

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Teori yang Berhubungan dengan Sistem Secara Umum**

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

Sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama. Secara garis besar, sebuah sistem terdiri atas tiga komponen utama. Ketiga komponen itu mencakup *software*, *hardware* dan *brainware*. Ketiga komponen ini saling berkaitan satu sama lain. (Agus, 2014:7)

##### *1. Software*

*Software* mencakup semua perangkat lunak yang dibangun dengan bahasa pemrograman tertentu, pustaka, untuk kemudian menjadi sistem operasi, aplikasi dan *driver*. Sistem operasi, aplikasi, *driver*, saling bekerja sama agar komputer dapat berjalan dengan baik.

##### *2. Hardware*

*Software* mencakup semua perangkat keras (*motherboard*, *processor*, VGA, dan lainnya) yang disatukan menjadi sebuah komputer. Dalam konteks yang luas, bukan hanya sebuah komputer, namun sebuah jaringan komputer.

### 3. *Brainware*

*Brainware* mencakup kemampuan otak manusia, yang mencakup ide, pemikiran, analisis di dalam menciptakan dan menggabungkan *hardware* dan *software*.

Suatu sistem dapat terdiri dari sistem-sistem bagian (subsistem). Misalnya, sistem komputer dapat terdiri dari subsistem perangkat keras dan subsistem perangkat lunak. Masing-masing subsistem dapat terdiri dari subsistem-subsistem yang lebih kecil lagi atau terdiri dari komponen-komponen. Subsistem-subsistem saling berinteraksi dan saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat tercapai.

Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen atau variabel-variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu. (Sutabri, 2003:3)

Berdasarkan beberapa pengertian sistem dari beberapa ahli, penulis menyimpulkan bahwa sistem adalah sekumpulan elemen-elemen yang saling berhubungan dan bekerjasama satu sama lain untuk mencapai sasaran atau tujuan tertentu.

#### **2.1.2 Pengertian Informasi**

Informasi merupakan hasil pengolahan data dari satu atau berbagai sumber, yang kemudian diolah sehingga memberikan nilai, arti dan manfaat. Proses pengolahan ini memerlukan teknologi. Berbicara mengenai teknologi

memang tidak harus berkaitan dengan komputer, namun komputer sendiri merupakan salah satu bentuk teknologi. Dengan kata lain, alat tulis dan mesin ketik pun dapat dimasukkan sebagai salah satu teknologi yang digunakan selain komputer dan jaringan komputer. (Agus, 2014:9)

Informasi merupakan salah satu sumber daya penting dalam suatu organisasi, digunakan sebagai bahan pengambilan keputusan. Sehubungan dengan hal itu, informasi haruslah berkualitas. (Kadir, 2003:546)

Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi akan mengolah data menjadi menjadi informasi atau mengolah data dari bentuk tak berguna menjadi berguna bagi yang menerimanya. Nilai informasi berhubungan dengan keputusan. Bila tidak ada pilihan atau keputusan maka informasi tidak diperlukan. Keputusan dapat berkisar dari keputusan berulang sederhana sampai keputusan strategis jangka panjang. Nilai informasi dilukiskan paling berarti dalam konteks pengambilan keputusan. (Sutabri, 2003:18)

Berdasarkan pendapat para ahli tentang pengertian informasi, maka penulis menyimpulkan bahwa informasi adalah hasil akhir dari pengolahan data yang telah memiliki arti yang berguna bagi pemakainya dan dapat digunakan untuk pengambilan suatu keputusan yang tepat.

### 2.1.3 Pengertian Sistem Informasi

Menurut Alter (1992), sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi. Menurut Hall (2001), sistem informasi adalah sebuah rangkaian prosedur formaaal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai (Kadir, 2014:5)

Sistem informasi dapat terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*), blok dasar data (*database block*), dan blok kendali (*control block*). Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya. (Agus, 2014:10)

1. Blok Masukan. Input mewakili data yang masuk ke dalam sistem informasi. input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.
2. Blok Model. Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di dasar data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran. Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
4. Blok Teknologi. Teknologi merupakan “kotak alat” (*tool-box*) dari pekerjaan sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem keseluruhan.
5. Blok Basis Data. Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data di dalam basis data perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas.
6. Blok Kendali. Agar sistem informasi dapat berjalan sesuai dengan yang diinginkan, maka perlu diterapkan pengendalian-pengendalian di dalamnya. Beberapa pengendali perlu dirancang dan diterapkan untuk menyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan dapat langsung diatasi.

Berdasarkan pengertian sistem informasi dari beberapa ahli, maka penulis menyimpulkan bahwa sistem informasi adalah kesatuan elemen-elemen yang saling berinteraksi secara sistematis dan teratur untuk menciptakan informasi sebagai pendukung pengambilan keputusan.

#### 2.1.4 Pengertian Basis Data

Basis Data terdiri dari 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan dan sebagainya, yang mewujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya. (Fathansyah, 2012:2)

Sebagai salah satu istilah, basis data (*database*) sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

- a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
- c. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan

Basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk penyimpanan data agar dapat diakses dengan mudah dan cepat. (Rosa, 2013:43)

Berdasarkan beberapa pengertian yang dijelaskan oleh para ahli tentang pengertian basis data, maka penulis menyimpulkan bahwa basis data atau *database* merupakan suatu media penyimpanan data yang terdiri dari kumpulan data yang saling berkaitan secara logis, dan dirancang untuk memenuhi kebutuhan sebuah organisasi.

