

BAB II

DASAR TEORI

2.1. Tinjauan Pustaka

Penelitian pertama oleh Sandika Adi Nugraha dan Ridwan Setiawan dengan judul “Perancangan Sistem Informasi Pengolahan Administrasi Keuangan Sekolah Menengah Pertama Islam Terpadu Assalam Garut”. Sistem informasi tersebut dirancang bertujuan untuk menghasilkan perancangan sistem informasi administrasi keuangan Sekolah Menengah Pertama Islam Terpadu Assalam Garut. Sebelum adanya sistem ini, proses pengolahan data administrasi di SMP IT Assalam Garut masih menggunakan sistem manual dan data-data tersebut dicatat secara manual di dalam suatu buku induk. Kondisi ini membuat staff Tata Usaha di SMP IT Assalam Garut kesulitan dalam mengolah data administrasi keuangan. Penelitian ini menggunakan metode Object Oriented Programming dengan pendekatan USDP (*Unified Software Development Process*) [3].

Penelitian kedua oleh Ray Septy Fajar Muhammad mengenai “Sistem Pembayaran SPP di SD Muhammadiyah16 Surakarta Berbasis Aplikasi Desktop”. Sistem ini dirancang untuk membantu guru dalam hal-hal yang berkaitan dengan proses pembayaran SPP dan laporan pembayaran SPP yang lebih efektif dan efisien dengan menggunakan aplikasi desktop. Proses administrasi yang sifatnya manual membutuhkan waktu yang lama sehingga menyebabkan manajemen keuangan tidak optimal. Penelitian ini menggunakan metode SDLC (*System Development Life Cycle*) model Waterfall dengan perangkat lunak Visual Basic (VB6) [4].

Penelitian ketiga yang dilakukan oleh Wahyu Eko Susanto mengenai “Aplikasi Sistem Informasi Administrasi Keuangan Masjid Menggunakan Metode Incremental”. Sistem ini dibangun untuk memudahkan takmir masjid dalam mengelola administrasi masjid dikarenakan pengerjaannya masih secara manual dan juga kurangnya pengetahuan mengenai pengadministrasian bagi para takmir. Aplikasi

sistem informasi ini dapat membantu dalam pengelolaan administrasi masjid yang bisa diakses publik, serta menjadi organisasi yang profesional dan mampu membantu pemerintah untuk membentuk SDM yang unggul [5].

Penelitian keempat oleh Weiskhy Steven Dharmawan, Deasy Purwaningtias, dan Deni Risdiansyah mengenai “Penerapan Metode SDLC Waterfall Dalam Perancangan Sistem Informasi Administrasi Keuangan Berbasis Desktop”. Sistem informasi tersebut dirancang untuk mempermudah dan mengolah data keuangan, serta mempermudah pembuatan laporan kas masuk dan kas keluar untuk menghasilkan laporan yang jauh lebih baik. Sebelum adanya sistem ini, diketahui bahwa sistem pencatatan media kertas merupakan alat yang digunakan untuk mencatat setiap data pada kantor tersebut. Akibatnya laporan yang dihasilkan sering terjadi kesalahan karena hilang dan rusaknya bukti transaksi. Rancangan yang dibuat menggunakan bahasa pemrograman Visual Studio 2010 dan menerapkan metode SDLC *Waterfall* sebagai metode pengembangan perangkat lunak [6].

Penelitian terakhir oleh Fildzah Fitriani dan Safitri Juanita dengan judul “Perancangan Sistem Informasi Akademik Berbasis Desktop di SMK Muhammadiyah 4 Jakarta”. Sistem ini dibangun untuk membantu pihak sekolah dalam pengolahan data akademik dengan baik agar layanan yang diberikan kepada guru dan staff Tata Usaha dapat berjalan dengan baik pada SMK dikarenakan proses pengolahan akademik siswa di SMK Muhammadiyah 4 Jakarta masih dilakukan secara manual dan belum terkomputerisasi sehingga menyebabkan beberapa permasalahan seperti nilai sering tertukar disebabkan staf tata usaha kurang teliti, pengolahan data penilaian siswa membutuhkan waktu yang lama dikarenakan perhitungan nilai dilakukan secara konvensional, informasi data siswa pada buku rapor kurang lengkap dikarenakan data orang tua dan data nilai sikap tidak diisi, informasi kelas pada daftar absensi kurang lengkap dikarenakan data kelas siswa tidak diisi, pengolahan data jumlah kehadiran siswa membutuhkan waktu yang lama dikarenakan dihitung secara manual dan pengolahan data siswa dengan peringkat terbaik membutuhkan waktu yang lama

dikarenakan proses perangkan peringat siswa dilakukan secara manual [7].

