

BAB II

TINJAUAN PUSTAKA

Berikut adalah beberapa teori seperti pengertian dasar, teori pengembangan sistem, teori alat pengembangan sistem dan teori perangkat lunak yang digunakan dalam penelitian yang dilakukan. Teori-teori tersebut adalah:

2.1 Pengertian Dasar

Berikut beberapa teori pengertian dasar yang berhubungan dengan penelitian yang dilakukan. Teori-teori tersebut adalah:

2.1.1 Sistem

Menurut Sutanta (2003), sistem dapat didefinisikan sebagai sekumpulan hal atau kegiatan atau elemen atau subsistem yang saling bekerja sama atau dihubungkan dengan cara-cara tertentu sehingga membentuk satu kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan. Adapun karakteristik yang dimaksud adalah sebagai berikut:

1. Mempunyai komponen (*component*)

Komponen sistem adalah segala sesuatu yang menjadi bagian penyusun sistem. Komponen sistem dapat berupa benda nyata ataupun abstrak. Komponen sistem disebut sebagai subsistem, dapat berupa orang, benda, hal atau kejadian yang terlibat di dalam sistem.

2. Mempunyai batas (*boundary*)

Batas sistem diperlukan untuk membedakan satu sistem dengan sistem yang lain. Tanpa adanya batas sistem, maka sangat sulit untuk menjelaskan suatu sistem. Batas sistem akan memberikan batasan *scope* tinjauan terhadap sistem.

3. Mempunyai lingkungan (*environment*)

Lingkungan sistem adalah segala sesuatu yang berada di luar sistem. Lingkungan sistem dapat menguntungkan ataupun merugikan.

4. Mempunyai penghubung (*interface*)

Penghubung dapat didefinisikan segala sesuatu yang bertugas menjembatani hubungan antar komponen dalam sistem dan merupakan sarana yang memungkinkan setiap komponen saling berinteraksi dan berkomunikasi dalam rangka menjalankan fungsi masing-masing komponen.

5. Mempunyai masukan (*input*)

Masukan yaitu segala sesuatu yang dimasukkan ke dalam sistem sebagai bahan yang akan diolah lebih lanjut untuk menghasilkan keluaran yang berguna. Dalam hal ini, masukan disebut sebagai data.

6. Mempunyai keluaran (*output*).

Keluaran merupakan komponen sistem yang berupa berbagai macam bentuk keluaran yang dihasilkan oleh komponen pengolahan. Dalam hal ini keluaran adalah informasi yang dihasilkan dari pengolahan program aplikasi yang dapat digunakan sebagai bahan pengambilan keputusan.

7. Mempunyai pengolahan (*process*)

Pengolahan adalah berupa program aplikasi komputer yang dikembangkan untuk keperluan khusus dan dapat menerima masukan, mengolah masukan, dan menampilkan hasil olahan sesuai dengan kebutuhan para pemakai.

8. Mempunyai sasaran (*objective*)

Sasaran adalah apa yang ingin dicapai oleh sistem untuk jangka waktu pendek dan merupakan hasil dari setiap tahapan tertentu yang mendukung upaya pencapaian tujuan.

9. Mempunyai kendali (*control*)

Kendali dapat berupa validasi masukan, validasi proses, maupun validasi keluaran yang dapat dirancang dan dikembangkan secara terprogram. Kendali merupakan sangat perlu dijaga agar sistem tetap bekerja sesuai dengan peran dan fungsinya masing-masing.

10. Mempunyai umpan balik (*feedback*)

Umpan balik diperlukan oleh bagian kendali sistem untuk mengecek terjadinya penyimpangan proses dalam sistem dan mengembalikannya ke dalam kondisi normal.

2.1.2 Informasi

Menurut Sutanta (2003), informasi adalah hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung saat itu juga atau secara tidak langsung pada saat mendatang.

2.1.3 Sistem Informasi

Hasil penelitian yang berjudul “SISTEM INFORMASI GEOGRAFIS RUMAH SAKIT DI KOTA BANDAR LAMPUNG BERBASIS WEB” yang dilakukan oleh Nurjoko menyebutkan bahwa ada beragam definisi dari sistem informasi, salah satunya menurut Hall (2001), sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi dan didistribusikan kepada pemanakainya.

2.1.4 *Customer Relationship Management (CRM)*

Customer Relationship Management (CRM) adalah sistem manajemen yang secara khusus menangani hubungan antara perusahaan dengan konsumennya dengan tujuan meningkatkan nilai perusahaan di mata para konsumennya (“Studi Kasus Sistem Informasi Manajemen”, 2018). CRM melingkupi semua aspek yang berhubungan dengan calon konsumen dan konsumen saat ini, termasuk di dalamnya adalah pusat panggilan, tenaga penjualan, pemasaran, dukungan teknis, dan layanan lapangan. Dalam penerapannya, CRM memiliki beberapa tujuan, diantaranya sebagai berikut: mendapatkan pelanggan, mengetahui pelanggan, mempertahankan pelanggan, mengembangkan pelanggan, merubah pelanggan yang belum menguntungkan menjadi menguntungkan, (Lukas, 2001).

Dalam jurnal yang berjudul “E-CUSTOMER RELATIONSHIP MANAGEMENT (CRM) UNTUK SISTEM INFORMASI PAKET WISATA PADA CV ALEA TOUR & TRAVEL BANDAR LAMPUNG” karya Halimah dan Dian Kinanti, CRM mendukung suatu perusahaan untuk menyediakan pelayanan kepada pelanggan secara real time dan menjamin hubungan dengan tiap pelanggan melalui penggunaan informasi tentang pelanggan.