## **2.2 Teori yang Berhubungan dengan Topik yang Diangkat**

### **2.2.1 Definisi Surat Kabar**

Lasa (194:98) menjelaskan bahwa surat kabar terbitan yang menitik beratkan penyampaian berita yang mengandung informasi keilmuan yang perlu dikelola dengan baik. Menurut Komaruddin (2006:250) surat kabar yaitu:

1. Kertas yang dicetak dan didistribusikan, biasanya harian atau mingguan serta berisi berita, opini, karangan dan iklan.
2. Suatu alat komunikasi tertulis yang berisi berita, tajuk rencana, artikel, reportase, kadang-kadang disertai dengan tulisan hasil kesenian gambar, karikatur, surat pembaca dan iklan. (Khotimah, 2012:307)

Menurut Setyosari dan Sihkabudin surat kabar adalah media komunikasi massa dalam bentuk cetakan yang tidak perlu diragukan lagi peranan dan pengaruhnya terhadap masyarakat pada umumnya. Sedangkan menurut Kossach dan Sullivan surat kabar merupakan sumber bahan bacaan tambahan yang memungkinkan guru membawa komunikasi bahasa ke dalam kelas. (Saputra, 2012:3)

Berdasarkan beberapa pengertian yang dijelaskan oleh para ahli tentang pengertian Surat kabar adalah Surat kabar atau koran adalah barang cetakan yang berisi berita, informasi dan pendidikan yang terbit secara kontiniu yang biasanya harian. Surat kabar merupakan salah satu bentuk media cetak yang tidak dijilid.

### 2.2.2 Dalil Tentang Surat Kabar

Al-Qur'an memberi petunjuk bahwa berita yang perlu diperhatikan dan diselidiki adalah berita yang sifatnya penting. Adapun isu-isu ringan, omong kosong, dan berita yang tidak bermanfaat tidak perlu diselidiki, bahkan tidak perlu didengarkan, karena menyita waktu dan energi. Dalam Al-Qur'an surar Al-Hujurat Ayat 6 Allah berfirman:

يٰۤاَيُّهَا الَّذِيْنَ ءَامَنُوْا اِنْ جَاءَكُمْ فَاَسِقُ بِنَبَاٍ فَتَبَيَّنُوْا اَنْ تُصِيبُوْا قَوْمًا جٰهَلَةً  
فَتُصِيبُوْا عَلٰى مَا فَعَلْتُمْ نٰدِمِيْنَ ﴿٦﴾

Artinya : “Hai orang-orang yang beriman, jika datang kepadamu orang fasik membawa suatu berita, maka periksalah dengan teliti, agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya yang menyebabkan kamu menyesal atas perbuatanmu itu”.

Tafsir pada ayat diatas kita jumpai kalimat فَتَبَيَّنُوْا diterjemahkan dengan “*periksalah dengan teliti*”. Maksudnya telitilah berita itu dengan



cermat, tidak tergesa-gesa menghukumi perkara dan tidak meremehkan urusan, sehingga benar-benar menghasilkan keputusan yang benar.

Berita yang perlu dikorfirmasikan adalah berita penting, ditunjukkan dengan kata *naba'* untuk menyebut berita, bukan kata khabar. M. Quraish Shihab dalam bukunya *secercah cahaya ilahi* halaman 262 membedakan makna dua kata itu. Kata *naba'* menunjukkan berita penting sedangkan khabar menunjukkan berita secara umum.

نَحْنُ نَقُصُّ عَلَيْكَ أَحْسَنَ الْقَصَصِ بِمَا أَوْحَيْنَا إِلَيْكَ هَذَا الْقُرْآنَ وَإِن  
كُنْتَ مِنْ قَبْلِهِ لَمِنَ الْغَافِلِينَ ﴿٢٠٠﴾

Artinya: “Kami menceriterakan kepadamu kisah yang paling baik dengan mewahyukan Al Qur’an ini kepadamu, dan sesungguhnya kamu sebelum (Kami mewahyukan) nya adalah termasuk orang-orang yang belum mengetahui”.

## 2.3 Teori Yang Berhubungan Dengan Teknik Analisa Yang Digunakan

### 2.3.1 *Flowchart*


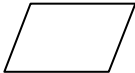
*Flowchart* (Diagram Alir) adalah bagan (*Chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. *Flowchart* merupakan metode untuk menggambarkan tahap-tahap pemecahan masalah dengan merepresentasikan simbol-simbol tertentu yang mudah dimengerti,


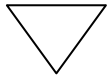
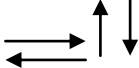


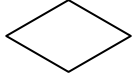


mudah digunakan dan standar. Tujuan penggunaan *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi, dan jelas dengan menggunakan simbol-simbol yang standar. Tahapan penyelesaian masalah yang disajikan harus jelas, sederhana, dan tepat. (Jogiyanto, 2000:795)

Pada waktu akan menggambar suatu bagan alir, *programmer* dapat mengikuti pedoman-pedoman sebagai berikut :

- a. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri suatu halaman.
- b. Kegiatan di dalam bagan alir harus ditunjukkan dengan jelas.
- c. Harus ditunjukkan dari mana kegiatan akan dimulai dan di mana akan berakhirnya.
- d. Masing-masing kegiatan di dalam bagan alir sebaiknya digunakan kata yang mewakili suatu pekerjaan.
- e. Masing-masing kegiatan di dalam bagan alir harus di dalam urutan yang semestinya.
- f. Kegiatan yang terpotong dan akan disambung di tempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
- g. Gunakan simbol-simbol bagan alir standar.

