



## **BAB II**

# **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

## BAB II

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Berdasarkan penelitian sebelumnya yang berjudul “Rancang bangun aplikasi pendaftaran pasien online dan pemeriksaan dokter di klinik pengobatan berbasis web” [2], yang dilakukan oleh Asri Amaliza Fathia Mausea dan Ir. Andi Suprianto (2021). Tujuan dibangunnya rancang bangun aplikasi pendaftaran dan pemeriksaan yaitu untuk meningkatkan efisiensi waktu dalam hal pelayanan terhadap pasien di Klinik Medisina. Metode yang digunakan dalam penelitian adalah dengan menggunakan metode Spiral.

Penelitian yang lain dengan judul “Perancangan Sistem Informasi Pendaftaran dan Riwayat Pemeriksaan Pasien di Klinik Berbasis Web” [3], yang dilakukan oleh Asri Mulyani dan Sinta Nurfatonah (2021), penelitian ini bertujuan untuk membuat sistem informasi untuk pendaftaran dan mencatat riwayat pemeriksaan pasien yang bertempat di klinik berbasis web, dimana dalam perancangannya menggunakan Metodologi *Unified Software Development Process* yang terdiri dari beberapa tahapan analisis, perancangan, implementasi *Deployment* dan pengujian serta menggunakan *Unified Modelling Language* untuk pemodelannya dan pengujian, serta menggunakan blackbox testing untuk tahap pengujian. Penelitian ini menghasilkan aplikasi yang dapat menampilkan informasi seputar pendaftaran dan Riwayat pemeriksaan pasien dan mempermudah petugas dalam mengelola pemeriksaan dengan kelengkapan fitur cetak data.

Penelitian yang lain dengan judul “Aplikasi pendaftaran pasien rawat jalan online pada Klinik Mulya Medika” [4], yang diteliti oleh Verra Sofica, Shelpy Tresia Agista, Rahayu Ningasih, Minda Septiani (2019), penelitian ini dilakukan di klinik mulya medika dengan mengamati proses pendaftaran pasien rawat jalan *online* dengan menggunakan metode *waterfall*, dimulai dari dari pengumpulan data secara akurat, menganalisis kebutuhan perangkat lunak, mendesain aplikasi yang akan dibuat, pembuatan program dan tahap terakhir melakukan pengujian. Tujuan aplikasi ini dibuat untuk mempermudah dokter mendapatkan data Riwayat kunjungan pasien dan data rekam

medis pasien dan mempermudah pada klinik dalam menyampaikan informasi mengenai klinik dan dapat diakses masyarakat tanpa terbatas ruang dan waktu.

Penelitian yang lain dengan judul “Rancang Bangun Sistem Informasi Rekam Medis Rawat Jalan pada Klinik Darussalam”[5], yang diteliti oleh Noor Cahyaningrum, Oddy Virgantara Putra, dan Anisa Dzulkarnain (2022), penelitian ini dilakukan untuk mempermudah admin staff, dan dokter dalam pengelolaan data pasien dan juga data rekam medis. Adapun metode yang digunakan dalam penelitian ini yaitu menggunakan Metode Waterfall.

Penelitian yang lain berjudul “Sistem Informasi Pelayanan Pasien Rawat Jalan Menggunakan Aplikasi pada Klinik Annisa Di Ciracas” [6], yang diteliti oleh Suzuki Syofian dan Wiwit Try Widyantoro (2019), penelitian ini dilakukan untuk memberikan informasi pelayanan pasien rawat jalan agar proses pendataan cepat, tepat dan akurat. Metode yang digunakan dengan *Programmable Hypertext Preprocessor (PHP)* dan *Structured Query Language* atau *Management System*.

**Tabel 2. 1** Tabel Kajian Penelitian Sebelumnya

No	Penulis (Tahun)	Judul	Tujuan	Metode Pengembangan	Proses dalam Sistem	Hasil
1	(Manusea, 2021)	Rancang bangun aplikasi pendaftaran pasien online dan pemeriksaan dokter di klinik pengobat an berbasis web	Memudahkan pasien dalam mengetahui hasil rekam mediknya	Metode yang digunakan dalam penelitian adalah dengan menggunakan metode spiral	<ul style="list-style-type: none"> <li>• Login</li> <li>• Registrasi berobat pasien</li> <li>• Kelola data rekam medis</li> <li>• Kelola penjadwalan dokter</li> </ul>	<ul style="list-style-type: none"> <li>• Pasien juga dapat melihat hasil rekam medisnya setelah melakukan pemeriksaan</li> <li>• Memudahkan pasien melakukan pendaftaran pasien secara online.</li> </ul>