Lukas (2001) mengatakan bahwa keberhasilan CRM ditentukan oleh tiga faktor utama yaitu manusia, proses dan teknologi untuk mengoptimalkan hubungan organisasi dengan semua tipe pelanggan. CRM dibagi ke dalam tiga komponen utama, sebagai berikut:

1. Manusia. Dikarenakan CRM sebenarnya adalah bagaimana mengelola hubungan atau relasi antara manusia sehingga diperlukan sentuhan pribadi dan manusiawi, oleh karena itu manusia adalah utama.

2. Proses. Disamping itu dibutuhkan proses yaitu sistem dan prosedur yang membantu manusia untuk dapat menjalin hubungan dekat dengan pelanggan.
3. Identifikasi. Pada aktivitas proses ini perusahaan dituntut memiliki analisa yang cukup kuat terhadap prospek, siapa pelanggan yang menguntungkan, mengapa dia menguntungkan, dan sebagainya.

Menurut Montana dan Noor (2010), CRM juga memiliki beberapa keuntungan dalam bisnis dan dapat dirangkum sebagai berikut:

1. Usaha pemasaran yang lebih efektif.
2. Interaksi yang lebih efektif dengan konsumen.
3. Keuntungan jangka panjang dari hubungan yang berkelanjutan.
4. Berbagi pengetahuan.
5. Menghemat biaya.
6. Tenaga penjualan yang lebih efektif dan efisien.

Menambahkan nilai pada apa yang kita tawarkan akan berdampak pada tingkat kepuasan pelanggan, dan meningkatnya tingkat kepuasan pelanggan dapat meningkatkan loyalitas pelanggan, (Barnes, 2007). Loyalitas atau kesetiaan didefinisikan sebagai komitmen yang dipegang kuat untuk membeli atau berlangganan lagi produk atau jasa tertentu di masa depan meskipun ada pengaruh situasi dan usaha pemasaran yang berpotensi menyebabkan perubahan perilaku, (Kotler dan Keller, 2012). Loyalitas konsumen dapat dikelompokkan menjadi dua yaitu loyalitas merek dan toko. Loyalitas merek adalah sikap menyenangkan terhadap suatu merek yang direpresentasikan dalam pembelian yang konsisten terhadap merek itu sepanjang waktu, sedangkan loyalitas toko juga ditunjukkan oleh perilaku konsisten tetapi perilakunya adalah mengunjungi toko dimana di tempat tersebut konsumen bisa membeli merek yang diinginkan, (Sutisna, 2006).

Menurut Peter dan Olson (2006), loyalitas dapat diukur berdasarkan:

1. Urutan pilihan. Metode ini banyak dipakai dalam penelitian dengan menggunakan panel-panel agenda harian konsumen lainnya, dan lebih terkini lagi, data *scanner* supermarket.
2. Proporsi pembelian. Metode ini menggunakan pengujian terhadap proporsi pembelian total dalam sebuah kelompok produk. Data yang dianalisis berasal dari panel konsumen.
3. Preferensi. Metode ini menggunakan komitmen psikologis atau pernyataan preferensi.

Hasil penelitian yang berjudul “PENGEMBANGAN APLIKASI CUSTOMER RELATIONSHIP MANAGEMENT RUMAH SAKIT” yang dilakukan oleh Rachmad Hidayat dan tim menyebutkan bahwa adapaun CRM yang diterapkan yang sesuai dengan tema peneliti adalah seperti :

1. Profil, menjelaskan mengenai rumah sakit.
2. Fasilitas, memberikan informasi mengenai fasilitas kepada pengunjung dan pasien.
3. Dokter, memberikan informasi mengenai dokter kepada pengunjung dan pasien.
4. Info Rekam Medik, merupakan fasilitas bagi pasien untuk mengetahui informasi rekam medik mereka.

2.1.5 Fasilitas Pelayanan Kesehatan (Fasyankes)

Berdasarkan Peraturan Pemerintah Republik Indonesia Nomor 47 Tahun 2016 tentang Fasilitas Pelayanan Kesehatan Bab I Pasal 1 Ayat 1, fasilitas pelayanan kesehatan adalah suatu alat dan/atau tempat yang digunakan untuk menyelenggarakan upaya pelayanan kesehatan, baik promotif, preventif, kuratif maupun rehabilitative yang dilakukan oleh pemerintah pusat, pemerintah daerah, dan/atau masyarakat.

Kemudian pada Bab II Pasal 4 Ayat 1 dijelaskan bahwa jenis fasyankes terdiri atas:

- a. Tempat praktik mandiri tenaga kesehatan
- b. Pusat kesehatan masyarakat
- c. Klinik
- d. Rumah sakit
- e. Apotek
- f. Unit transfusi darah
- g. Laboratorium kesehatan
- h. Optik
- i. Fasilitas pelayanan kedokteran untuk kepentingan hukum
- j. Fasilitas pelayanan kesehatan tradisional

2.1.6 Lokakarya

Lokakarya adalah pertemuan antara para ahli (pakar) untuk membahas masalah praktis atau yang bersangkutan dengan pelaksanaan dalam bidang keahliannya, (KBBI, 2019).

2.1.7 Wawancara

Wawancara adalah tanya jawab peneliti dengan narasumber, (KBBI, 2019).