**Tabel 2.1 Simbol *Flowchart***

Simbol	Keterangan
	<p><b>Penghubung</b> Simbol untuk keluar/masuk atau proses dalam lembar atau halaman lain.</p>
	<p><b><i>Input Output</i></b> Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.</p>

	<p><b>Dokumen</b>          Simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau output dicetak di kertas.</p>
	<p><b>On Line Storage</b>          Simbol yang menunjukkan bahwa data di dalam simbol ini akan di simpan.</p>
	<p><b>Simbol Garis Alir</b>          Digunakan untuk menunjukkan arah selanjutnya yang akan dituju dari simbol-simbol dan <i>flowchart</i>.</p>
	<p><b>Simbol Manual</b>          Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer.</p>
	<p><b>Terminal</b>          Simbol yang menunjukkan untuk permulaan atau akhir suatu sistem.</p>
	<p><b>Kondisi</b>          Simbol keputusan yang menunjukkan kondisi.</p>
	<p><b>Proses</b>          Simbol yang menunjukkan pengolahan dilakukan oleh komputer.</p>
	<p><b>Penghubung</b>          Simbol untuk keluar/masuk prosedur atau proses dalam lembar atau halaman yang masih sama.</p>

Sumber : (Jogiyanto, 2000:796)

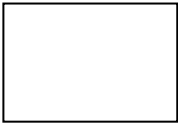

### 2.3.2 Data Flow Diagram (DFD)

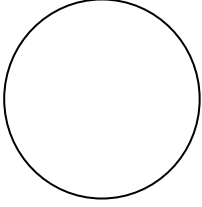
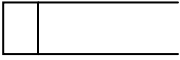
Pendekatan analisis terstruktur dikembangkan oleh Chris Gane dan Gane Sarson (1979) melalui buku metodologi struktur analisis dan desain sistem informasi. Mereka menyarankan untuk menggunakan Data Flow Diagram (DFD) dalam menggambarkan atau membuat model sistem.

Namanya, Data Flow Diagram seakan-akan mencerminkan penekanan pada data, namun sebenarnya DFD lebih menekankan segi proses. Adapun pengertian secara umum dari Data Flow Diagram ini adalah suatu network yang menggambarkan suatu sistem automat/komputerisasi, manualisasi atau gabungan dari keduanya, yang penggambarannya disusun dalam bentuk kumpulan komponen sistem yang saling berhubungan sesuai dengan aturan mainnya. (Rosa, 2013:69)

DFD sering juga digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem yang baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir (misal lewat telepon atau surat) serta lingkungan fisik dimana data tersebut akan disimpan (misal *hard disk* atau *disket*). Adapun simbol-simbol yang digunakan dalam DFD dapat dilihat pada tabel 2.2.

**Tabel 2.2 Simbol DFD**

Simbol	Keterangan
	<p><b><i>External Entity</i></b>            Kesatuan luar merupakan kesatuan (<i>entity</i>) dilingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada dilingkungan luarnya yang akan memberikan <i>input</i> atau menerima <i>output</i> dari sistem.</p>
	<p><b><i>Data Flow</i></b>            Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem</p>

	<p><b><i>Process</i></b></p> <p>Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin, atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses.</p>
	<p><b><i>Data Store</i></b></p> <p>Simpanan data merupakan simpanan data suatu <i>file</i>.</p>

Sumber : (Rosa, 2013:71)

### 2.3.2.1 Diagram Hubungan (*Context Diagram*)

Diagram hubungan merupakan level tertinggi dari DFD yang menggambarkan seluruh *input* ke sistem atau *output* dari sistem. Diagram konteks memberikan gambaran keseluruhan tentang sistem. *Entitas* (terminal) yang memberikan masukan ke sistem disebut *source*, sedangkan *entitas* (terminal) yang menerima keluaran disebut *sink*. Pada diagram ini hanya terdapat satu proses dan tidak boleh ada *datastore*.

### 2.3.2.2 Diagram 0 (Nol)

Pada diagram 0 (Nol) ditampilkan *datastore* yang digunakan. Untuk proses yang tidak dirinci lagi *level* selanjutnya ditambahkan \* pada akhir nomor proses. Keseimbangan *input* dan *output* antara diagram 0 dan diagram konteks harus diperhatikan. Aliran data yang masuk kedalam dan keluar dari proses harus sama dengan aliran data yang masuk dan keluar dari rincian proses tersebut.

### 2.3.2.3 Diagram Rinci

Diagram rinci merupakan rincian dari proses yang ada pada diagram 0 atau diagram *level* di atasnya. Keseimbangan antara diagram rinci dan diagram nol, dapat dilihat dari :

- a. Aliran data yang masuk dan keluar dari proses yang akan kita rinci,
- b. Jumlah *datastore* yang terhubung dengan proses yang akan kita rinci,
- c. Kondisi baca dan tulis di dalam *datastore*.

### 2.3.3 Kamus Data (*Data Dictionary*)

Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum. (Rosa, 2013:73)

Pada tahap perencanaan sistem, kamus data digunakan untuk merancang *input*, merancang laporan-laporan dan *database*. Simbol yang digunakan dalam kamus data dapat dilihat pada tabel 2.3.

**Tabel 2.3 Simbol Kamus Data**

Nama Simbol	Keterangan
=	Terdiri dari, terbentuk dari, sama dengan.
+	Penggabungan elemen data dengan elemen data lain.
{ }	Pengulangan elemen data.
[ ] dan	Memilih salah satu dari beberapa alternatif.
( )	Data tambahan, boleh ada boleh tidak.
*...*	Penjelasan atau keterangan tentang suatu data.

Sumber : (Rosa, 2013:74)

#### 2.3.4 *Entity Relationship Diagram (ERD)*

*ERD* adalah diagram yang memperlihatkan *entitas-entitas* yang terlibat dalam suatu sistem serta hubungan-hubungan atau relasi antar *entitas* tersebut. Model *Entity-Relationship* yang berisi komponen-komponen himpunan *entitas* dan relasi yang masing-masing dilengkapi dengan *atribut-atribut* yang merepresentasikan seluruh fakta dari “*Dunia Nyata*” yang ditinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan diagram *Entity-Relationship*. (Fathansyah, 2012:79)

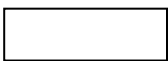
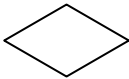

*ERD* menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data, pada dasarnya ada tiga macam komponen yang digunakan yaitu :

1. ***Entitas*** adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat. Sebagai contoh pelanggan, pekerja dan lain-lain.
2. ***Atribut*** berfungsi mendeskripsikan karakter entiti. Misalnya atribut nama pekerja dari entiti pekerja. Setiap entiti bisa terdapat lebih dari satu atribut.
3. **Hubungan** atau (***Relationship***) sebagaimana halnya entiti maka dalam hubunganpun harus dibedakan antara hubungan atau bentuk hubungan antar entiti dengan isi dari hubungan itu sendiri. Misalnya dalam kasus hubungan antara entiti siswa dan entiti mata\_kuliah adalah mengikuti,

sedangkan isi hubungannya dapat berupa nilai\_ujian. *Relationship* disimbolkan dalam bentuk intan / *diamonds*.