Setelah mengevaluasi penelitian sebelumnya, penulis akan membuat sistem yang berjudul “Aplikasi Administrasi Keuangan SPP Menggunakan Notifikasi SMS Gateway Berbasis *Website*” yang bertujuan untuk membantu staff tata usaha dan bendahara sekolah dalam proses pencatatan dan pengelolaan data administrasi SPP di SMP Purnama Kesugihan, membantu siswa dalam mendapatkan informasi mengenai pembayaran SPP & menjadi pertanda bahwa transaksi yang dilakukan sudah berhasil, serta membantu kepala sekolah dalam melihat laporan administrasi pembayaran SPP siswa. Sistem yang akan dibuat oleh penulis yaitu sistem dapat diakses oleh siapapun jika memiliki username dan password dan dapat diakses dimanapun serta terdapat notifikasi SMS sebagai sarana informasi kepada siswa. Perbedaan sistem yang dibangun penulis dengan penulis sebelumnya yaitu berbasis *website* menggunakan *framework CodeIgniter* dan menggunakan notifikasi SMS Gateway dengan metode pengembangan sistem menggunakan *waterfall*.

2.2. Landasan Teori

2.2.1. Aplikasi

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju [8].

2.2.2. Administrasi

Administrasi dapat diartikan sebagai cara ilmiah untuk mendapatkan data yang valid dengan tujuan dapat ditemukan, dibuktikan dan dikembangkan suatu pengetahuan tertentu sehingga pada gilirannya dapat digunakan memahami, memecahkan dan mengantisipasi masalah dalam bidang administrasi dan manajemen [9].

2.2.3. SPP (Sumbangan Pembinaan Pendidikan)

Sumbangan Pembinaan Pendidikan atau biasa disingkat dengan SPP merupakan iuran pembayaran dari siswa yang menjadi kewajiban bagi setiap siswa di sekolah. Pembayaran SPP diambil berdasarkan kesepakatan rapat komite sekolah dan orang tua siswa. Pembayaran SPP bertujuan untuk menunjang peningkatan mutu pendidikan yang terkait dengan sarana dan prasarana sekolah sehingga kegiatan belajar mengajar di sekolah dapat berjalan lancar dengan adanya bantuan dari dana iuran tersebut [10].

2.2.4. SMS Gateway

SMS merupakan singkatan dari *Short Message Service* yaitu komponen layanan komunikasi teks dari sistem komunikasi *mobile* yang menggunakan standar protocol komunikasi yang memungkinkan pertukaran pesan antara perangkat *mobile phone*. Layanan *SMS Gateway* menyediakan layanan dari *phone to computer* atau dari *computer to phone* [11].

SMS Gateway adalah suatu platform yang menyediakan mekanisme untuk mengirim dan menerima SMS. *SMS Gateway* dapat berkomunikasi dengan perangkat lain yang memiliki SMS platform untuk menghantar dan menerima pesan SMS dengan sangat mudah. Hal ini dimungkinkan karena *SMS Gateway* juga dibekali tampilan antarmuka yang mudah dan standar. Cara kerja *SMS Gateway* pada dasarnya sama dengan mengirim SMS melalui *handphone* pada umumnya. Jadi secara umum *SMS Gateway* adalah sebuah sistem aplikasi atau perangkat lunak yang digunakan untuk mengirim atau menerima SMS [12].

2.2.5. Website

Website merupakan sebuah sistem dengan informasi yang disajikan dalam bentuk teks, gambar, suara, dan lain – lain yang tersimpan dalam sebuah server Web Internet yang disajikan dalam bentuk *hypertext* [9]. Lebih lanjut dikatakan bahwa bentuk – bentuk *Website* dapat dijelaskan sebagai berikut :

- a. Web Server : *Apache Web Server* merupakan program aplikasi yang berjalan di server, berfungsi untuk menjalankan aplikasi web sehingga bisa diakses oleh klien baik melalui jaringan *intranet* maupun *internet*.
- b. Web Browser : Web Browser merupakan *software* yang diinstall di mesin client, berfungsi untuk menerjemahkan tag HTML menjadi halaman web.
- c. HTTP (*Hyper Text Tranfer Protocol*) : HTTP merupakan protokol komunikasi *stateless* yang berbasiskan pada TCP yang awalnya digunakan untuk mengambil kembali file – file HTML dari server *Web*.