2.2 Teori Pengembangan Sistem

Berikut adalah teori pengembangan sistem yang digunakan dalam penelitian yang sedang dilakukan, teori tersebut adalah:

2.2.1 Pemrograman terstruktur

Metode pemrograman terstruktur membagi-bagi program berdasarkan fungsi-fungsi atau prosedur-prosedur yang dibutuhkan program komputer. Pendekatan terstruktur digunakan karena mempunyai kelebihan dalam kemudahan untuk memahami sistem, dan konsep tentang dekomposisi permasalahan mulai dari paling dasar sampai paling detail cukup memudahkan dalam pemahaman bentuk sistem khususnya bagi pengguna yang mempunyai pemahaman tentang sistem cukup rendah (Sukanto & Shalahuddin, 2018).

2.2.2 James Martin Rapid Application Development (JMRAD)

Dalam pengembangan sistem ini, penulis menggunakan metodologi James Martin *Rapid Application Development* (JMRAD). RAD sendiri adalah metodologi pengembangan perangkat lunak yang berfokus pada rapid prototype dan pengembangan aplikasi. Menurut James Martin, metode ini dapat memberikan pengembangan yang jauh lebih cepat dan hasil yang lebih berkualitas tinggi daripada yang dicapai.

Selain itu, James Martin juga berpendapat JMRAD memiliki beberapa karakteristik, yaitu:

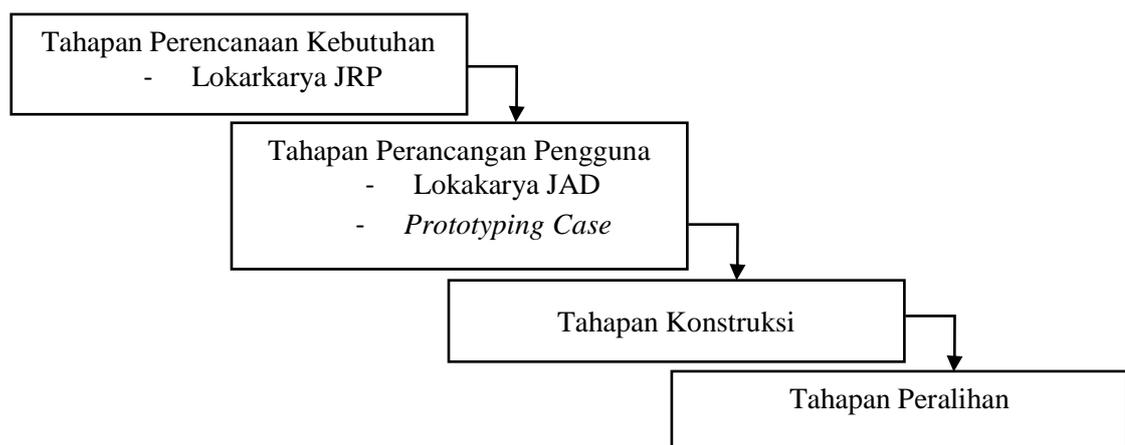
- 1 Mengadopsi pendekatan *evolutionary prototyping*.
- 2 Berfokus pada pengguna yang dianggap penting dan melibatkan mereka seawal mungkin dalam proses pengembangan sistem informasi.
- 3 Pengguna diminta berkomitmen terhadap pengembangan sistem informasi.
- 4 Menggunakan alat-alat bantu pengembangan sistem yang termutakhir.

Seperti yang sudah dijelaskan sebelumnya, pada poin nomor 2 dijelaskan bahwa JMRAD berfokus pada pengguna yang dianggap penting dan melibatkan mereka seawal mungkin dalam proses pengembangan sistem informasi, oleh karena itu penulis menanggapi bahwa JMRAD sangat cocok digunakan dalam pengembangan sistem khususnya yang bertemakan CRM yang pada dasarnya untuk meningkatkan layanan kepada pelanggan.

Hasil penelitian yang berjudul “STUDI ANALISIS RAPID APPLICATION DEVELOPMENT SEBAGAI SALAH SATU ALTERNATIF METODE PENGEMBANGAN PERANGKAT LUNAK” yang dilakukan oleh Agustinus Noertjahyana, menyebutkan bahwa salah satu alasan menggunakan RAD adalah Apabila menggunakan RAD untuk mendapatkan suatu desain yang dapat diterima oleh konsumen dan dapat dikembangkan dengan mudah.

2.2.3 Tahapan JMRAD

Ada 4 tahapan yang terdapat pada metodologi JMRAD, tahapan tersebut akan disajikan pada gambar dibawah ini.



Gambar 2.1 Siklus Hidup Pengembangan Sistem.

1. Tahapan Perencanaan Kebutuhan

JMRAD sangat menekankan pentingnya menemukan kebutuhan pengguna sistem. Ada dua teknik yang digunakan untuk mendefinisikan kebutuhan pengguna, yaitu *Joint Requirements Planning* (JRP) dan kemudian disusul dengan *Joint Application Design* (JAD). Pada tahap awal digunakan teknik JRP, JRP digunakan untuk mengidentifikasi kebutuhan manajemen (level atas) akan sistem informasi yang dikembangkan. Peserta dari JRP sendiri adalah para manajer level atas yang nantinya diharapkan dapat memberikan alasan yang kuat serta arahan sehingga sistem informasi yang dibuat dapat berkontribusi dengan baik. Tidak hanya itu, pada tahap ini juga diharapkan dapat menentukan berapa pengguna yang akan terlibat dalam sistem yang akan dibuat atau dikembangkan.