Simbol-simbol yang digunakan dalam *Entity Relationship Diagram* dapat dilihat pada tabel 2.4.

**Tabel 2.4 Simbol ERD**

Simbol	Keterangan
	<b>Entitas</b> Melambangkan himpunan entitas.
	<b>Relasi</b> Melambangkan himpunan relasi.
	<b>Penghubung</b> Melambangkan penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya.

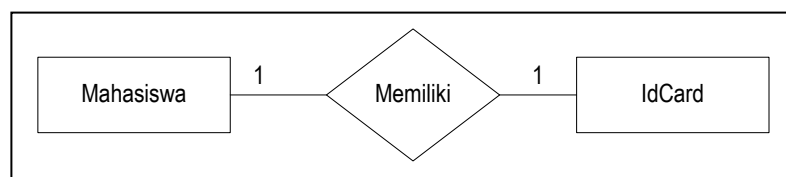
Sumber : (Fathansyah, 2012:80)

Jenis-jenis *relationship* (Fathansyah, 2007:80-81) :

1. Menurut Peter Chen :

b. *One – to – One*

Contoh :

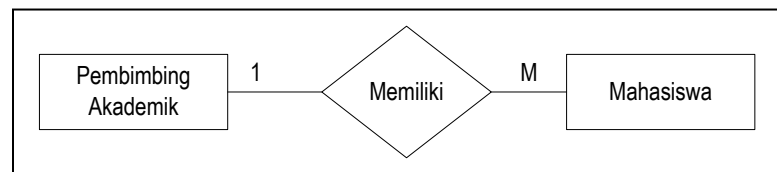




- 1) Setiap mahasiswa memiliki satu *Id Card*.
- 2) Satu *Id Card* hanya dimiliki oleh satu mahasiswa.

c. *One – to – Many* (1:M atau M:1)

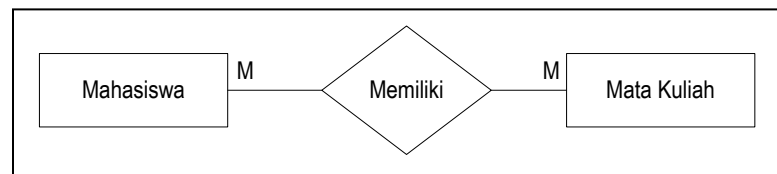
Contoh :



- 1) Pembimbing akademik memiliki banyak mahasiswa.
- 2) Satu mahasiswa hanya memiliki satu pembimbing akademik.

d. *Many – to – Many* (M:M)

Contoh :

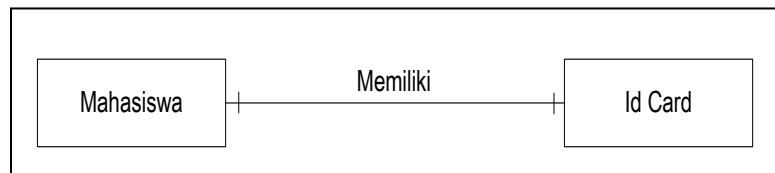


- 1) Mahasiswa memiliki banyak mata kuliah.
- 2) Mata kuliah dimiliki oleh banyak Mahasiswa.

2. Menurut James Martin, yaitu :

a. *One – to – One* (1:1)

Contoh :



- 1) Setiap mahasiswa memiliki satu *Id Card*.
- 2) Satu *Id Card* hanya dimiliki oleh satu mahasiswa.

b. *One – to – Many* (1:M atau M:1)

Contoh :



- 1) Pembimbing akademik memiliki banyak mahasiswa.
- 2) Satu mahasiswa hanya memiliki satu pembimbing akademik.

c. *Many – to – Many* (M:M)

Contoh :



- 1) Mahasiswa memiliki banyak mata kuliah.
- 2) Mata Kuliah dimiliki oleh banyak Mahasiswa.

Menentukan *primary key* merupakan salah satu hal yang penting, pertama tentukan suatu *attribute (field)* yang menyebabkan setiap *entry (record)* menjadi unik, kedua bila *attribute* tersebut tidak ada, tentukan gabungan beberapa *attribute* yang menyebabkan setiap *entry* menjadi unik, dan yang ketiga, pilih *primary key* dengan jumlah *attribute* yang paling sedikit.

### 2.3.5 Normalisasi

Normalisasi merupakan peralatan yang digunakan untuk melakukan proses pengelompokan data menjadi tabel-tabel yang menunjukkan entitas dan relasinya (Fathansyah, 2012:39). Dalam proses normalisasi, persyaratan sebuah tabel masih harus dipecah didasarkan adanya kesulitan kondisi pengorganisasian data seperti untuk menambah atau menyisipkan, menghapus atau mengubah, serta pembacaan data dari tabel tersebut. Bila masih ada kesulitan, maka tabel harus dipecah menjadi beberapa lagi, dan dilakukan proses normalisasi kembali sampai diperoleh tabel yang optimal.

Tahapan dalam normalisasi (Fathansyah, 2012:65) :

- Normal pertama (1NF)

Hilangkan semua kelompok data yang berulang (*repeating group*) dan menentukan *primary-key*. Hasilnya adalah relasi dibagi-bagi menjadi relasi yang lebih sederhana.

- Normal kedua (2NF)

Semua non-key *atribute* (*Field* yang bukan *record-key*) harus *full dependent* kepada *primary-key*. Jika terjadi relasi yang lain maka dibuat relasi yang baru.