2.2.6. Rekayasa Perangkat Lunak

Perangkat lunak adalah seluruh perintah yang digunakan untuk memproses informasi. Definisi Rekayasa Perangkat Lunak adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, desain, pengkodean, pengujian, sampai pemeliharaan sistem setelah dikembangkan [13].

Proses perangkat lunak sebagai sebuah kerangka kerja untuk tugas-tugas yang dibutuhkan untuk membangun perangkat lunak dengan kualitas yang baik. Proses perangkat lunak menentukan pendekatan yang digunakan ketika perangkat lunak yang dikembangkan, tetapi pengembangan perangkat lunak juga meliputi teknologi yang mempopulasikan metode-metode (*methods*), teknik, alat-alat bantu (*tools*) otomatis, dan prosedur-prosedur (*procedures*) atau sering disebut dengan proses.

1. Metode Pengembangan Sistem

Metode dapat diartikan sebagai menyediakan cara bagaimana secara teknis membangun perangkat lunak yang harus berada pada sebuah komitmen dasar menuju kualitas. Metode ini menyangkut serangkaian tugas yang luas yaitu:

1. Perencanaan proyek dan estimasi.
2. Analisis kebutuhan sistem dan software.

3. Rancangan struktur data, yang terdiri dari:
 - a. Variabel.
 - b. Elementary Data: Int, Char, String dll.
 - c. Struktur Data: *Record*, File, Array, String.
4. Arsitektur program.
5. Algoritma prosedur.
6. Pengkodean (*coding*).
7. Testing.
8. Pemeliharaan.

Metode pengembangan sistem yang digunakan adalah metode *waterfall*. Metode *waterfall* merupakan metode yang sering digunakan oleh penganalisa sistem pada umumnya. Metode *waterfall* adalah pengerjaan dari suatu sistem yang dilakukan secara berurutan, sebagai berikut :

- a. Analisis Kebutuhan

Langkah ini merupakan analisis terhadap kebutuhan sistem. Pengumpulan data pada tahap ini bisa dilakukan melalui sebuah penelitian, wawancara atau studi literatur. Seorang sistem analisa akan menggali informasi sebanyak-banyaknya dari *user* sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh *user* tersebut. Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan sistem analisa untuk menerjemahkan ke dalam bahasa pemrograman.

- b. Desain

Proses desain sistem yang menerjemahkan syarat kebutuhan sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada struktur data, arsitektur perangkat lunak, representasi *interface* dan detail (algoritma) prosedural. Tahapan ini menghasilkan dokumen yang disebut *requirement*. Dokumen inilah yang akan digunakan pemrograman untuk melakukan aktivitas pembuatan sistemnya.

c. Pengkodean

Coding merupakan penerjemah desain dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh *programmer* yang akan menerjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahap ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat tadi. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.

d. Pengujian

Tahapan ini dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, desain dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*.

e. Pemeliharaan

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan atau karena pelanggan membutuhkan perkembangan fungsional.

2. Metode Pengujian Sistem

Pada tahap pengujian, penulis menggunakan metode pengujian *blackbox*. Pengujian *blackbox* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *blackbox* memungkinkan perencana perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

- a. Fungsi - fungsi yang tidak benar atau hilang
- b. Kesalahan *interface* dan kesalahan kerja
- c. Kesalahan dalam struktur data atau akses database eksternal
- d. Inisialisasi dan kesalahan terminasi

2.2.7. UML (Unified Modeling Language)

UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek [14]. Penggunaan model ini bertujuan untuk mengidentifikasi bagian – bagian yang termasuk dalam lingkup sistem yang dibahas dan bagaimana hubungan antara sistem dengan sub sistem maupun sistem lain diluarnya. Beberapa artifak di dalam UML antara lain sebagai berikut :

a. *Flowchart*

Flowchart atau bagan alir adalah bagan (*chart*) yang menunjukkan alir (*flow*) didalam program atau prosedur sistem secara logika. Bagan alir (*flowchart*) digunakan terutama untuk alat bantu komunikasi dan dokumentasi [15].

Terdapat beberapa jenis *flowchart* diantaranya :

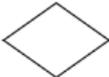
1. Bagan alir sistem (*system flowchart*)
2. Bagan alir dokumen (*document flowchart*)
3. Bagan alir skematik (*schematic flowchart*)
4. Bagan alir program (*program flowchart*)
5. Bagan alir proses (*process flowchart*)

Kesimpulan yang didapat yaitu “*Flowchart*” adalah suatu bagan dengan simbol – simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses dengan proses lainnya dalam satu program.

