

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian tentang Surat Perintah Perjalanan Dinas pernah dilakukan oleh Nindya Eka Puspitasari dan Putri Aisyiyah Rakhma Devi. Dengan judul “Rancang Bangun Sistem Informasi Surat Perintah Perjalanan Dinas Di Dinas Koperasi dan Usaha Mikro Lamongan”[4]. Pada penelitian ini dibahas tentang proses pembuatan surat perjalanan dinas yang masih manual dengan menggunakan aplikasi microsoft word dimana dalam pembuatannya harus dilakukan pengeditan setiap kali akan melakukan perjalanan dinas sehingga proses pembuatan surat tersebut cenderung membutuhkan waktu yang lama. Selain itu dalam mencari data setiap pegawai yang akan melaksanakan perjalanan dinas masih menggunakan cara yang manual yaitu masih menggunakan microsoft excel. Penelitian ini bertujuan supaya dapat mempermudah kinerja pegawai dalam pengelolaan atau perencanaan perjalanan dinas serta membantu pegawai mempermudah dalam membuat surat dinas pegawai, pengolah biaya, administrasi biaya, pencetak surat dinas dan hasil tugas dinas sehingga nantinya mempermudah bagian administrator dalam penginputan pegawai yang akan melakukan perjalanan dinas. Pada penelitian ini menggunakan metode *waterfall*, model ini mengusulkan sebuah pendekatan perkembangan perangkat lunak yang sistematis dan sekuensial yang dimulai pada tahap awal pengembangan sistem yaitu tahap perencanaan sampai akhir pengembangan sistem tahap pemeliharaan.

Penelitian serupa juga pernah dilakukan oleh Heriyanto Syahidu, Idawati Masulili. Dengan judul “Sistem Informasi Perjalanan Dinas Berbasis Web di DP3AP2KB Kabupaten Banggai Kepulauan”[5]. Pada penelitian ini dibahas tentang bagaimana proses pembuatan surat perintah perjalanan dinas masih manual yaitu menggunakan aplikasi Microsoft Word dimana harus dilakukan pengeditan setiap kali membuat surat perintah perjalanan dinas. Sehingga penerapan cara manual ini memiliki kelemahan yaitu kurang efisien, butuh waktu yang lama karena sering dilakukan pengeditan dalam pembuatannya. Metode yang digunakan dalam pengembangan sistem ini adalah metode kualitatif. Sumber data

kualitatif diperoleh melalui obeservasi dan wawancara kepada pihak DP3AP2KB.

Penelitian sejenis juga pernah dilakukan oleh Sajjah, Haris Tehuayo, Sulistianingsih A.S. Dengan judul “Sistem Informasi Surat Perintah Perjalanan Dinas pada Kantor BPS (Badan Pusat Statistik) Kabupaten Maros”[6]. Metodologi pengembangan sistem yang digunakan dalam penelitian ini adalah metode *waterfall*. Sistem ini bertujuan untuk mempercepat proses pembuatan surat perintah perjalanan dinas.

Pada penelitian ini, peneliti bermaksud untuk membuat sistem informasi pengajuan surat perintah perjalanan dinas Politeknik Negeri Cilacap yang digunakan untuk mempermudah proses pengajuan surat tersebut. Metode yang digunakan dalam penelitian ini adalah metode *waterfall* dan menggunakan Bahasa pemrograman PHP dengan framework CI. Dengan dilakukannya penelitian ini diharapkan petugas bagian umum dan bagian keuangan tidak lagi menemukan kendala dalam proses pengajuan surat perintah perjalanan dinas.

2.2 Landasan Teori

2.2.1 Pengertian Perjalanan Dinas

Perjalanan dinas adalah perjalanan yang dilakukan oleh seorang karyawan atau pegawai suatu lembaga atau perusahaan yang berkaitan dengan tugas pekerjaan kedinasan. Ada beberapa dokumen perjalanan dinas yang harus dipersiapkan oleh seorang karyawan sebelum perjalanan dinas, yaitu: Surat Tugas dan Surat Perintah Perjalanan Dinas (SPPD)[7].

2.2.2 Sistem

A. Pengertian Sistem

Sistem merupakan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan tertentu untuk dapat mencapai tujuan tertentu[8].

B. Pengertian Informasi

Informasi merupakan sebuah rangkaian data yang memiliki sifat sementara, tergantung dengan waktu, mampu memberikan kejutan pada penerimanya. Informasi yang tidak memiliki nilai, pada umumnya karena rangkaian data yang tidak lengkap atau kadaluarsa[9].