2	(Mulyani & Nurfarona h, 2021)	Perancangan Sistem Informasi Pendaftaran Riwayat Pemeriks aan Pasien di Klinik Berbasis Website	Membuat sistem informasi pendaftaran dan mencatat Riwayat pemeriksaan pasien.	Metode yang digunakan yaitu Metodologi Unified Software Development Process yang terdiri dari beberapa tahapan analisis, perancangan, implementasi Deployment dan pengujian serta menggunakan Unified Modelling Language untuk pemodelannya dan pengujian.	<ul style="list-style-type: none"> <li>• Pendaf taran pasien data</li> <li>• Kelola rekam medis</li> </ul>	<ul style="list-style-type: none"> <li>• Dapat menampilkan informasi seputar pendaftaran an dan riwayat pemeriks aan pasien</li> <li>• Dapat mengelola pemeriks aan dengan kelangkaan fitur cetak data.</li> </ul>
3	Sofea et al., 2019)	Aplikasi pendaftaran pasien rawat jalan online pada Klinik Mulya Medika	Memberikan kemudahan kepada masyarakat yang ingin memperoleh informasi meliputi profil, sejarah, fasilitas rawal dokter dll.	Metode yang digunakan adalah metode waterfall	<ul style="list-style-type: none"> <li>• Memilih dokter</li> <li>• Melihat jadwal dokter</li> <li>• Pendaftaran pasien</li> </ul>	<p>Yang dihasilkan dari aplikasi pendaftaran pasien dengan menggunakan metode waterfall dan melakukan pengujian menggunakan black box testing</p>

4	(Noor Cahyaningrum, Oddy Virgantara Putra, dan Anisa Dzulkarnain, 2022)	Rancang Bangun Sistem Informasi Rekam Medis Rawat Jalan pada Klinik Darussalam	Untuk mempermudah staf, dan dokter dalam pengelolaan data pasien dan juga data rekam medis.	Metode yang digunakan dalam penelitian ini yaitu menggunakan metode waterfall	<ul style="list-style-type: none"> <li>• Pendaftran pasien</li> <li>• Pencarian data</li> <li>• Pengolahan data rekam medis</li> </ul>	Sistem Informasi dapat mempercepat dan mempermudah dalam pencarian data-data pasien atau riwayat pasien
5	(Syofian & Widyanto, 2018)	Sistem Informasi Pelayanan pasien rawat jalan menggunakan aplikasi pada Klinik Annisa di Ciracas	Membuat aplikasi pelayanan pasien rawat jalan yang dikembangkan untuk pendataan cepat, tepat/akurat	Metode yang digunakan dengan <i>Programmable Hypertext Preprocessor (PHP)</i> dan <i>Structured Query Language</i> atau <i>Management System</i>	<ul style="list-style-type: none"> <li>• Pendaftran pasien</li> <li>• Pelayanan pada pasien</li> <li>• Rekam Medis</li> </ul>	Sistem Informasi pendaftran pasien yang dikembangkan untuk pendataan pasien yang tersimpan di dalam database agar lebih efisien.

## 2.2 Landasan Teori

Landasan teori berisi hal-hal atau teori-teori yang berkaitan dengan permasalahan dan ruang lingkup permasalahan sebagai landasan dalam pembuatan laporan ini.

### 2.2.1 Konsep Dasar Sistem

Pendekatan sistem berhubungan dengan pemahaman gabungan urutan operasi, jaringan kerja yang melibatkan elemen/komponen untuk menjamin penanganan yang seragam dari urutan operasi yang terjadi. Maka dari itu diperlukan pemahaman mengenai pengertian, karakteristik dan klasifikasi sistem.

Pengertian sistem (Rina & Fatkur, 2019) adalah sekumpulan objek-objek yang saling berelasi dan berinteraksi serta hubungan antara objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai satu tujuan. Dengan demikian, secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur variable-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung satu sama lain [7].

Pengertian Sistem menurut (Jeperson Hutahaeen, 2021:2) dalam buku yang berjudul Konsep Sistem Informasi, Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu [7].