2. Tahapan Perancangan Pengguna

Tahap selanjutnya adalah dengan menggunakan teknik JAD, peserta dari JAD adalah semua pengguna yang ikut terlibat dalam sistem informasi tersebut. Dalam tahap ini menggunakan dua metode yaitu lokakarya dan wawancara bergantung dari narasumber yang dipilih dan jika sudah selesai dapat dilanjutkan ke tahap *Prototyping Case*. Tahap selanjutnya menggunakan alat bantu *prototype* secara intensif, *prototype* meliputi pengolahan data, antarmuka, laporan, tampilan layar, dan lainnya. Dalam tahap ini pengguna dilibatkan dalam prosesnya dan jika terdapat revisi maka dapat dilakukan revisi sampai pengguna menyatakan menerima dan memberikan persetujuan secara formal, namun pengguna juga diminta berkomitmen dengan terhadap dengan pengembangannya. Jika sudah selesai dapat melanjutkan ke tahapan selanjutnya.

3. Tahapan Konstruksi

Setelah tahap sebelumnya selesai, selanjutnya kebutuhan pengguna yang telah didefinisikan dalam tahap sebelumnya diterjemahkan ke dalam aplikasi dengan menuliskan *sourcecode* aplikasi yang dibangun. Dalam tahap ini jika diperlukan revisi oleh pengguna maka akan dilakukan revisi sampai pengguna menyatakan menerima dan memberikan persetujuan secara formal. Dalam tahap ini, semua revisi yang dilakukan oleh pengguna harus di dokumentasikan secara rapi guna menjadi dasar pembenaran munculnya fitur atau fungsi pada sistem informasi yang dibangun. Dikarenakan pengguna ikut terlibat tahap ini, para pengguna menjadi mengenal sistem dengan lebih baik dan proses belajar dengan sistem yang dikembangkan menjadi lebih cepat.

4. Tahapan Peralihan

Pada tahap ini merujuk pada upaya untuk mempersiapkan informasi bagi pengguna dalam menggunakan sistem yang telah dibangun untuk sehari-hari. Pada tahap ini juga dilakukan uji coba pada sistem untuk memastikan semua kebutuhan yang telah dikemukakan sebelumnya sudah terpenuhi. Jika sistem informasi yang baru berjalan dengan baik, maka sistem informasi yang lama dapat dihentikan penggunaannya.

2.3 Alat dan Teknik Pengembangan Sistem

Berikut adalah beberapa alat dan teknik pengembangan sistem yang digunakan dalam penelitian. Teori tentang alat dan teknik pengembangan sistem adalah sebagai berikut:

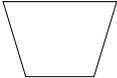
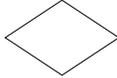
2.3.1 Bagan Alir (*Flowchart*)

Menurut Indrajani (2015), *flowchart* merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program. Biasanya mempermudah penyelesaian masalah khususnya yang perlu dievaluasi lebih lanjut, sedangkan menurut Rifka (2017), *flowchart* adalah sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran algoritma atau proses yang menampilkan langkah-langkah yang disimbolkan dalam bentuk kotak beserta urutannya. Simbol-simbol yang digunakan dalam *flowchart* terbagi menjadi 3 bagian, yaitu *flowchart direction*, *processing* dan *input/output*, (Jalinus dan Ambiyar, 2016). Penjelasan simbol-simbol pada masing-masing bagian *flowchart* akan disajikan dalam tabel 2.1, tabel 2.2 dan tabel 2.3 dibawah ini.

Tabel 2.1 Simbol *flowchart* untuk *direction*.

Simbol	Keterangan
	Simbol arus Menyatakan jalannya arus suatu data.
	Simbol konektor Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.
	Simbol luring konektor Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.

Tabel 2.2 Simbol *flowchart* untuk *processing*.

Simbol	Keterangan
	Simbol proses Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
	Simbol manual Menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer.
	Simbol <i>decision</i> Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban.
	Simbol <i>predefined process</i> Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
	Simbol terminal Menyatakan permulaan atau akhir suatu program.

Tabel 2.3 Simbol *flowchart* untuk masukan atau keluaran.

Simbol	Keterangan
	Simbol masukan/keluaran Menyatakan proses masukan atau keluaran tanpa tergantung jenis peralatannya.
	Simbol <i>magnetic tape</i> Menyatakan masukan berasal dari pita magnetis atau keluaran disimpan ke pita magnetis.
	Simbol tampilan Mencetak keluaran dalam layar monitor.

Tabel 2.3 Simbol *flowchart* untuk masukan atau keluaran.

Simbol	Keterangan
	Simbol <i>punched card</i> Menyatakan masukan berasal dari kartu atau keluaran ditulis ke kartu.
	Simbol dokumen Mencetak keluaran dalam bentuk dokumen (melalui printer).
	Simbol <i>disk storage</i> Menyatakan masukan berasal dari disk atau keluaran disimpan ke disk.

2.3.2 Bagan Alir Sistem (*System Flowchart*)

System Flowchart merupakan bagan yang menunjukkan alur kerja secara keseluruhan. Bagan tersebut menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem. Dengan kata lain, *system flowchart* merupakan deskripsi secara grafik dari urutan prosedur-prosedur yang terkombinasi dan membentuk suatu sistem (Rifka, 2017).

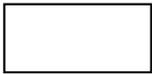
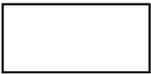
2.3.3 Bagan Alir Program (*Program Flowchart*)

Program Flowchart merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. *Program Flowchart* terdiri dari dua macam. Pertama, bagan alir logika program yang digunakan untuk menggambarkan tiap-tiap langkah di dalam program komputer secara logika dan dipersiapkan oleh analis sistem. Kedua, bagan alir program komputer terinci (Rifka, 2017).