Berikut simbol – simbol yang ada didalam *flowchart* beserta fungsinya.

Tabel 2. 1. Simbol *Flowchart* dan Fungsinya

| Simbol | Nama | Keterangan |
|---|---------------------|---|
|  | <i>Input/output</i> | Mempresentasikan <i>input</i> data atau <i>output</i> data yang di proses atau informasi. |

| | | |
|---|-------------------------|---|
|  | Proses | Mempresentasikan operasi. |
|  | Penghubung | Keluar atau masuk dari bagian lain, khususnya halaman yang sama. |
|  | Anak Panah | Mempresentasikan alur kerja. |
|  | Keputusan | Keputusan dalam program. |
|  | <i>Terminal Port</i> | Awal / akhir <i>Flowchart</i> . |
|  | Dokumen | <i>Input / Output</i> dalam format yang dicetak. |
|  | <i>Manual operation</i> | Operasi manual. |
|  | <i>Off-line Storage</i> | Penyimpanan yang tidak dapat diakses oleh komputer secara langsung. |

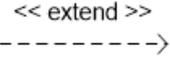
b. Use Case Diagram

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 [14]. Urutan langkah – langkah yang menerangkan antara pengguna sistem disebut *scenario* sedangkan pengguna disebut *actor*. *Actor* adalah sebuah perang yang biasa dimainkan oleh pengguna dalam interaksinya dengan sistem. Model *use case* adalah bagian dari model *requirement*. *Use case* dibuat sesuai berdasarkan keperluan *actor*.

Berdasarkan definisi diatas maka dapat disimpulkan bahwa “*Use case*” adalah kontruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata pengguna potensial yang terdiri dari sekumpulan *scenario* dan *actor*. Sedangkan *use case diagram* memfasilitasi komunikasi diantara analisis dan pengguna serta analisis klien. Berikut simbol – simbol yang sering digunakan dalam *Use Case Diagram* beserta fungsinya.

Tabel 2. 2. Komponen *Use Case Diagram*

| Simbol | Nama | Keterangan |
|--|-----------------|---|
|  | <i>Actor</i> | Orang proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. |
|  | <i>Use Case</i> | Fungsionalitas yang disediakan sistem sebagai unit – unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan menggunakan kata kerja di awali di awal frase nama – nama <i>use case</i> . |

| | | |
|---|--------------------|---|
|  | <i>Interaction</i> | <p>Digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek. Biasanya <i>interaction</i> dilengkapi dengan <i>teks</i> bernama <i>operation signature</i> yang tersusun dari nama operasi, parameter yang dikirim dan tipe parameter yang dikembalikan.</p> |
|  | <i>Dependency</i> | <p>Merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain. Elemen yang ada di bagian tanda panah adalah elemen yang tergantung pada elemen yang ada di bagian tanpa tanda panah.</p> |
|  | <i>Include</i> | <p>Menunjukkan bahwa suatu bagian dari elemen memicu eksekusi bagian dari elemen lain (yang ada pada tanda panah).</p> |
|  | <i>Extend</i> | <p>Menunjukkan bahwa suatu bagian dari elemen bisa disisipkan ke dalam elemen yang lain.</p> |

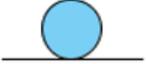
| | | |
|---|------------------------------|---|
|  | <p><i>Association</i></p> | <p>Menggambarkan navigasi antar <i>class</i>, berapa banyak obyek lain yang bisa berhubungan dengan satu obyek, dan apakah suatu <i>class</i> menjadi bagian dari <i>class</i> lainnya.</p> |
|  | <p><i>Generalization</i></p> | <p>Menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik. Dengan <i>generalization</i>, <i>class</i> yang lebih spesifik (<i>subclass</i>) akan menurunkan atribut dan operasi dari <i>class</i> yang lebih umum (<i>superclass</i>). Notasi ini digunakan pada konsep <i>inheritance</i>.</p> |

c. *Sequence Diagram*

Sequence diagram (diagram urutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi – interaksi antar objek didalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, *display*, dan sebagainya berupa pesan [16].

Sequence diagram digunakan untuk menggambarkan skenario atau rangkaian langkah – langkah yang dilakukan sebagai sebuah respon dari suatu kejadian yang menghasilkan *output* tertentu. *Sequence diagram* diawali dari apa yang me-*trigger* aktivitas tersebut proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Berikut simbol-simbol yang sering digunakan dalam *Sequence Diagram* beserta fungsinya.