C. **Karakteristik Sistem**

Sistem memiliki karakteristik yang tidak dapat terpisahkan antara satu karakteristik dengan karakteristik lainnya. Berapa karakteristik tersebut antara lain yaitu[10]:

- a. **Komponen Sistem**
Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama. Komponen sistem dapat berupa subsistem maupun bagian-bagian dari sistem.
- b. **Batasan Sistem**
Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya.
- c. **Lingkungan Luar Sistem**
Lingkungan luar dari suatu adalah apapun di luar batas dari sistem yang dapat mempengaruhi operasi sistem.
- d. **Penghubung Sistem**
Masukan merupakan energi yang dimasukkan ke dalam sistem.
- e. **Keluaran Sistem**
Keluaran adalah hasil energi yang diolah serta diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.
- f. **Pengolah Sistem**
Sebuah sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.
- g. **Sasaran Sistem**
Suatu sistem tentunya memiliki tujuan atau sasaran. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan di hasilkan oleh sistem.

2.2.3 **Perangkat Lunak Pendukung**

Membangun sebuah sistem informasi diperlukan adanya perangkat lunak pendukung. Perangkat lunak yang digunakan untuk membangun sistem ini meliputi PHP, MySQL, HTML, XAMPP, CSS, Framework CI.

- a. **PHP**
PHP adalah bahasa pelengkap HTML yang memungkinkan dibuatnya aplikasi dinamis yang memungkinkan adanya pengolahan data dan pemrosesan data. Seluruh *sintax* yang di berikan akan sepenuhnya

dijalankan pada *server* sedangkan yang dikirimkan ke *browser* hanya hasilnya saja[11].

b. MySQL

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirim datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar *SQL (Structure Querred Lenguage*[12].

c. HTML

HTML adalah singkatan dari *Hypertext Markup Language* merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman web tersebut[13].

d. XAMPP

XAMPP adalah paket PHP dan MySQL berbasis *open source* yang mampu membuat web dinamis. XAMPP merupakan aplikasi web *server* yang mendukung instalansi Linux dan Windows, selain itu XAMPP juga termasuk dalam aplikasi *open source*. XAMPP terdiri dari beberapa paket aplikasi *server* seperti Apache Http *Server*, MySQL *Database Server*, Filezilla FTP *Server* XAMPP juga mudah untuk di download diberbagai situs jaringan[14].

e. CSS

CSS atau (*Cascading Style Sheet*) biasanya digunakan untuk mengatur tampilan elemen yang tertulis dalam bahasa *markup*, seperti HTML. CSS berfungsi untuk memisahkan konten dari tampilan visualnya di situs. HTML dan CSS memiliki keterikatan yang erat. Karena HTML adalah bahasa *markup* dan CSS memperbaiki *style*, maka kedua bahasa pemrograman ini harus berjalan beriringan[15].

f. Framework CI

CodeIgniter merupakan framework PHP yang dibuat berdasarkan model *view Controlleer (MVC)*. CI memiliki *library* yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh aplikasi berbasis web misalnya mengakses *database*, memvalidasi form sehingga sistem yang dikembangkan mudah. CI menjadi satu-satunya framework dengan dokumentasi yang lengkap dan jelas. *Source code* CI yang dilengkapi dengan comment didalamnya sehingga lebih memperjelas fungsi sebuah kode program dan CI yang dihasilkan sangat Bersih (*clean*) dan *search Engine Friendly (SEF)*[16].

2.2.4 Rekayasa Web

Rekayasa web adalah sebuah aplikasi yang menggunakan pendekatan sistematis, disiplin, dan terukur untuk pengembangan, operasi dan pemeliharaan aplikasi berbasis *Website*.

Rekayasa web adalah subdisiplin dari rekayasa perangkat lunak yang membantu menyediakan metodologi untuk merancang, mengembangkan, memelihara, dan melibatkan aplikasi Web [17].

Tahapan-tahapan dalam rekayasa web antara lain[18]

1. *Customer communication*

Komunikasi yang baik dengan user merupakan sarana efektif dalam membuat maupun menerjemahkan apa saja yang *user* inginkan (*requirements*).

2. *Planning*

Planning merupakan tahap penggabungan *requirements* (kebutuhan) dan informasi dari user bisa berupa wawancara dan perencanaan teknis. Perencanaan teknis dilakukan dengan cara mengidentifikasi perangkat lunak maupun perangkat keras apa saja yang dibutuhkan.

3. *Modeling*

Analysis modeling, adalah tahap berikutnya dari *planning* dan komunikasi dengan *user*. Yakni merumuskan kebutuhan (*requirements*) dari *user* serta permasalahan apa saja yang akan diselesaikan. Mengidentifikasi interaksi antara user dengan sistem berdasarkan hak akses pengguna, serta analisis fungsional apa saja yang akan dijalankan pada sistem.