- Normal ketiga (3NF)

Hilangkan *transive dependency*, yaitu sebuah non-key *atribute* yang tergantung pada non-key *atribute* yang lain dalam satu relasi.

### 2.3.6 Bagan Terstruktur

Bagan terstruktur adalah gambaran atau pendefinisian dan mengilustrasikan organisasi sistem secara berjenjang ke dalam modul-modul dan sub modul yang digunakan untuk menggambarkan struktur dari suatu sistem (Sutabri, 2003:179). Bagan terstruktur menjelaskan :

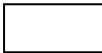




- a. Komunikasi antara masukan dan keluaran
- b. Pembagian suatu sistem menjadi modul

Bagan terstruktur digunakan dalam perancangan terstruktur dari suatu sistem informasi yang menunjukkan hubungan elemen data dan elemen kontrol, modul dan hubungan antar modul dalam suatu program.

Dengan adanya bagan terstruktur dapat diketahui :

- a. Masukan dari sebuah modul
- b. Keluaran dari modul tersebut
- c. Apa yang diketahui oleh modul tersebut

**Tabel 2.5 Simbol Bagan Terstruktur**

Simbol	Keterangan
	<b>Module</b> Mengembangkan suatu modul (kumpulan perintah / instruksi program).
	<b>Connection</b> Menghubungkan suatu modul dengan modul yang lain
	<b>Loop</b> Menyatakan perulangan selama kondisi terpenuhi dalam suatu modul.
	<b>Decision</b> Simbol ini menunjukkan suatu penyeleksian kondisi di dalam modul.
	<b>Coaple</b> Menunjukkan suatu data atau elemen kontrol yang dikirimkan dari suatu modul ke modul lainnya.

Sumber : (Sutabri, 2003:180)

### 2.3.7 Spesifikasi File

Spesifikasi *file* memberikan rincian yang lebih lengkap. Hasil normalisasi data hanya menunjukkan atribut (*field*) apa saja yang terdapat dalam sebuah *file*. Spesifikasi *file* berisi kode *file*, nama *file*, organisasi, *record key* dan deskripsi *field* (nama tipe, panjang, desimal, keterangan). *Record key* boleh lebih dari satu jenis dan boleh bebarapa *field*. Tipe field antara lain *numeric*, *character*, *date*, *Boolean*.

Contoh :

Nama File : Surat Transfer

Kode File : Tabel Surat\_Transfer

Primary Key : No\_ST

Foreign Key : No\_MPP

No	Nama Field	Type	Panjang	Keterangan
1	No_ST	<i>character</i>	7	Nomor Transaksi ST
2	Tgl_ST	<i>Date/ Time</i>	-	Tanggal Transaksi ST
3	Kettr	<i>character</i>	50	Keterangan Transaksi ST
4	No_MPP	<i>character</i>	8	Nomor MPP

### 2.3.8 Spesifikasi Proses (*Pseudocode*)

*Pseudocode* merupakan kode yang mirip dengan kode pemrograman yang sebenarnya. Berasal dari kata *pseudo* yang berarti imitasi atau menyerupai dan *code* yang berarti program. (Jogiyanto,2000:765). *Pseudocode* ditulis berbasis bahasa pemrograman. Sehingga lebih tepat digunakan untuk menggambarkan algoritma yang akan dikomunikasikan kepada *programmer*.

Aturan penulisan *pseudocode* adalah :

- a. Satu *pseudocode* satu baris.
- b. Pisahkan modul-modul atau kelompok *pseudocode* dengan memberikan spasi beberapa baris untuk mempermudah pembacaan.

- c. *Pseudocode* ditulis dengan huruf kapital, sedangkan komentar atau variabel dengan huruf kecil.

Metode yang digunakan dalam pembuatan spesifikasi proses dapat berupa :

- a. Narasi

Penjelasan proses dalam bentuk kalimat.

Contoh :

Untuk setiap pembelian dari pelanggan harap diperiksa : apabila batas kredit dilampaui maka periksa histori pembayarannya. Apabila historinya kurang baik maka ditolak. Apabila historinya baik dan pembelian tidak melebihi satu juta rupiah mintalah persetujuan manajer, bila lebih dari satu juta rupiah maka ditolak.

- b. Bahasa Indonesia Terstruktur

MODULE Validasi\_penjualan

JIKA Saldo\_Piutang > Limit\_Kredit MAKA

JIKA Histori\_Bayar = Buruk MAKA

Tolak

SELAIN ITU

JIKA Nilai\_Barang <= 1 juta MAKA

Minta\_persetujuan\_manajer

SELAIN ITU

Tolak

AKHIR JIKA

AKHIR JIKA

AKHIR MODUL

c. Structured English

MODULE Validasi\_Penjualan

*IF* Saldo\_Piutang > Limit\_Kredit *THEN*

*IF* Histori\_Bayar = Buruk *THEN*

Tolak

*ELSE*

*IF* Nilai\_Barang <= 1 juta *THEN*

Minta\_persetujuan\_manajer

*ELSE*

Tolak

*END IF*

*END IF*

*ELSE*

Pembelian\_dilayani

*END MODULE*

d. *Model Pre/Post Conditions*

Contoh:

*Precondition1*



Pelanggan mencocokkan nomor\_rekening dengan nomor\_rekening pada penyimpanan keuangan dan dilakukan jika kode\_status sah.

*Postcondition1*

Dikeluarkan tanda\_pembayaran dengan nomor\_rekening dan nilai\_penjualan

#### d. Model Tabel Keputusan

Model dari tabel keputusan terdiri dari empat bagian utama, yaitu *condition stub*, *condition entry*, *action stub* dan *action entry*.

