosa dan Shalahuddin (2015:50)

Gambar II.6. Entitas

b. Atribut (*Attribute*)

Menurut Rosa dan Shalahuddin (2015:53), “Atribut sering juga disebut sebagai properti, merupakan keterangan-keterangan yang terkait pada sebuah *entitas* yang perlu disimpan sebagai basis data”.



Sumber: Rosa dan Shalahuddin (2015:50)

Gambar II.7. Atribut

c. Atribut kunci primer

Field atau kolom data yang butuh disimpan dalam suatu entitas dan

digunakan sebagai kunci akses *record* yang diinginkan, biasanya berupa id.

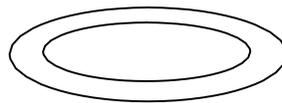


Sumber: Rosa dan Shalahuddin (2015:50)

Gambar II.8. Atribut Kunci Primer

d. Atribut multivalai (*multivalue*)

Field atau kolom data yang butuh disimpai dalam suatu entitas yang dapat memiliki nilai lebih dari satu.



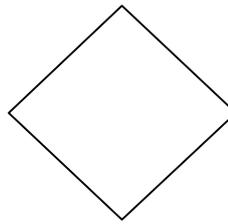
Sumber: Rosa dan Shalahuddin (2015:51)

Gambar II.9. Atribut Multinilai

a. Relasi (*relation*)

Menurut Rosa dan Shalahuddin (2015:56), “relasi atau hubungan adalah kejadian atau transaksi yang terjadi diantara dua *entity* yang keterangannya perlu disimpan dalam basis data”. Aturan penggambaran antar *entity* adalah:

- 1) Relasi dinyatakan dengan simbol belah ketupat.
- 2) Nama relasi dituliskan didalam simbol belah ketupat.
- 3) Relasi menghubungkan dua entity.
- 4) Nama relasi menggunakan kata kerja aktif (diawali awalan me) tunggal.
- 5) Nama relasi sedapat mungkin menggunakan nama yang mudah dipahami
- 6) dan dapat menyatakan makna dengan jelas.



Sumber: Rosa dan Shalahuddin (2015:51)

Gambar II.10. Relasi

b. Asosiasi/association

Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki *multiciplicity*.



Sumber: Rosa dan Shalahuddin (2015:51)

Gambar II.11. Asosiasi

2. Derajat *Relationship*

Menurut Ladjamudin (2013:144), mengemukakan bahwa “Derajat *Relationship* adalah jumlah entitas yang berpartisipasi dalam satu *Relationship*”.

Derajat *relationship* yang sering dipakai didalam ERD Ladjamudin (2013:145), adalah sebagai berikut:

a. Unary Relationship

Unary relationship adalah model yang terjadi diantara entity yang berasal dari *entity* set yang sama. Sering juga disebut sebagai *recursive relationship* atau *reflective relationship*.

b. *Binary Relationship*

Adalah model *relationship* antara *instance-instance* dari suatu tipe entitas (dua *entity* yang berasal dari *entity* yang sama). *Relationship* ini paling umum digunakan dalam pembuatan model data.

c. *Ternary Relationship*

Ternary Relationship merupakan *relationship* antara *instance-instance* dari tiga tipe entitas sepihak.

2.2.7. Kunci Elemen Data (*Key*)

Menurut pendapat Ladjamudin (2013:138), menyatakan bahwa “*key* adalah elemen *record* yang dipakai untuk menemukan *record* tersebut pada waktu akses, atau bisa digunakan untuk mengidentifikasi setiap *entity/record/baris*.”

Menurut Ladjamudin (2013:138), terdiri dari 6 buah jenis *key* yaitu:

1. *Superkey*

Superkey merupakan satu atau lebih atribut (kumpulan atribut), dari satu tabel yang dapat digunakan untuk mengidentifikasi *entity/record* dari tabel tersebut secara unik (tidak semua atribut dapat menjadi *superkey*).

2. *Candidate key*

Superkey dengan jumlah atribut yang minimal, disebut dengan *candidate key*. *Candidate key* tidak boleh berisi atribut dan tabel yang lain, sehingga *candidate key* sudah pasti *superkey* namun belum tentu sebaliknya.

3. *Primary Key*

Salah satu atribut dari *candidate key* dapat dipilih /ditentukan menjadi *primary key* dengan 3 kriteria sebagai berikut.

- a. *Key* tersebut lebih natural untuk digunakan sebagai acuan.
- b. *Key* tersebut lebih sederhana.
- c. *Key* tersebut terjamin keunikannya.

