



BAB II

TINJAUAN PUSTAKA DAN

LANDASAN TEORI

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian mengenai sistem informasi registrasi pernah dilakukan oleh Nopriandi [2]. Penelitian ini bertujuan untuk mengatasi masalah pengolahan sistem informasi registrasi mahasiswa pada Sekolah Tinggi khususnya dibagian administrasi, untuk memperoleh suatu keuntungan dalam memberikan informasi yang cepat dan tepat dalam pengolahan data mahasiswa. Metode yang digunakan yaitu pengumpulan data dan observasi secara langsung pada bagian pengolahan sistem informasi registrasi mahasiswa. Hasil dari penelitian ini sistem informasi registrasi mahasiswa yang menerapkan database, maka pelayanan terhadap mahasiswa seperti penambahan, penghapusan, koreksi atau penyisipan dan pembuatan laporan registrasi baik yang sudah ada maupun data yang baru akan dapat diatasi dalam waktu yang relatif cepat karena semua data tersimpan dalam sebuah *database*.

Penelitian ini dilakukan oleh Fikri [3]. Tujuan penelitian ini perlu dibuat aplikasi pengelolaan registrasi dan penerimaan berbasis web untuk menjamin kinerja dan keamanan data terkait kegiatan registrasi, her-registrasi dan penerimaan siswa baru di SMP Negeri 2 Baleendah. Aplikasi web ini dibuat menggunakan metode *waterfall*, *framework CodeIgniter*, bahasa pemrograman *PHP*, *Database Developer MySql*, *Web Server Apache*. Hasil dari penelitian ini untuk dapat beroperasi dengan baik dan dapat menjadi solusi bagi sekolah dalam proses pengelolaan data kegiatan registrasi dan penerimaan siswa agar lebih terintegrasi dan terjamin validasi datanya serta dapat mengurangi resiko dokumen fisik yang mudah hilang ataupun rusak.

Penelitian selanjutnya dilakukan oleh Suyanto [4]. Penelitian ini dilakukan untuk membuat sebuah perancangan sistem informasi pendaftaran mahasiswa aktif kembali di STMIK AMIKOM Yogyakarta. Metode yang digunakan dalam penelitian ini adalah metode *prototyping* dalam SDLC (*System Development Life Cycle*). Hasil dari penelitian ini adalah dokumen untuk pengembangan sistem berikutnya, Sistem informasi ini merupakan pemecahan masalah *human error* yang sebelumnya manual mengatur rumus di *Microsoft Excel*, sekarang otomatis dihitung oleh sistem dan Sistem informasi pendaftaran aktif kembali memberikan laporan dan kwitansi yang lebih presisi.

Penelitian serupa dilakukan oleh Anisah [5]. Tujuan dari penelitian ini dapat mempermudah saat proses pendaftaran siswa baru,

penyimpanan data yang berkaitan dengan penerimaan siswa baru bisa disimpan secara rapi dengan menggunakan *database*, dan informasi yang berkaitan dengan penerimaan siswa baru bisa diperoleh secara tepat dan akurat sebagai bahan pertimbangan dalam pengambilan sebuah keputusan. Metodologi penelitian yang digunakan dalam penelitian ini adalah dengan menggunakan model *waterfall*. Hasil dari penelitian ini yaitu dapat ditingkatkan menjadi sebuah sistem yang terkomputerisasi sehingga penyimpanan yang sebelumnya masih menggunakan lemari arsip dapat di *update* menggunakan *database*, sehingga data dapat disimpan dengan rapi dan pada saat ingin mencari data yang diinginkan akan dapat dengan mudah untuk menemukannya.

Penelitian yang sama dilakukan oleh Taufiq [6]. Tujuan penelitian ini adalah untuk memberikan layanan yang terbaik kepada calon mahasiswa mulai sejak memperkenalkan produk hingga calon mahasiswa tersebut diterima sebagai mahasiswa, baik dalam bentuk *offline* maupun *online*, maka dibutuhkan strategi khusus yang berkaitan dengan media penyajian informasi yang terintegrasi secara spesifik agar dapat menghasilkan Informasi yang berkualitas. Hasil dari penelitian ini adalah agar tercapai optimasi sistem secara keseluruhan didalam manajemen perusahaan, sehingga pencapaian target yang diharapkan dapat terwujud dengan baik, sesuai yang direncanakan.

Penelitian lainnya dilakukan oleh Adisaputra [7]. Tujuan dari penelitian ini adalah untuk mencapai sistem informasi pendaftaran dan penerimaan siswa yang baik, dapat dijadikan sarana yang penting bagi setiap sekolah dalam meningkatkan mutu dari program pendidikan. Tugas Akhir ini menggunakan metode pengerjaan *Waterfall* dengan menggunakan *Framework* bahasa pemrograman *PHP CodeIgniter* dan *database MySQL*. Hasil dari penelitian ini dapat mempermudah bagian Tata Usaha di SMP Negeri 2 Cianjur dalam menangani registrasi pendaftaran, penerimaan siswa baru, menangani dalam penerimaan siswa pindahan maupun siswa keluar. Sehingga data calon maupun riwayat siswa dapat disimpan dengan baik.