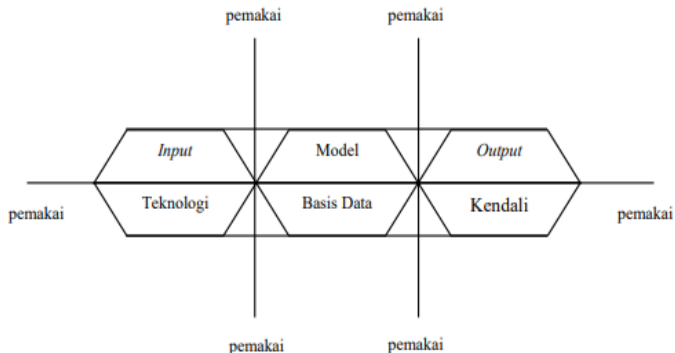
### 2.2.2 Sistem Informasi

Menurut Jeperson Hutahaeen (2021:15), sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan. Sistem Informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building blok*) yaitu [7] :

a. Blok masukan (*Input Block*)

Input mewakili data yang masuk kedalam sistem informasi. Input disini termasuk metode-metode dan media yang digunakan untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen dasar.

- b. Blok Model (*Model Block*)  
Blok ini terdiri dari kombinasi prosedur, logika dan metode matematik, yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang sudah diinginkan.
- c. Blok Keluaran (*Output Block*)  
Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
- d. Blok Teknologi (*Technology Block*)  
Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian diri secara keseluruhan.
- e. Blok Basis Data (*Database block*)  
Merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.
- f. Blok Kendali (*Control Blok*)  
Seberapa pengendali perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atau bila terlanjur kesalahan dapat langsung diatasi.



**Gambar 2. 1** Blok Sistem Informasi yang berinteraksi

Sedangkan menurut Marimin et al [8], Sistem Informasi adalah suatu komponen yang saling berhubungan dengan proses penciptaan dan penyampaian informasi dalam perusahaan, yang memproses

masukannya berupa sumber data, kemudian diproses dengan komponen perangkat keras, perangkat lunak, dan manusia yang menghasilkan informasi sebagai output

### **2.2.3 Pendaftaran**

Menurut Kamus Besar Bahasa Indonesia (KBBI), arti kata pendaftaran adalah proses, cara, perbuatan mendaftar (mendaftarkan), pencatatan nama, alamat, dan sebagainya; perihal mendaftar (mendaftarkan). Pendaftaran adalah pencatatan hal atau identitas seperti nama, alamat, dan sebagainya dalam suatu Lembaga [9].

### **2.2.4 Pasien**

Berdasarkan keputusan Menteri Kesehatan RI Nomor 296/MENKES/PER/III/2008 [10], tentang rekam medis, pasien adalah setiap orang yang melakukan konsultasi masalah kesehatannya untuk memperoleh pelayanan kesehatan yang diperlukan baik secara langsung maupun tidak langsung kepada dokter.

### **2.2.5 Klinik**

Permenkes RI No.9,2014, Klinik adalah fasilitas pelayanan kesehatan yang menyelenggarakan dan menyediakan pelayanan medis dasar dan atau spesialisik, diselenggarakan oleh lebih dari satu jenis tenaga kesehatan dan dipimpin oleh seorang tenaga medis [11].

### **2.2.6 Rekam Medis**

Rekam Medis merupakan berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, Tindakan dan pelayanan lain yang telah diberikan kepada pasien. Penerapan teknologi informasi akan membuat pelayanan rekam medis lebih efisien dan efektif [2].

### **2.2.7 Website**

*Website* adalah kumpulan dari keseluruhan halaman-halaman web yang berisi sebuah data atau informasi baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman [12].



### 2.2.8 Rekayasa Perangkat Lunak

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].

Rekayasa Perangkat Lunak memiliki beberapa pokok Bahasa Penting yaitu :

- a. *Domain Engineering*  
Kemampuan untuk memahami suatu permasalahan yang muncul dan dijadikan suatu proyek perangkat lunak.
- b. *Requirement Engineering*  
Kemampuan untuk memahami kebutuhan seorang pengguna sekaligus dalam memecahkan permasalahan.
- c. *Software Design*  
Kemampuan untuk memahami dan mengimplementasikan dalam perancangan perangkat lunak yang didalamnya terdapat aspek HCI (*Human Computer Interaction*)
- d. *Development*  
Proses pengembangan yang melibatkan pembelajaran mengenai algoritma, Bahasa pemrograman yang diterapkan 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 adalah metode *waterfall* menurut Ian Sommerville. Metode *waterfall* adalah suatu proses pengembangan perangkat lunak secara berurutan dan diibaratkan seperti air terjun karena kemajuannya dipandang terus mengalir ke bawah melewati fase perencanaan, pemodelan, implementasi, dan pengujian [14]. Dalam proses implementasi metode *waterfall*, sebuah langkah dimulai dari tahapan pertama yang harus diselesaikan sebelum melanjutkan ke tahap berikutnya. Penggunaan metode *waterfall* dapat memungkinkan adanya sedikit perubahan terhadap proyek yang sedang berlangsung. Metode ini mengusulkan sebuah pendekatan kepada pengembangan *software* yang

sistematik dan sekuensial yang dimulai dari tingkat kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan