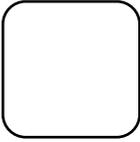
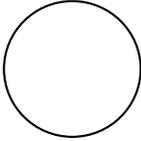
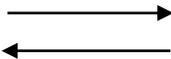
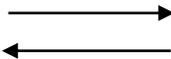
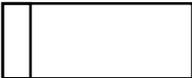
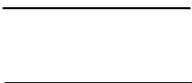
2.3.4 Diagram Alir Data (*Data Flow Diagram*)

Data Flow Diagram (DFD) atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan dan keluaran, (Sukamto & Shalahuddin, 2018). Notasi-notasi pada DFD memiliki dua versi, yaitu versi Gane-Sarson dan DeMarco-Yourdon. Perbedaan utama kedua notasi ini adalah pada simbol yang digunakan meskipun cara penggambaran dan membacanya sama. Berikut adalah notasi dan keterangan yang disajikan dalam tabel 2.4 di bawah ini

Tabel 2.4 Simbol – simbol *Data Flow Diagram*

Gane-Sarson	DeMarco-Yourdon	Keterangan
		Entitas Eksternal Entitas luar atau masukan atau keluaran atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan.

Tabel 2.4 Simbol – simbol *Data Flow Diagram* (lanjutan)

Gane-Sarson	DeMarco-Yourdon	Keterangan
		<p>Proses</p> <p>Proses atau fungsi atau prosedur. Pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program</p>
		<p>Aliran data.</p> <p>Merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukkan atau keluaran</p>
		<p>Penyimpanan Data</p> <p>Pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang seharusnya dibuat menjadi tabel basis data yang dibutuhkan dan tabel ini juga harus sesuai dengan perancangan tabel pada basis data</p>

Menurut Sarosa (2017), panduan untuk menggambar DFD adalah sebagai berikut:

1. Mendokumentasikan sistem dalam bentuk narasi.
2. Dari narasi tersebut dibuatlah tabel entitas dan aktivitas.
3. Gambar diagram konteks dengan menunjukkan satu proses dan satu atau lebih entitas eksternal beserta arah aliran data.
4. Dari diagram konteks, digambar diagram level 0 yang menjabarkan sistem ke dalam rinciannya. Diagram fisik dan logis digambarkan terpisah.
5. Tiap-tiap proses atau entitas eksternal dalam level 0 digambarkan lebih detail (jika ada) pada level selanjutnya.

Perlu diingat DFD tidak menunjukkan urutan proses maupun waktu eksekusi suatu proses.

Pembuatan DFD dibagi menjadi beberapa tahapan, berikut adalah tahapan perancangan dengan menggunakan DFD (Sukamto dan Shalahuddin, 2018):

1. Membuat DFD level 0 atau sering disebut juga *Context Diagram* DFD level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain.
2. Membuat DFD level 1
DFD level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan di kembangkan. DFD level 1 merupakan hasil *breakdown* DFD level 0 yang sebelumnya dibuat.

3. Membuat DFD level 2

DFD level 2 merupakan hasil *breakdown* modul-modul DFD level 1 yang sebelumnya dibuat. DFD level 2 merupakan penggambaran modul yang lebih detail dari modul yang terdapat pada DFD level 1. Apabila modul yang terdapat pada DFD level 2 sudah cukup terperinci, maka tidak diperlukan untuk membuat *breakdown* dari modul tersebut, namun sebaliknya jika belum terperinci maka akan dibuat *breakdown* dari modul tersebut.

4. Membuat DFD level 3 dan seterusnya

DFD level selanjutnya merupakan *breakdown* level sebelumnya dengan menggunakan aturan yang sama dengan DFD level 1 atau 2.

2.3.5 Kamus Data

Kamus data adalah kumpulan daftar element data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum, (Sukamto & Shalahuddin, 2018). Berikut contoh tampilan format kamus data pada gambar di bawah ini.

Nama Database	:		
Nama Tabel	:		
Primary Key	:		
Forign Key	:		
Nama Field	Type	Size	Keterangan

Gambar 2.2 Format Kamus Data.

2.3.6 Basis Data

Hasil penelitian yang berjudul “STUDI PENGEMBANGAN PROTOTYPE KNOWLEDGE MANAGEMENT PADA PENGECEKAN JUDUL TUGAS AKHIR ATAU SKRIPSI FAKULTAS ILMU KOMPUTER IBI DARMAJAYA” yang dilakukan oleh Neni Purwati dan Hendra Kurniawan, menyebutkan bahwa basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program untuk memperoleh informasi dari basis data tersebut. Sedangkan menurut Kusri (2007), basis data (*database*) adalah kumpulan data yang saling berelasi. Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan dan kecepatan dalam pengambilan kembali.

Untuk mencapai tujuannya, syarat sebuah basis data yang adalah sebagai berikut:

1. Tidak adanya redundansi dan inkonsistensi data. Redundansi terjadi jika suatu informasi disimpan di beberapa tempat yang mengakibatkan inkonsistensi data atau data yang tidak konsisten.
2. Kesulitan pengaksesan data. Untuk mengatasi hal tersebut, dapat digunakan *query* ataupun *tool* dari perangkat lunak DBMS untuk melihat dan mencari data secara cepat dan akurat.
3. Banyak pengguna. Basis data juga memungkinkan penggunaan data secara bersama-sama oleh banyak pengguna pada saat yang bersamaan maupun saat yang berbeda.