4. *Construction*

Implementasi dilakukan dengan menggunakan aplikasi halaman web dalam bentuk HTML dengan berdasarkan hasil perancangan isi pada aktivitas pada *non technical member* sedangkan untuk implementasi isi dan fungsi logika dibuat dalam bentuk PHP.

5. *Testing*

Testing dilakukan untuk dapat mengetahui kemungkinan terjadinya kesalahan seperti kesalahan pada skrip atau form, navigasi ataupun tampilan, maupun bagian lainnya.

6. *Deployment*

Deployment merupakan serah terima dan respon dilakukan dengan cara menyebarkan suatu kuisioner kepada *user* berupa respon untuk mendapatkan penilaian dari setiap kriteria sebagai hasil evaluasi bagi

pengembang. Kriteria yang dilakukan pada umumnya *usability*, *functionality*, dan *reliability*.

2.2.5 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah sebuah proses kegiatan perangkat lunak itu sendiri yang berguna untuk mengembangkan, memelihara, dan membangun kembali dengan menggunakan prinsip rekayasa perangkat lunak yang dapat bekerja lebih efisien dan efektif untuk pengguna[19].

Rekayasa perangkat lunak mempunyai beberapa pokok bahasa penting yaitu:

- a. *Domain Engineering*
Kemampuan untuk dapat memahami suatu permasalahan yang muncul dan dijadikan suatu proyek perangkat lunak.
- b. *Requirement Engineering*
Kemampuan untuk dapat memahami kebutuhan pengguna sekaligus dalam memecahkan sebuah masalah.
- c. *Software Design*
Kemampuan untuk dapat memahami serta mengimplementasikan dalam perancangan perangkat lunak yang di dalamnya terdapat aspek HCI (*Human Computer Interaction*).
- d. *Development*
Proses pengembangan yang melibatkan pembelajaran tentang algoritma, bahasa pemrograman yang di terapkan dan teknik yang didalamnya berkaitan dengan basis data dan sistem informasi.
- e. *Maintenance*
Pemeliharaan dan pengembangan suatu sistem yang berguna untuk banyak orang.

1. Metode Pengembangan Sistem

Metode yang digunakan yaitu metode *waterfall*. Metode air terjun atau yang sering disebut metode *waterfall* seing dinamakan siklus hidup klasik (*classic life cycle*), nama model ini sebenarnya adalah "*Linear Sequential Model*" dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modelling*), konstruksi (*contruction*), serta penyerahan sistem ke para pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan[20].

2. Metode Pengujian Sistem

Pengujian merupakan suatu proses pelaksanaan program yang bertujuan untuk menemukan kesalahan dan memperbaikinya sehingga sistem dikatakan layak untuk digunakan[21]. Pengujian pada sistem menggunakan metode *Black box*.

a. *Black Box Testing*

Metode *Black Box Testing* adalah metode pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, *tester* dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. Proses *Black Box Testing* dengan cara mencoba program yang telah dibuat dengan mencoba memasukan data pada setiap prosesnya[22].

Black Box Testing berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. *Black Box Testing* bukanlah solusi alternatif dari *White Box Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing*. *Black Box Testing* cenderung untuk menemukan hal-hal berikut[3]:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka.
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi.
5. Kesalahan inisialisasi dan terminasi.

2.2.6 Pemrograman Berorientasi Objek

Pemrograman Berorientasi Objek adalah pemrograman yang berorientasikan kepada objek, dimana seluruh data dan fungsi dibungkus dalam *class-class* atau *object-object*. Setiap *object* bisa menerima pesan, memproses data, mengirim, menyimpan serta memanipulasi data. Berapa *object* berinteraksi dengan saling memberikan suatu informasi satu terhadap yang lainnya. Dalam konsep Pemrograman Berorientasi Objek dikenal beberapa istilah umum yakni[23]:

1. *Attribute*

Atribut dari sebuah kelas merupakan variabel global yang dimiliki sebuah kelas, Atribut bisa mempunyai hak akses *private*, *public* maupun *protected*.

2. *Method*

Method merupakan fungsi atau prosedur yang dibuat oleh seorang programmer didalam suatu *Class*. Dengan kata lain, *method* pada sebuah kelas hampir sama dengan fungsi maupun prosedur pada pemrograman prosedural.

3. *Class*

Class didefinisikan sebagai suatu *blueprint* (denah), atau prototipe, yang mendefinisikan variabel-variabel dan metode-metode yang umum untuk semua objek dari jenis tertentu. Sebuah kelas menyerupai sebuah struktur yang merupakan tipe data sendiri, contoh misalnya tipe data titik yang terdiri dari koordinat x dan y.