## 2. Metode Pengujian Sistem

Pengujian ini dilakukan pada sistem komputer (*computer-based system*) untuk mengurangi masalah yang berkaitan dengan antarmuka dan perancangan pada jalur penanganan terjadinya kesalahan antar sistem pada perangkat lunak [15]. Perangkat lunak dapat diuji dengan dua cara yaitu :

- a. *Black Box Testing* merupakan pengujian yang untuk menguji kelancaran pada program yang dikembangkan. Pengujian ini sangat penting supaya tidak terjadi kesalahan pada proses atau alur yang dibuat

### 2.2.9 Pemrograman Berorientasi Objek

Metodologi pemrograman berorientasi objek merupakan pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam kelas-kelas atau objek-objek. Setiap objek dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. Beberapa objek berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya [16]. Beberapa konsep dasar yang harus dipahami mengenai metodologi pemrograman berorientasi objek :

#### 1. Kelas (*class*)

*Class* adalah *prototype*, atau *blueprint*, atau rancangan yang mendefinisikan variabel dan metode-metode pada seluruh objek tertentu. *Class* berfungsi untuk menampung isi dari program yang akan dijalankan, didalamnya berisi atribut / *type* data dan *method* untuk menjalankan suatu program.

#### 2. Objek (*object*)

Objek adalah *instance* dari *class*. Jika *class* secara umum mempresentasikan (*templates*) sebuah objek, sebuah *instance* adalah representasi nyata dari *class* itu sendiri.

#### 3. Metode (*method*)

Metode adalah merupakan suatu operasi berupa fungsi-fungsi yang dapat dikerjakan oleh suatu objek. *Method* didefinisikan pada *class* akan tetapi dipanggil melalui objek.

**4. Atribut (*attribute*)**

Atribut merupakan nilai data yang terdapat pada suatu objek yang berasal dari *class*. *Attribute* mempresentasikan karakteristik dari suatu objek.

**5. Abstraksi (*abstract*)**

Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

**6. Enkapsulasi (*encapsulation*)**

Pembungkusan *attribute* data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

**7. Pewarisan (*inheritance*)**

Mekanisme yang memungkinkan suatu objek mewarisi sebagian atas seluruh definisi objek lain sebagai bagian dari dirinya.

**8. Antarmuka (*interface*)**







Antarmuka atau *interface* sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah *interface* dapat diimplementasikan oleh kelas lain.

Pada pemrograman berorientasi objek, UML digunakan untuk pemodelan sistem. UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya.

**a. Use Case Diagram**

Merupakan model UML gunanya untuk mengetahui hubungan antara *user* dengan sistem. Diagram ini sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan. Berikut merupakan tabel simbol Use Case :

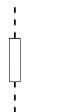

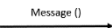



**Tabel 2. 2** Tabel *Use Case Diagram*

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.
4.		<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

**b. Sequence Diagram**

Merupakan *diagram* yang digunakan untuk menampilkan dan menjelaskan interaksi antar objek pada sebuah sistem. *Sequence diagram* juga akan menampilkan sebuah perintah atau pesan yang dikirim. Berikut merupakan tabel *Sequence Diagram* :



**Tabel 2. 3** Tabel 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.

### c. *Class Diagram*

*Class diagram* merupakan diagram yang digunakan untuk menampilkan beberapa kelas yang ada pada sebuah sistem yang dibangun, *class diagram* menunjukkan bagaimana *class* dalam sistem yang akan dibangun saling berkolaborasi untuk mencapai suatu tujuan [17]. *Class diagram* digambarkan dengan bentuk kotak.

**Tabel 2. 4** Simbol Class Diagram

No	Simbol	Nama	Keterangan
1.		Kelas	Kelas pada struktur sistem.
2.		Asosiasi/ <i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

### 2.2.10 Basis Data

Basis data merupakan kumpulan data yang saling terintegrasi serta berelasi secara *logic* serta dirancang untuk melengkapi informasi yang dibutuhkan [18].

#### A. *Database Management System (DBMS)*

*Database Management System (DBMS)* perangkat lunak komputer yang berinteraksi dengan Pengguna, aplikasi lain, dan basis data untuk mengambil dan menganalisis data. DBMS memungkinkan definisi, pembuatan, *query*, *update*, dan administrasi basis data [19]. DBMS menyediakan fasilitas antara lain :

#### 1. *Data Definition Language (DDL)*

DDL memperbolehkan pengguna untuk mendeskripsikan *database*, misalnya merinci tipe dan batasan data yang akan disimpan dalam *database*. Terdapat tiga perintah penting dalam DDL, yaitu :

- a. *CREATE* adalah perintah yang digunakan untuk membuat, termasuk didalamnya membuat *database* baru, tabel baru, *view* baru, dan kolom baru. Contoh : *CREATE DATABASE* nama *database*.