Tabel 2. 3. *Komponen Sequence Diagram*

| Simbol | Nama | Keterangan |
|---|-------------------------|---|
|  | <i>Object Lifeline</i> | Menggambarkan <i>object</i> apa saja yang terlibat. |
|  | <i>Actor</i> | Menggambarkan <i>actor</i> yang terlibat. |
|  | <i>Lifeline Control</i> | Digunakan untuk menggambarkan sebuah form. |
|  | <i>Lifeline Entity</i> | Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan. |
|  | <i>Lifeline Control</i> | Digunakan untuk menghubungkan <i>boundary</i> dengan tabel. |
|  | <i>Time Active</i> | Menyatakan objek dalam keadaan aktif dan berinteraksi pesan. |
|  | <i>Synchronous</i> | Relasi ini digunakan untuk memanggil operasi atau <i>method</i> yang dimiliki oleh suatu objek. <i>Synchronous</i> mengharuskan kita menyelesaikan 1 proses baru kemudian memanggil proses berikutnya. |

| | | |
|---|---------------------|---|
|  | <i>Asynchronous</i> | <p>Relasi ini digunakan untuk memanggil operasi atau <i>method</i> yang dimiliki oleh suatu objek.</p> <p><i>Asynchronous</i> memberikan kita fasilitas untuk menjalankan proses lain ketika proses sebelumnya belum selesai.</p> |
|---|---------------------|---|

2.2.8. Basis Data (*Database*)

Basis data atau *Database* adalah kumpulan informasi yang disusun dan merupakan suatu kesatuan yang utuh yang disimpan di dalam perangkat keras (komputer) secara sistematis sehingga dapat diolah menggunakan perangkat lunak. Dengan sistem tersebut data yang terhimpun dalam suatu *database* dapat menghasilkan informasi yang berguna [17].

MySQL merupakan *software database open source* yang paling populer di dunia. MySQL menjadi pilihan utama bagi banyak pengembang *software* dan aplikasi hal ini dikarenakan kelebihan MySQL diantaranya sintaksnya yang mudah dipahami, didukung program – program umum seperti C, C++, Java, PHP, Python. Pengguna MySQL tidak hanya sebatas pengguna perseorangan maupun perusahaan kecil, namun perusahaan seperti Yahoo!, Google, Nokia, Youtube, Wordpress juga menggunakan DBMS MySQL. MySQL sebenarnya merupakan turunan salah satu konsep utama *database* sejak lama, yaitu SQL (*Structured Query Language*) [16].

A. SQL (*Structured Query Language*)

SQL adalah sebuah konsep pengoperasian *database* terutama untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keadaan suatu sistem *database* (DBMS) yang diketahui dari kerja optimizernya dalam melakukan proses perintah – perintah SQL yang dibuat oleh *user*

maupun program – program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *Query* data. SQL dibagi menjadi dua bentuk *query* yaitu :

1. *Data Definition Language* (DDL)

DDL adalah sebuah metode *query* SQL yang berguna untuk mendefinisikan data sebuah *database*. Adapun *query* yang dimiliki adalah :

- a. CREATE : Digunakan untuk pembuatan *table* dan *database*.
- b. DROP : Digunakan untuk penghapusan *table* dan *database*.
- c. ALTER : Digunakan untuk pengubahan struktur *table* yang dibuat baik menambah *field* (*add*), mengganti nama *field* (*change*), ataupun menamakannya kembali (*rename*) serta menghapus (*drop*).

2. *Data Manipulation Language* (DML)

DML adalah sebuah metode *query* SQL yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari *query* ini adalah untuk melakukan pemanipulasian *database* yang telah ada atau telah dibuat sebelumnya. Adapun *query* yang termasuk didalamnya adalah :

- a. INSERT : Digunakan untuk penginputan data pada *table* dan *database*.
- b. UPDATE : Digunakan untuk pengubahan data pada *table* dan *database*.
- c. DELETE : Digunakan untuk penghapusan data pada *table database*.

B. ERD (*Entity Relationship Diagram*)

ERD (*Entity Relationship Diagram*) adalah model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model [18]. Didalam hubungan ini tersebut dinyatakan yang utama dari ERD adalah menunjukkan objek data (*Entity*) dan hubungan

(*Relationship*), yang ada pada *Entity* berikutnya. Proses memungkinkan analis menghasilkan struktur basis data dapat disimpan dan diambil secara efisien.