Basis data juga mempunyai banyak manfaat, berikut akan dijelaskan beberapa manfaat dari basis data menurut Kusri (2007):

1. Kecepatan dan Kemudahan (*speed*)
Dengan menggunakan basis data, pengambilan informasi dapat dilakukan dengan cepat dan mudah. Basis data memiliki kemampuan dalam mengelompokkan, mengurutkan bahkan perhitungan. Dengan perancangan yang benar, maka penyajian informasi dapat dilakukan dengan cepat dan mudah.
2. Kebersamaan Pemakai (*shareability*)
Sebuah basis data dapat digunakan oleh banyak pengguna dan banyak aplikasi. Untuk data-data yang diperlukan oleh banyak bagian/orang, tidak perlu dilakukan pencatatan di masing-masing bagian, tetapi cukup dengan satu basis data untuk dipakai bersama.
3. Pemusatan Kontrol Data
Karena cukup dengan satu basis data untuk banyak keperluan, maka pengontrolan data juga cukup dilakukan di satu tempat saja. Jika ada perubahan data, cukup dilakukan perubahan pada satu tempat saja tanpa perlu melakukan perubahan pada masing-masing bagian.
4. Efisiensi Ruang Penyimpanan (*space*)
Dengan pemakaian bersama, pengguna tidak perlu menyediakan tempat di berbagai tempat, tetapi cukup satu tempat saja sehingga ini akan menghemat ruang penyimpanan yang dimiliki oleh sebuah organisasi.
5. Keakuratan (*accuracy*)
Penerapan secara ketat aturan tipe data, domain data, keunikan data, hubungan antar data, dan lain lain, dapat menekan ketidakakuratan dalam pemasukan/penyimpanan data.
6. Ketersediaan (*availability*)
Dengan basis data, kita dapat membuat cadangan data, memilah data antara yang masih diperlukan dan data mana yang perlu disimpan di tempat lain.

7. Kelengkapan (*completeness*)

Untuk mengkomodasi kebutuhan kelengkapan data yang semakin berkembang, maka kita tidak hanya dapat menambah *record-record* data, tetapi juga dapat melakukan perubahan struktur dalam basis data, baik dalam penambahan objek baru (tabel) atau dengan penambahan *field-field* baru pada suatu tabel.

8. Keamanan (*security*)

Sudah banyak *Data Base Management System* (DBMS) yang dilengkapi dengan fasilitas manajemen pengguna yang mengatur hak akses yang berbeda sesuai dengan kepentingan dan posisinya sehingga meningkatkan keamanan data yang tersimpan.

9. Kemudahan dalam pembuatan program aplikasi baru

Dengan tersedianya basis data, pembuatan aplikasi baru dapat langsung memanfaatkan basis data yang tersedia sehingga pembuat aplikasi tidak perlu membuat ulang basis data, cukup mengatur tampilan aplikasi untuk pengguna.

10. Pemakaian Secara Langsung

Basis data memiliki fasilitas yang disediakan oleh DBMS untuk melihat data secara langsung yang ada di dalam tabel maupun dengan menggunakan *query*.

11. Kebebasan Data

Dengan adanya basis data, perubahan isi/struktur yang diperlukan hanya perlu dilakukan pada level DBMS tanpa harus membongkar kembali program aplikasinya.

12. Sudut Pandang Pengguna

Dengan adanya basis data, memungkinkan penyediaan sudut pandang yang berbeda-beda untuk tiap-tiap pengguna yang terlibat di dalamnya.

2.4 Perangkat Lunak Pendukung

Berikut adalah beberapa perangkat lunak pendukung yang digunakan dalam penelitian. Perangkat lunak pendukung adalah sebagai berikut:

2.4.1 *Notepad++*

Notepad++ merupakan salah satu penyunting teks dan penyunting kode sumber dan pengganti notepad yang tersedia dalam beberapa bahasa. Berbeda dengan *notepad*, *Notepad++* mempunyai kelebihan yaitu terdapat nomor pada setiap barisnya, hal tersebut akan memudahkan kita dalam mengedit kode-kode program, (Dadan dan Kerendi, 2013). Saat ini versi terbaru dari *Notepad++* adalah v7.6.6 (8 April 2019) dan dapat diunduh pada situs notepad-plus-plus.org.

2.4.2 *XAMPP*

XAMPP adalah salah satu aplikasi *web server apache* yang terintegrasi dengan *MySQL* dan *phpMyAdmin*. *XAMPP* dapat diinstal pada sistem operasi *Windows*, *Linux* dan *MacOS* (Dadan dan Kerendi, 2015). Saat ini versi terbaru dari *XAMPP* adalah versi v7.3.3 (8 April 2019) dan dapat diunduh pada situs apachefriends.org dengan catatan untuk versi v7.3.3 tidak lagi mendukung untuk sistem operasi *Windows XP*. Berikut beberapa komponen - komponen yang terdapat pada *XAMPP*, yaitu *Apache* (*web server*), *MySQL* (*database server*), *Filezila* *FTP server*, *Mercury Mail* (*mail server*), *phpMyAdmin* (*web-based interface MySQL*).

2.4.3 *Website*

Website merupakan sebuah halaman berisi informasi yang dapat dilihat jika komputer anda terkoneksi dengan internet. Dengan adanya *website*, semua orang di dunia dapat mendapatkan dan mengelola informasi dengan berbagai sumber yang ada di internet. *Website* sendiri saat ini dapat memuat berbagai media mulai dari teks, gambar, suara, dan video.