## 2.4 Teori yang Berhubungan dengan Bahasa Pemrograman yang digunakan

### 2.4.1 Bahasa Pemrograman Visual Basic 2010

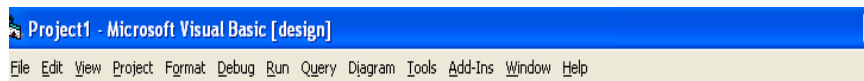
*Visual Basic* merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh microsoft, yaitu microsoft visual studio 2010. Sebagai produk lingkungan pengembangan terintegrasi atau IDE andalan yang dikeluarkan oleh microsoft, Visual Studio 2010 menambahkan perbaikan-perbaikan fitur-fitur baru yang lebih lengkap dibandingkan versi Visual Studio pendahulunya yaitu Microsoft Visual Studio 2008 (Sulistiyani, 2010:2).

*Microsoft Visual Basic* merupakan salah satu aplikasi pemrograman visual yang memiliki bahasa pemrograman yang cukup populer dan mudah untuk dipelajari. Basis 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. (Prabawati, 2010:1)

### 2.4.1.1 Elemen-elemen pada Visual Basic

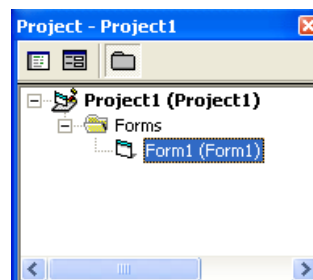
- a) Menu Bar, digunakan untuk memilih tugas-tugas tertentu seperti menyimpan project, membuka project, dan



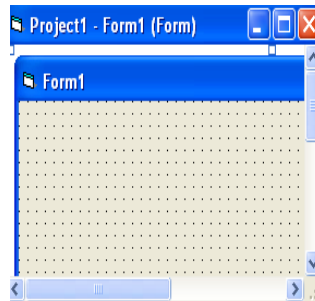
- b) Main *tool bars*, digunakan untuk melakukan tugas-tugas tertentu dengan cepat.



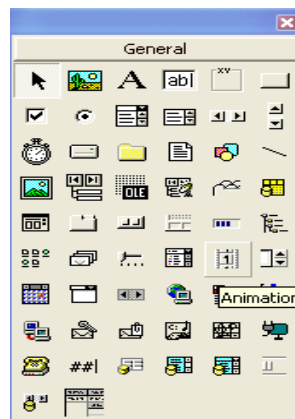
- c) Jendela *Project*, jendela ini berisi gambaran tentang dari semua modul yang terdapat dalam aplikasi.



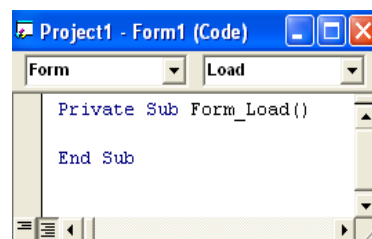
- d) Jendela *Form Designer*, jendela ini merupakan tempat untuk merancang *User Interface* dari aplikasi dan merupakan tempat untuk “menggambar” aplikasi.



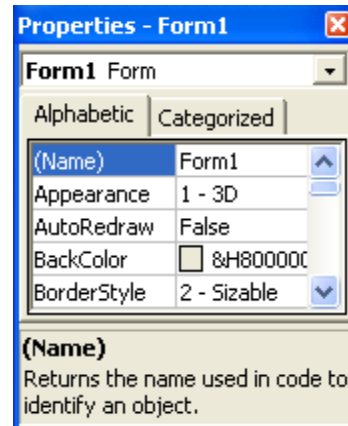
e) Jendela *Toolbox*, merupakan isi dari komponen-komponen yang dapat digunakan untuk mengembangkan *user interface*.



f) Jendela *Code*, merupakan tempat untuk menulis koding.



g) Jendela *Properties*, merupakan daftar properti-properti *object* yang sedang dipilih.



h) Jendela *Form Layout*, menunjukkan bagaimana form bersangkutan ditampilkan ketika *runtime*.



## 2.4.2 Microsoft Acces 2007

### 2.4.2.1 Sekilas Tentang Microsoft Acces 2007

*Microsoft Acces* adalah sebuah Program Aplikasi untuk mengolah Database ( Basis Data ) model relasional, karena terdiri dari lajur kolom dan lajur baris. Selain itu *Microsoft Acces* merupakan program aplikasi yang sangat mudah dan sangat familiar dalam pembuatan dan perancangan system manajemen database. *Microsoft Acces* saat ini banyak digunakan dalam

pembuatan aplikasi program yang sangat sederhana dan mudah, sedangkan program yang sejenis dengan *Microsoft Acces* antara lain yaitu Fox Pro, Visula Fox Pro, Fox Base, Dbase, Paradox dan lain – lain. (Suarna, 2008:11)

#### **2.4.2.2 Fungsi *Microsoft Acces 2007***

- a. untuk membuat program aplikasi persediaan barang
- b. untuk membuat program aplikasi gaji karyawan
- c. untuk membuat program aplikasi penjualan dan pembelian
- d. untuk membuat program aplikasi absensi
- e. untuk membuat program aplikasi administrasi pendidikan
- f. untuk membuat program aplikasi antrean kunjungan berobat
- g. dan lain-lain.

Dengan *Microsoft Acces* kita dapat merancang dan mengolah database yang saling terkait antara pembuatan Tabel, Form, Query, Report, Macro, Modul sehingga dapat dikoordinasikan dalam pembuatan: Tombol Macro, Menu Drop Down dan *Switchboard*

### 2.4.2.3 Mengenal Tampilan *Microsoft Acces 2007*

Tampilan utama pada saat kita menjalankan *Microsoft Acces 2007*



**Gambar 2.1** Tampilan *Microsoft Acces 2007*

Elemen dasar jendela kerja Access 2007 yang perlu diketahui adalah sebagai berikut.