Berikut daftar simbol ERD (*Entity Relationship Diagram*).

Tabel 2. 4. Simbol *Entity Relationship Diagram*

| Simbol | Nama | Keterangan |
|---|---------------------|---|
|  | <i>Entity</i> | Himpunan Entity atau himpunan entitas merupakan suatu entitas yang memiliki nama benda, orang atau lokasi digambarkan dengan persegi panjang. |
|  | <i>Attribute</i> | Atribut merupakan sifat atau karakteristik dari tiap – tiap entitas maupun tiap relationship. |
|  | <i>Relationship</i> | Himpunan Relasi merupakan kumpulan semua relasi diantara entitas – entitas yang terdapat dalam himpunan entitas. |
|  | <i>Link</i> | <i>Link</i> merupakan penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atribut. |

Elemen – elemen *Entity Relationship Diagram* :

a. Entity

Entity digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah suatu apa saja yang ada didalam sebuah sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data.

b. Attribute

Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap relationship.

c. Relationship

Relationship dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* hubungan alamiah yang terjadi antara entitas.

d. Kardinalitas

Kardinalitas relasi menyatakan jumlah maksimum tupel yang dapat berelasi dengan entitas dengan entitas lain. Terdapat empat macam kardinalitas relasi, yaitu:

1. Satu ke satu (*One to One*)

Setiap elemen dari Entitas A berhubungan paling banyak dengan elemen pada Entitas B. Demikian juga sebaliknya setiap elemen B berhubungan paling banyak satu elemen pada Entitas A.

2. Satu to banyak (*One to Many*)

Setiap elemen dari Entitas A berhubungan dengan maksimal banyak elemen pada Entitas B. Dan sebaliknya setiap elemen dari Entitas B berhubungan dengan paling banyak satu elemen di Entitas A.

3. Banyak ke satu (*Many to One*)

Setiap elemen dari Entitas A berhubungan paling banyak dengan satu elemen pada Entitas B. Dan sebaliknya setiap elemen dari Entitas B berhubungan dengan maksimal banyak elemen di entitas A.

4. Banyak ke banyak (*Many to Many*)
Setiap elemen dari Entitas A berhubungan maksimal banyak elemen pada Entitas B demikian sebaliknya.

2.2.9. Rekayasa Web

Rekayasa web adalah sebuah aplikasi yang menggunakan pendekatan sistematis, disiplin, dan terstruktur untuk pengembangan, operasi dan pemeliharaan aplikasi berbasis web (*Web-Based applications*) [12]. Kebanyakan *literature* tentang rekayasa perangkat web mengacu kepada sistem web, aplikasi berbasis, sistem berbasis web dan aplikasi web.

Rekayasa web adalah subdisiplin dari rekayasa perangkat lunak yang membantu menyediakan metodologi untuk merancang, mengembangkan, memelihara dan melibatkan aplikasi web. Munculnya disiplin rekayasa web ini menunjukkan suatu kebutuhan yang dipusatkan pada kesuksesan pengembangan aplikasi dan sistem berbasis web. Rekayasa web menggunakan sains, rekayasa dan pendekatan-pendekatan sistematis serta prinsip-prinsip manajemen untuk kesuksesan terhadap pengembangan, penyebaran, pemeliharaan aplikasi dan sistem web yang berkualitas tinggi. Rekayasa web membantu para pengembang system dibawah control, memperkecil resiko – resiko yang akan terjadi dan meningkatkan kualitas, dapat dipelihara dan memiliki skalabilitas aplikasi web.

2.2.10. PHP

PHP (*Hypertext Preprocessor*) adalah bahasa script yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk membuat program situs web dinamis [19]. PHP dapat digunakan dengan gratis (*free*) dan bersifat *Open Source*. PHP dirilis dalam lisensi *PHP license*. Untuk membuat program PHP kita diharuskan menginstal *web server* terlebih dahulu. Tentunya bahasa pemrograman PHP berbeda dengan HTML, pada PHP kode yang dibuat tidak dapat ditampilkan pada halaman / muka *website* begitu saja, tapi harus diproses terlebih dahulu oleh *web server*, *script* PHP disisipkan pada HTML. Manajemen *database* yang biasanya digunakan untuk pemrograman PHP misalnya

seperti MySQL, namun ada juga yang menggunakan *Oracle*, *Microsoft Access*, dan lain-lain. PHP disebut juga sebagai bahasa pemrograman *script server side*, karena PHP diproses pada komputer *server*.

~Halaman ini sengaja dikosongkan~