4. *Object*

Object adalah perwujudan dari class, setiap object akan memiliki *attribute* dan *method* yang dimiliki oleh classnya.

5. *Abstraction*

Abstraksi adalah kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yakni kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari “pelaku” abstrak yang bisa melakukan kerja, laporan dan perubahan keadaannya, serta berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan.

6. *Encapsulation*

Encapsulation adalah suatu mekanisme untuk menyembunyikan atau memproteksi suatu proses dari kemungkinan interferensi atau penyalahgunaan dari luar sistem dan sekaligus untuk menyederhanakan penggunaan sistem tersebut.

7. *Inheritance*

Inheritance adalah konsep mewariskan *attribute* dan *method* yang dimiliki oleh sebuah *class* kepada *class* turunannya. Dengan konsep tersebut *class* yang dibuat cukup mendefinisikan *attribute* dan *method* yang spesifik didalamnya, sedangkan untuk *attribute* dan *method* yang lebih umum akan didapatkan dari *class* yang menjadi induknya.

8. *Polymorphism*



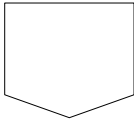
Polymorphism adalah konsep yang memungkinkan digunakannya suatu *interface* yang sama untuk memerintah suatu *object* agar melakukan suatu tindakan yang mungkin secara prinsip sama akan tetapi secara proses berbeda.

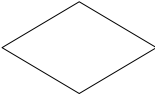

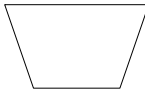
Di dalam pemrograman berorientasi objek, UML digunakan untuk pemodelan sistem. UML adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem.

A. *Flowchart*

Flowchart adalah suatu jenis diagram yang mempresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem. *Flowchart* sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang akan dibangun kepada *programer*. Dengan demikian, *flowchart* dapat membantu untuk memberikan solusi terhadap masalah yang bisa saja terjadi dalam membangun sistem[24]. Berikut merupakan simbol-simbol *flowchart* yang dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Simbol *flowchart*


No.	Simbol	Nama	Keterangan
1.		<i>Terminal</i>	Memulai dan mengakhiri suatu program
2.		<i>Input/Output</i>	Memasukan data maupun menunjukkan hasil dari suatu <i>process</i> tanpa tergantung dengan jenis peralatannya
3.		<i>Off-Page Connector Symbol</i>	Simbol penghubung yang digunakan untuk keluar-masuk atau penyambungan proses dalam halaman atau lembar yang berbeda.






No.	Simbol	Nama	Keterangan
4.		<i>Decision</i>	Memilih proses berdasarkan kondisi yang ada.
5.		<i>Document</i>	Merupakan simbol untuk data yang terbentuk informasi.
6.		<i>Manual peration</i>	Menunjukkan pengolahan yang tidak dilakukan oleh komputer/pc.

B. Use Case Diagram

Use case adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor”[25]. *Use case diagram* merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* bekerja dengan mendeskripsikan tipikal interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem itu dipakai[26]. Berikut merupakan simbol-simbol use case diagram yang dapat dilihat pada Tabel 2.2.

Tabel 2. 2 *Use case* diagram



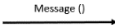



No.	Simbol	Nama	Keterangan
1.		<i>Use case</i>	Deskripsi dari urutan aksi – aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .

2.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
3.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
F		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5.		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas

C. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antara sejumlah objek dalam urutan waktu"[25]. *Sequence* diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Gambaran *sequence* diagram dibuat minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada *sequence* diagram sehingga semakin banyak *use case* yang didefinisikan, maka *sequence* diagram yang harus dibuat juga semakin banyak[26]. Berikut merupakan simbol-simbol *sequence* diagram yang dapat dilihat pada Tabel 2.3.





Tabel 2. 3 *Sequence* diagram

No.	Simbol	Nama	Keterangan
1.		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2.		<i>Actor</i>	Menggambarkan <i>user</i> atau pengguna.
3.		<i>Message</i>	Spesifikasi dari komunikasi antar <i>objek</i> yang memuat informasi – informasi tentang aktivitas yang terjadi.
4.		<i>Boundary</i>	Menggambarkan sebuah <i>form</i> .
5.		<i>Control Class</i>	Menghubungkan <i>boundary</i> dengan Tabel.
6.		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan.