4. *Alternate key*

Setiap atribut dari *candidate key* yang tidak terpilih menjadi *primary key*, maka atribut-atribut tersebut dinamakan *alternate key*.

5. *Foreign Key*

Foreign key merupakan sembarang atribut yang menunjuk kepada *primary key* pada tabel yang lain. *Foreign key* akan terjadi pada suatu relasi yang memiliki kardinalitas *one to many* (satu banyak) atau *many to many* (banyak ke banyak). *Foreign key* biasanya selalu diletakkan pada tabel/relasi yang mengarah ke banyak.

6. *External Key*

External key merupakan suatu *lexical attribute* (atau himpunan *lexical attribute*) yang nilai-nilainya selalu mengidentifikasi satu *object instance*.

2.2.8. Pengkodean

Pengkodean sering digunakan dalam perancangan suatu program dimana kode yang dibuat tersusun dari aturan-aturan yang dirancang berdasarkan elemen- elemen tertentu yang digunakan oleh perancang. Menurut Jogiyanto (2009:384), “struktur kode adalah suatu bentuk struktur yang berfungsi untuk mengklasifikasikan data, memasukkan data ke dalam komputer dan untuk mengambil bermacam-macam informasi yang berhubungan dengannya”.

Beberapa kemungkinan susunan digit (angka), huruf dan karakter-karakter khusus dapat dirancang ke dalam bentuk kode. dalam merancang suatu

kode harus diperhatikan beberapa hal (Jogiyanto, 2009:384), yaitu:

1. Mudah diingat

Supaya kode mudah diingat maka dapat dilakukan dengan cara menghubungkan kode tersebut dengan objek yang diwakili dengan kodenya.

2. Unik

Kode harus unik untuk masing-masing item yang diwakilinya.

3. Fleksibel

Kode harus fleksibel sehingga memungkinkan perubahan atau penambahan *item* baru dapat tetap diwakili oleh kodenya masing-masing.

4. Efisien

Kode harus sesingkat mungkin, selain mudah diingat juga akan efisien apabila direkam dan disimpan di luar komputer.

5. Konsisten

Kode harus konsisten dengan kode yang telah digunakan.

6. Harus distandarisasi

Kode harus distandarisasi untuk seluruh tingkatan dan departemen dalam organisasi.

7. Hindari penggunaan spasi

Spasi dalam kode sebaiknya dihindarkan karena dapat menyebabkan kesalahan dalam menggunakannya.

8. Hindari karakter yang mirip

Karakter-karakter yang hampir mirip bentuk dan bunyi pengucapannya sebaiknya tidak digunakan dalam kode. Masing-masing kode yang sejenis harus memiliki panjang kode yang sama.

Ada beberapa macam tipe dari kode yang dapat digunakan, menurut Jogiyanto (2009:386), sebagai berikut:

1. Kode Mnemonik (*Mnemonic Code*)

Kode mnemonik dibuat dengan dasar singkatan atau mengambil sebagian karakter dari *item* yang akan diwakili dengan kode ini.

2. Kode Urut (*Sequential Code*)

Kode yang nilainya urut antara satu kode dengan kode berikutnya.

3. Kode Blok (*Block Code*)

Kode ini mengklasifikasikan *item* ke dalam kelompok blok tertentu yang mencerminkan satu klasifikasi tertentu atas dasar pemakaian maksimum yang diharapkan.

4. Kode Group (*Group Code*)

Kode yang dibentuk berdasarkan *field-field* dan tiap-tiap *field* kode mempunyai arti.

5. Kode Desimal (*Decimal Code*)

Kode ini mengklasifikasikan dalam bentuk sepuluh unit angka desimal dimulai dari angka 0 sampai dengan angka 9 atau dari 00 sampai angka 99 tergantung dari banyaknya kelompok.

2.2.9. HIPO (Hierarchy Input Process Output)

HIPO sering digunakan untuk teknik pendokumentasian program yang digunakan untuk mengkomunikasikan spesifikasi sistem kepada para programmer melalui proses perancangan.

Menurut Ladjamudin (2013:211), “HIPO merupakan tehnik untuk mendokumentasikan sistem pemrograman”. Menurut Jogiyanto (2017:1),

mengemukakan “metodologi yang dikembangkan dan didukung oleh IBM. HIPO sebenarnya adalah alat dokumentasi program”.

Berdasarkan pengertian para ahli di atas, dapat disimpulkan bahwa HIPO merupakan metodologi yang dikembangkan dan didukung oleh IBM yang sebenarnya adalah alat dokumentasi program dan sekarang banyak digunakan sebagai alat *design* dan teknik dokumentasi dalam pengembangan sistem.