- a. **Baris Judul (*Title Bar*)**, berisi nama program aplikasi dan nama file yang sedang aktif, baris judul ini dapat juga digunakan untuk memindahkan jendela ke posisi lain yang diinginkan.
- b. **Baris Menu (*Menu Bar*)**, berisi barisan perintah berupa menu, seperti menu *File, Edit View, Insert, Tools, Windows* dan *Help*
- c. **Baris Toolbar (*toolbars*)**, berisi tombol-tombol yang digunakan untuk menjalankan suatu perintah dengan cepat dan mudah.
- d. **Tombol Ukuran (*Sizing Button*)**, untuk mengatur ukuran jendela kerja Access 2007.
- e. **Task Pane**, berupa jendela yang ditampilkan esebelah kanan jendela kerja Access 2007 yang akan membantu pada saat mengedit *database*.

- f. **Jendela Database**, menampilkan file *database* yang sedang aktif, lengkap dengan seluruh objek *database*-nya.

### 2.4.3 Crystal Reports

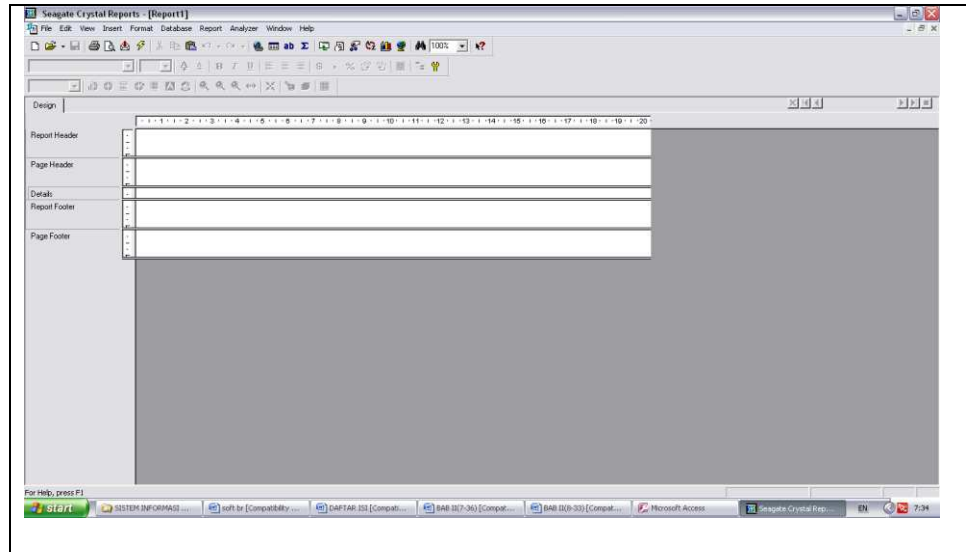
*Crystal Reports* merupakan program khusus untuk membuat laporan yang terpisah dengan program Microsoft Visual Basic 2010 tetapi keduanya dapat dihubungkan. Hasil mencetak dengan *crystal report* lebih baik dan lebih mudah karena pada *crystal reports* banyak tersedia obyek maupun komponen yang mudah digunakan. (Divisi Penelitian dan Pengembang Madcoms:159)

Dari segi membuat laporan, Crystal report tidak diragukan lagi, berikut ini merupakan kemampuan-kemampuan dari Crystal Report, yaitu:

1. Memilih layout laporan, dari laporan standar sampai form-form surat, serta juga dapat membuat lporan lebih baik lagi.
2. Menampilkan grafik yang begitu beragam, dengan menampilkan laporan data lebih rinci.
3. Mengkalkulasi total nilai keseluruhan (summary), subtotal, dan prosentase dari setiap kelompok data.
4. Menampilkan hasil akhir teratas dan terbawah dari seluruh data.
5. Kemampuan memutar objek teks dan memformat teks.

### 2.4.3.1 Tampilan dari *Crystal Report Designer*

Tampilan utama pada saat kita membuka *Crystal Report Designer*



**Gambar 2.2** Tampilan dari *Crystal Report Designer*

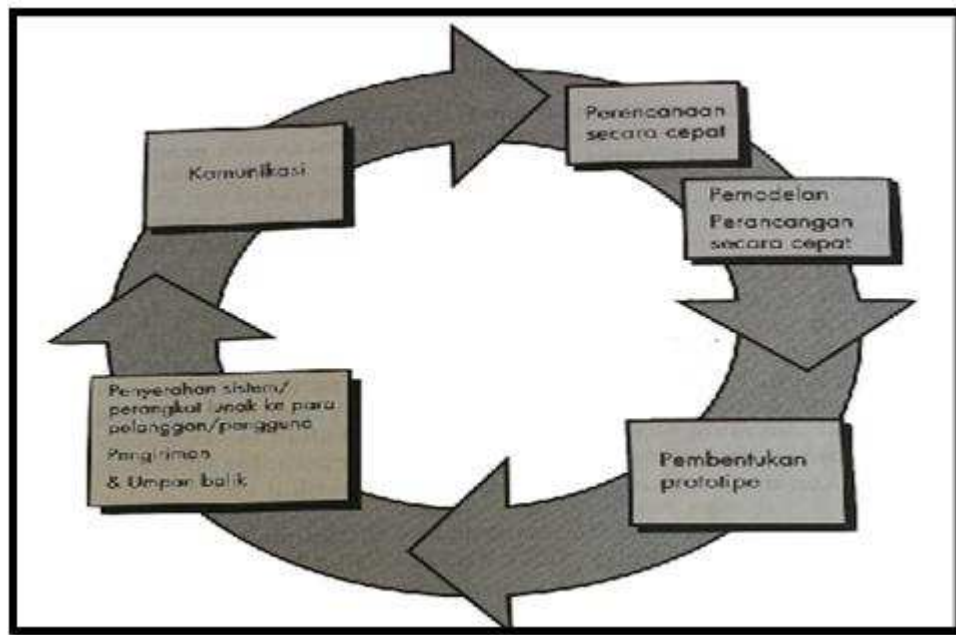
## 2.5 Metode Pengembangan Sistem

Menurut Rogert S. Pressman, Ph.D dalam bukunya yang berjudul *Rekayasa Lunak* (2010 : 50) Model Prototipe adalah sejumlah sasaran perangkat lunak secara umum, tetapi tidak bisa mengidentifikasi spesifikasi kebutuhan secara rinci untuk fungsi-fungsi dan fitur-fitur yang nantinya akan dimiliki perangkat lunak yang akan dikembangkan. Dalam kasus yang lain pengembangan perangkat lunak mungkin merasa tidak pasti tentang efisiensi suatu algoritma yang akan digunakan dalam pengembang perangkat lunak atau juga merasa tidak pasti akan kemampuan perangkat lunak untuk beradaptasi dengan sistem



operasi yang akan digunakan atau merasa tidak pasti akan bentuk interaksi manusia-komputer yang akan digunakan.