- b. *ALTER* berfungsi untuk mengubah struktur tabel yang telah dibuat. Mencakup didalamnya mengubah nama tabel, menambah kolom, mengubah kolom, menghapus kolom, dan memberikan atribut pada kolom. Contoh : *Alter Table* nama tabel *ADD* nama kolom data *type*.
- c. *DROP* perintah *DROP* berfungsi untuk menghapus *database* atau tabel. Contoh : *DROP DATABASE* nama *database*

## 2. **Data Manipulation Language (DML)**

DML memperbolehkan pengguna untuk memanipulasi data dalam *database* yang telah dibuat. Terdapat 4 (empat) perintah penting dalam DML, yaitu :

- a. *INSERT* perintah ini digunakan untuk memasukkan data baru ke dalam sebuah tabel. Perintah ini tentu saja bisa dijalankan ketika *database* dan tabel sudah dibuat. Contoh : *INSERT INTO* nama tabel *VALUES* (data1, data2, dst..)
- b. *SELECT* perintah ini digunakan untuk mengambil dan menampilkan data dari tabel atau bahkan dari beberapa tabel dengan penggunaan relasi. Contoh : *SELECT* nama\_kolom1, nama\_kolom2 *FROM* nama tabel.
- c. *UPDATE* perintah ini digunakan untuk memperbaharui data pada sebuah tabel. Contoh : *UPDATE* nama\_tabel *SET* kolom1=data1, kolom2=data2,.. *WHERE* kolom=data.
- d. *DELETE* perintah ini digunakan untuk menghapus data dari sebuah tabel. Contoh : *DELETE FROM* nama tabel *WHERE* kolom=data.






## B. **MySQL**

MySQL merupakan *database server* yang terkenal dan banyak digunakan untuk membangun sebuah aplikasi *web* yang databasenya digunakan sebagai sumber pengolahan data [20].

### C. *Entity Relationship Diagram (ERD)*

*Entity Relationship Diagram (ERD)* merupakan suatu bentuk hubungan atau rancangan kegiatan pada sebuah sistem mempunyai fungsi dan berkaitan langsung di dalam proses tersebut. ERD merupakan suatu bentuk pemodelan dari basis data relasional yang dibangun berdasarkan persepsi dalam dunia nyata terdiri dari objek-objek yang saling berhubungan satu dengan lainnya. Suatu objek disebut *entity* sedangkan hubungan yang dimiliki disebut dengan *relationship*. *Entity* sifatnya unik serta memiliki atribut yang menjadi pembeda dengan *entity* lainnya.

**Tabel 2. 5 Simbol ERD**



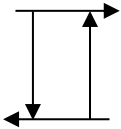
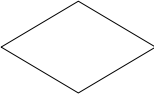

No.	Nama	Simbol	Keterangan Fungsi
1.	Entitas		Persegi panjang menyatakan himpunan entitas adalah orang, kejadian, atau berada.
2.	Atribut		Atribut merupakan informasi yang diambil tentang sebuah entitas.
3.	Relasi		Belah ketupat menyatakan himpunan relasi merupakan hubungan antar entitas.
4.	<i>Link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


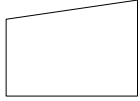


#### D. *Flowchart*

*Flowchart* atau biasa disebut dengan diagram alir merupakan suatu jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem. *Flowchart* sebagai bukti dokumentasi untuk menjelaskan gambaran yang logis pada sebuah sistem yang dibangun. Untuk menghubungkan satu proses ke proses selanjutnya dapat digambarkan dengan menggunakan garis penghubung [21]

**Tabel 2. 6** Simbol - 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>Flow</i>	Menghubungkan antara simbol satu dengan simbol yang lain atau menyatakan jalannya arus dalam suatu proses. Simbol arus ini sering disebut juga dengan <i>connecting line</i> .
6.		<i>Decision</i>	Memilih proses berdasarkan kondisi yang ada.
7.		<i>Document</i>	Merupakan simbol untuk data yang terbentuk informasi.

<b>No.</b>	<b>Simbol</b>	<b>Nama</b>	<b>Keterangan</b>
8.		<i>Manual Operation</i>	Menunjukkan pengolahan yang tidak dilakukan oleh komputer/pc.
9.		<i>Manual Input</i>	Memasukan data secara manual <i>on-line</i> keyboard.

*~Halaman ini sengaja dikosongkan~*