Menurut Yuhefizar dan tim (2009) ada beberapa jenis *website*, yaitu berdasarkan sifatnya, tujuannya dan bahasa pemrogramannya. Berdasarkan sifatnya, *website* terbagi atas:

1. *Website Statis*

Website statis merupakan *website* yang kontennya sangat jarang diubah. Misalnya *website* profil organisasi dan lain-lain.

2. *Website Dinamis*

Website dinamis merupakan *website* yang menyediakan konten atau isi yang selalu berubah-ubah setiap saat. Misalnya *website* berita seperti www.detik.com.

Berdasarkan tujuannya, *website* terbagi atas:

- 1 *Personal website*, *website* yang berisi informasi pribadi seseorang.
- 2 *Corporate website*, *website* yang dimiliki oleh sebuah perusahaan.
- 3 *Forum website*, *website* yang mempunyai banyak layanan, mulai dari layanan berita, email dan jasa-jasa lainnya.
- 4 Di samping itu ada juga *website e-Government*, *e-Banking*, *e-Payment*, *e-Procurement* dan sebagainya.

Ditinjau dari bahasa pemrograman yang digunakan, *website* terbagi atas:

1. *Server side*, merupakan *website* yang menggunakan bahasa pemrograman yang tergantung kepada tersedianya *server*. Seperti PHP, ASP dan lain sebagainya. Jika tidak ada *server*, *website* yang dibangun tidak dapat berfungsi sebagaimana mestinya.
2. *Client side*, adalah *website* yang tidak membutuhkan *server* dalam menjalankan.

2.4.4 *Webserver*

Menurut Yudhanto dan Prasetyo (2018), ada dua pengertian *webserver* berdasarkan jenis perangkatnya, yaitu sebagai bagian dari perangkat keras (hardware) maupun sebagai perangkat lunak (software). Jika merujuk pada perangkat keras, *webserver* digunakan untuk menyimpan semua data seperti HTML, dokumen, gambar, *file* CSS dan *file* JavaScript. Sedangkan pada sisi perangkat lunak, *webserver* adalah sebagai pusat kontrol untuk memproses permintaan yang diterima dari *browser*. Jadi, sebenarnya semua yang berhubungan dengan *website* biasanya juga berhubungan dengan *webserver* karena tugas *webserver* adalah mengatur semua komunikasi yang terjadi antara *browser* dengan *server* untuk memproses data.

2.4.5 *Apache Webserver*

Saat ini banyak *webserver* yang tersedia, sebagai contoh *apache*, *nginx*, IIS dan lainnya. *Apache* adalah salah satu *webserver* yang populer dan banyak dipakai, *Apache* atau yang terkadang disebut dengan *Apache Tomcat* memiliki dukungan dari pengembang *open source* dan mempunyai komunitas yang mampu menjadikannya raja dari *webserver* lain, (Yudhanto dan Prasetyo, 2018). Ada beberapa keunggulan dari *apache*, yaitu pengaturan lebih mudah, *open source*, komunitas yang besar dan gratis.

2.4.6 **DBMS (Database Management System)**

Hasil penelitian yang berjudul “SISTEM INFORMASI PENGOLAHAN DATA BARANG UNTUK SERTIFIKASI ISO 9001:2008 PADA PT.TUNAS BARU LAMPUNG. TBK” yang dilakukan oleh Nurjoko, menyebutkan bahwa menurut wahana komputer (2010), DBMS didefinisikan sebagai sebuah piranti yang memberikan akses menuju ke basis data. Sedangkan menurut Ramakrishnan dan Gehrke (2003), DBMS adalah perangkat lunak yang mendesain untuk membantu memelihara dan memanfaatkan kumpulan data yang besar, memberikan fasilitas untuk melakukan fungsi pengaturan, pengawasan, pengendalian, pengolahan dan

koordinasi terhadap semua proses yang terjadi pada sistem basis data. Menurut Subandi dan Syahidi (2018), fungsi dari DBMS adalah sebagai berikut:

1. Mendefinisikan, maksudnya adalah basis data melibatkan penentuan tipe data, struktur dan batasan dari data yang akan disimpan dalam basis data.
2. Membangun, maksudnya adalah proses penyimpanan data pada beberapa media penyimpanan yang dikendalikan oleh DBMS.
3. Memanipulasi, maksudnya adalah mencakup berbagai fungsi seperti *query* database untuk mengambil data tertentu/data yang dicari, termasuk operasi *insert*, *update* dan *delete* serta menghasilkan report data.
4. Proteksi, maksudnya adalah mencakup perlindungan terhadap sistem yang menangani kondisi baik pada perangkat keras ataupun perangkat lunak, dan perlindungan pengamanan yang menangani pengaksesan oleh pengguna yang tidak memiliki hak akses terhadap basis data.
5. Perawatan, maksudnya adalah DBMS harus memiliki kemampuan dalam memberikan perawatan pada sistem basis data terhadap segala perubahan yang terjadi pada data dalam basis data kapan pun diperlukan.

Komponen utama DBMS menurut Subandi dan Syahidi (2018) adalah sebagai berikut:

1. *Query Language*, digunakan oleh bagian lain dengan sedikit perintah sederhana.
2. *Report Generator*, dirancang untuk membuat cetakan, yang memiliki perintah-perintah untuk membuat header, judul, kolom, summary dan lain-lain.
3. DML (*Data Manipulation Language*), terdiri dari perintah-perintah yang disediakan dalam aplikasi untuk melakukan manipulasi data seperti *append*, *list* dan *update*.