Meskipun pembuatan prototipe dapat digunakan sebagai model proses yang berdiri sendiri, pembuatan prototipe lebih umum digunakan sebagai teknik yang dapat diimplementasikan didalam konteks setiap model proses perangkat lunak yang telah ada. Dalam hal ini, tidak terlalu peduli dengan dimana ia diterapkan, paradigma pembuatan prototipe seringkali membantu tim mengembangkan perangkat lunak dan para *stakeholder* untuk memahami lebih baik apa yang akan dikembangkan saat spesifikasi kebutuhan belum jelas.



**Sumber** : Prototipe Rogert S. Pressman, Ph.D

**Gambar 2.3** Model prototipe

Dalam gambar di atas tahapam pembuatan prototipe dimulai dengan :

1. Komunikasi

Dilakukannya komunikasi antara tim pengembang perangkat lunak dengan pelanggan. Tim pengembang perangkat lunak akan melakukan pertemuan-pertemuan dengan para *stakeholder* untuk mendefinisikan sasaran keseluruhan untuk perangkat lunak yang akan dikembangkan, mengidentifikasi spesifikasi kebutuhan apapun yang saat ini diketahui dan menggambarkan area-area dimana definisi lebih jauh pada iterasi selanjutnya merupakan keharusan. Iterasi pembuatan prototipe direncanakan dengan cepat dan dan permodelan (dalam bentuk “rancangan cepat”) dilakukan.

2. Perancangan dan pemodelan secara cepat

Suatu rancangan cepat berfokus pada representasi semua aspek perangkat lunak yang akan terlihat oleh para pengguna akhir (misalnya rancangan antarmuka pengguna [*user interface*] atau format tampilan). Rancangan cepat (*quick design*) akan memulai konstruksi perbuatan prototipe. Prototipe kemudian akan diserahkan kepada para *stakeholder* dan kemudian mereka akan melakukan evaluasi-evaluasi tertentu terhadap prototipe yang telah dibuat sebelumnya, kemudian akhirnya akan memberikan umpan-balik yang akan digunakan untuk memperhalus spesifikasi kebutuhan.

### 3. Perbentukan Prototipe

Dalam kebanyakan proyek perangkat lunak, sistem yang dibentuk pertama kali (baca prototipe awal) biasanya bukan merupakan sistem yang dapat langsung digunakan oleh pengguna. Sistem yang dibentuk pertama kali tadi mungkin terlalu lambat, terlalu besar, terlalu sukar digunakan, atau ketiganya. Tidak ada alternatif pengembang sistem yang lain tanpa harus mengulangi analisis dan perancangan dari awal dengan cara yang cerdas dan mengembangkan versi yang dirancang ulang untuk menyelesaikan permasalahan-permasalahan yang dijumpai.

### 4. Penyerahan sistem perangkat lunak ke para pelanggan/pengguna pengirim dan umpan-balik.

Prototipe kemudian akan diserahkan kepada para *stakeholder* dan kemudian mereka akan melakukan evaluasi-evaluasi tertentu terhadap prototipe yang telah dibuat sebelumnya, kemudian akhirnya akan memberikan umpan-balik yang akan digunakan untuk memperhalus spesifikasi kebutuhan. Program yang sudah jadi diuji oleh pelanggan, dan bila ada kekurangan pada program bisa ditambahkan.

## 2.6 Hasil Penelitian Terdahulu

Agar penelitian ini dapat dipertanggung jawabkan secara akademis, maka peneliti menampilkan beberapa penelitian terhadap sistem informasi yang dibangun, yaitu:

### **2.6.1.1 Sistem Informasi Koran Berbasis *Web* Pada Harian Bisnis Radar Palembang**

Dengan menggunakan Metode *Web Engineering* Skripsi yang ditulis oleh M. Husni Mubarak yang berjudul, “Sistem Informasi Langganan Koran Berbasis *Web* Pada Harian Bisnis Radar Palembang dengan menggunakan Metode *Web Engineering*”. Penelitian ini bertujuan untuk pengembangan sistem langganan koran *online*, Harian Bisnis Radar Palembang Diharapkan Menjadi Solusi dalam penyampaian berita bagi masyarakat luas yang diharapkan sebagai kemudahan bagi masyarakat untuk menjadi pelanggan. Pengumpulan data yang diperlukan untuk pengembangan sistem langganan koran ini adalah melalui studi pustaka dan metode penelitian.

### **2.6.2 Skripsi yang berjudul “Membangun Koran *Online* Tribun Jabar” oleh Gian Pratama pada tahun 2008**

Memiliki tujuan mempunyai *Website* untuk menampilkan informasi yang nantinya akan diakses masyarakat luas. Koran *online* ini juga dimaksudkan bisa membantu redaktur dalam mengelolah berita. Peneliti pada pembuatan skripsi ini menggunakan Flowchart, DFD dan ERD, bertujuan untuk menggambarkan model fungsional dan ERD menggunakan model data software yang digunakan untuk membangun aplikasi ini yaitu PHP, Macromedia Dreamweaver dan MySql sebagai pendukung data basenya.

Dari beberapa penelitian di atas dapat diambil kesimpulan bahwa pemograman yang digunakan adalah PHP dan MySql, serta pemodelan yang

digunakan adalah Flowchart, DFD dan ERD ini semoga dapat memberi pemahaman bagi peneliti selanjutnya.