D. ERD (*Entity Relationship Data*)

Entity Relationship Data (ERD) merupakan diagram yang berbentuk notasi grafis, yang berada dalam pembuatan *database* yang menghubungkan antara data satu dengan yang lain. Fungsi dari ERD adalah sebagai alat bantu dalam pembuatan *database* dan memberikan gambaran bagaimana kerja *database* yang akan dibuat. ERD memiliki 3 elemen dasar yaitu entitas, atribut, dan relasi[28]. Berikut merupakan simbol-simbol ERD yang dapat dilihat pada tabel 2.4:

Tabel 2. 4 Tabel ERD

No.	Nama	Simbol	Keterangan Fungsi
1.	Entitas		Persegi panjang menyatakan himpunan entitas adalah orang, kejadian, atau berada.
2.	Atribut		Atribut merupakan informasi yang diambil tentang sebuah entitas.
3.	Relasi		Belah ketupat menyatakan himpunan relasi merupakan hubungan antar entitas.
4.	<i>Associative Entity</i>		<i>Entitas</i> yang digunakan pada relasi <i>many-to-many</i> .

2.2.7 Basis Data

Basis data adalah suatu kumpulan koleksi data-data yang saling berhubungan secara logika yang isinya di desain untuk memenuhi kebutuhan informasi dari suatu perusahaan[27].

Pengoperasian Arsitektur Basis Data dibangun dengan menggunakan format paket bahasa yaitu DDL, dan DML.

1. DDL (*Data Definition Language*)

Merupakan satu paket bahasa DBMS yang berguna untuk melakukan spesifikasi terhadap skema basis data. Terdapat tiga perintah penting di dalam DDL, yaitu:

- a) *CREATE* adalah sebuah perintah yang digunakan untuk membuat, termasuk di dalamnya membuat *database* baru, tabel

baru, *view* baru, dan kolom baru. Contoh: *CREATE DATABASE* nama *database*.

- b) *ALTER* adalah perintah untuk mengubah struktur tabel yang telah dibuat. Mencakup didalamnya yaitu mengubah nama tabel, menambah kolom, mengubah kolom, menghapus kolom, dan memberikan atribut pada kolom. Contoh: *Alter table* nama *table ADD* nama kolom *datatype*.
- c) *DROP* adalah perintah untuk menghapus *database* atau tabel. Contoh: *DROP DATABASE* nama *database*.

2. DML (*Data Manipulation Language*)

Merupakan satu paket DBMS yang memperbolehkan pemakai untuk mengakses atau memanipulasi data sebagaimana yang telah diorganisasikan sebelumnya dalam model data yang tepat. Terdapat empat perintah penting dalam DML, yaitu:

- a) *INSERT* adalah perintah untuk memasukan data baru ke dalam sebuah tabel. Perintah ini tentu saja bisa dijalankan pada saat *database* dan tabel sudah dibuat. Contoh: *INSERT INTO* nama tabel *VALUES* (data1, data2, dst...).
- b) *SELECT* adalah perintah untuk mengambil dan menampilkan data dari satu tabel atau beberapa tabel dengan penggunaan relasi. Contoh: *SELECT* nama_kolom1, nama_kolom2 *FROM* nama tabel.
- c) *UPDATE* adalah perintah untuk memperbaharui data pada sebuah tabel. Contoh: *UPDATE* nama_tabel kolom2=data1, kolom2=data2, ... *WHERE* kolom=data.
- d) *DELETE* adalah perintah untuk menghapus data dari sebuah tabel. Contoh: *DELETE FROM* nama tabel *WHERE* kolom=data.

A. *Database Management System (DBMS)*

DBMS adalah sistem perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, memelihara, dan mengendalikan akses ke basis data.

Pengertian lain dari DBMS adalah program komputer yang digunakan untuk membuat, memproses komputer yang digunakan untuk membuat, memproses dan mengelola basis data.

Sehingga kesimpulan dari pengertian DBMS di atas adalah suatu sistem atau perangkat lunak yang dirancang untuk mengelola suatu basis

data dan menjalankan operasi terhadap data yang diminta banyak pengguna[28].

B. MySQL

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirim datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar *SQL (Structure Quered Lenguage*[12].

MySQL termasuk jenis RDBMS (*Relational Database Manajement System*). MySQL mendukung bahasa pemrograman PHP, bahasa permintaan yang terstruktur, karena pada penggunaannya SQL memiliki beberapa aturan yang telah distandarkan oleh asosiasi yang bernama ANSI.

MySQL merupakan RDBMS (*Relational Database Management System*) *server*. RDBMS adalah program yang memungkinkan pengguna *database* untuk membuat, mengelola, dan menggunakan data pada suatu model *relational*. Dengan demikian, tabel-tabel yang ada pada *database* memiliki relasi antara satu tabel dengan tabel lainnya[11].