4. *DDL (Data Definition Language)*, dengan bahasa ini kita dapat membuat tabel baru, membuat indeks, mengubah tabel, menentukan struktur dan lain-lain.
5. *Recovery*, merupakan kemampuan untuk mengembalikan data yang rusak atau hilang akibat operasi basis data.
6. *Data Dictionary*, digunakan untuk memelihara definisi-definisi standar seluruh rinci data dalam lingkup kecil pada sistem basis data.
7. *Database*, merupakan bagian dari DBMS yang menyediakan dalam berbagai tipe dan format untuk memenuhi kebutuhan pemakai.
8. *Access Routine*, merupakan suatu rutin yang dapat dipanggil dan dipergunakan oleh program lain untuk mengakses basis data.

2.4.7 *PhpMyAdmin*

PhpMyAdmin adalah aplikasi *web* untuk mengelola basis data *MySQL* dan basis data *MariaDB* dengan mudah melalui antarmuka grafis (Yudhanto dan Prasetyo, 2018). *PhpMyAdmin* yang dianggap sangat penting dan juga sifatnya yang terbuka menjadikannya salah satu aplikasi yang selalu ada di dalam *cPanel* (aplikasi populer untuk pengendali *website*) dan sebagai bukti bahwa penyedia *web hosting* menaruh kepercayaan yang sangat besar pada *phpMyAdmin* sebagai salah satu aplikasi *web* yang dipasang di *server*.

2.4.8 HTML

HTML adalah *Hypertext Markup Language* yang artinya adalah sebuah teks berbentuk *link* dan mungkin juga foto atau gambar yang saat di klik akan membawa si pengakses internet dari satu dokumen ke dokumen yang lainnya, (Jubilee, 2016). Untuk versi saat ini adalah HTML 5 (8 April 2019).

2.4.9 PHP

PHP singkatan dari PHP: *Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat open source. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server* (Anhar). Untuk saat ini versi terbaru adalah PHP 7 (8 April 2019).

2.4.10 CSS

CSS adalah singkatan dari *Cascading Style Sheet*. Pada dasarnya CSS digunakan untuk mengubah tampilan yang dihasilkan oleh tag HTML. Karena bekerja dengan kode yang berbeda, maka CSS membutuhkan suatu mekanisme untuk “menempelkan” kode-kode CSS tersebut dengan tag HTML yang baik (Jubilee, 2016). Ada tiga metode penulisan CSS, yaitu *Inline Style Sheet*, *Embedded Style Sheet* dan *Linked Style Sheet* (Juju, 2007). *Inline Style Sheet* adalah penulisan sintaks CSS di dalam elemen HTML, untuk metode ini jarang digunakan karena dinilai tidak efisien, penulisan sintaks diletakkan pada tag HTML yang ingin diubah saja. Selanjutnya adalah *Embedded Style Sheet* yang hampir sama dengan metode sebelumnya, perbedaannya terdapat pada penulisan sintaks yang diletakkan pada bagian tag <head> dan dapat dipakai pada semua tag HTML selama berada dalam halaman HTML yang sama. Yang terakhir adalah *Linked Style Sheet*, berbeda dengan sebelumnya, penulisan sintaks CSS di letakkan di halaman yang berbeda atau terpisah dari HTML, jadi untuk pemakaian CSS pengguna cukup melakukan *link* ke *file* CSS yang telah dibuat.

2.4.11 JavaScript

JavaScript adalah bahasa *scripting* yang populer di internet dan dapat bekerja di sebagian browser populer seperti *Mozilla Firefox* dan *Google Chrome*. Kode *JavaScript* dapat disisipkan dalam halaman *web* menggunakan tag SCRIPT (Sunyoto, 2007).

Beberapa hal tentang *JavaScript* menurut Sunyoto (2007) adalah:

1. *JavaScript* didesain untuk menambah interaktif suatu *web*.
2. *JavaScript* merupakan sebuah bahasa *scripting* dan merupakan bahasa pemrograman yang ringan.
3. *JavaScript* berisi baris kode yang dijalankan di komputer dan biasanya disisipkan dalam halaman HTML.
4. *JavaScript* adalah bahasa interpreter yang berarti skrip dieksekusi tanpa proses kompilasi.

2.4.12 Bootstrap

Bootstrap adalah *framework front-end* yang intuitif dan *powerfull* untuk pengembangan aplikasi *web* yang lebih cepat dan mudah. *Bootstrap* menggunakan HTML, CSS, dan *JavaScript* (Jubilee, 2016). Kelebihan *bootstrap* menurut Jubilee (2016) adalah sebagai menghemat waktu, fitur yang responsive, desain yang konsisten, mudah digunakan, didukung oleh semua *browser* populer dan gratis. Menurut Rozi dan tim (2015), daya tarik *bootstrap* terletak pada kemudahan dan kepraktisan penggunaannya, tata *layout* dan pewarnaannya juga bersih, simpel, indah, dan berkesan modern. Selain itu mengingat akses *website* via piranti *mobile* semakin meningkat tajam, maka pendekatan desain *website* responsive dan *mobile-friendly* yang ditawarkan *bootstrap* juga menjadi solusi praktis dan murah bagi pengembang *website*. Cukup dengan *bootstrap*, *website* dapat diakses secara nyaman via ponsel tanpa menggunakan aplikasi khusus untuk piranti *mobile*.

Terinspirasi dari hasil penelitian yang diberi nama aplikasi pengolahan data hasil penelitian pustekbang lapan yang dibuat oleh penulis dalam penelitian sebelumnya, template *bootstrap* yang digunakan dalam aplikasi tersebut digunakan kembali dalam penelitian kali ini.