Pada penelitian kali ini, akan dibuat sistem informasi registrasi dan heregistrasi mahasiswa berbasis *web* di Politeknik Negeri Cilacap. Aplikasi web ini dibuat menggunakan metode *prototype*, *native*, bahasa pemrograman *PHP*, *Database Developer MySql*, *Web Server Apache*. Metode pengujian yang digunakan adalah *black box*. Terdapat fitur e-mail gateway untuk memberikan notifikasi pemberitahuan kepada mahasiswa terkait pembayaran. Hasil dari penelitian ini adalah sebuah

sistem yang dapat memudahkan mahasiswa baru dalam melakukan registrasi dan memudahkan mahasiswa lama yang akan melakukan heregistrasi, serta memudahkan bagian akademik untuk memonitori status mahasiswa yang aktif dan mahasiswa tidak aktif. Selain itu sistem ini dapat membantu proses pelaksanaan pembelajaran bagi mahasiswa di Politeknik Negeri Cilacap.

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.

2.1.1 Sistem Informasi

Sistem adalah element-element yang saling berhubungan membentuk satu kesatuan untuk melaksanakan suatu tujuan pokok dari sistem tersebut. Untuk mengetahui sistem atau bukan, antara lain dapat dilihat dari ciri-cirinya . ada beberapa rumusan mengenai sistem ciri-ciri sistem ini yang pada dasarnya satu sama lain saling melengkapi. Pada umumnya ciri-ciri sistem ini adalah : bertujuan, punya batas, terbuka, tersusun dari sub sistem, ada yang saling berkaitan dan tergantung merupakan kebulatan yang sistematis [2].

Informasi adalah data yang diolah menjadi lebih berguna dan berarti bagi penerimanya dan untuk mengurangi ketidakpastian dalam proses pengambilan keputusan mengenai suatu keadaan [2].

Sistem informasi merupakan suatu cara tertentu untuk menyediakan informasi yang dibutuhkan untuk beroperasi dengan cara yang menguntungkan. Pada dasarnya sistem informasi merupakan suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. [2].

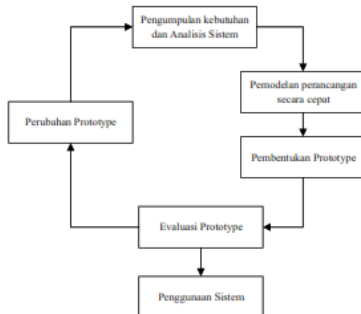
2.1.2 Rekayasa Web

Rekayasa Web merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin [9]. Berikut adalah metode dan tools yang digunakan :

A. 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 yang digunakan pada penelitian ini adalah metode *prototype*. Metode ini dibuat secara

terstruktur dan memiliki beberapa tahap-tahap yang harus dilalui dalam pembuatannya namun jika tahap final dinyatakan bahwa sistem yang telah dibuat belum sempurna maka sistem dievaluasi kembali.



Gambar 2. 1 Pemodelan prototype [1].

Metode *prototype* memiliki 6 tahapan, yaitu [1] :

a. Pengumpulan Kebutuhan dan Analisis Sistem

Tahapan pertama yang digunakan dalam penelitian ini adalah pengumpulan kebutuhan dan analisis sistem. Pada tahapan ini dilakukan identifikasi kebutuhan sistem dan garis besar dari sistem yang akan dibuat. Dalam mengembangkan sistem informasi tugas akhir ini, diperlukan biodata mahasiswa dan judul dari tugas akhir.

b. Pemodelan perancangan secara cepat

Tahapan selanjutnya adalah pemodelan perancangan secara cepat yang digunakan sebagai acuan yang digunakan dalam pembuatan model *prototype*.

c. Pembentukan *prototype*

Dalam tahap ini, dilakukan pembentukan *prototype* berdasarkan rancangan pemodelan yang telah dilakukan sebelumnya.

d. Evaluasi *prototype*

Pada tahapan ini, dilakukan evaluasi terhadap *prototype* yang disesuaikan dengan kebutuhan. Jika belum sesuai dengan kebutuhan, maka dapat melakukan tahap selanjutnya yaitu melakukan perubahan *prototype*.

e. Perubahan *prototype*

Tahapan ini dilakukan untuk menyempurnakan *prototype* yang dibangun agar menghasilkan *prototype* yang sesuai dengan kebutuhan.

f. Penggunaan Sistem

Tahap terakhir dari metode penelitian ini adalah penggunaan sistem. Pada tahap ini sistem yang telah dievaluasi siap untuk digunakan.

B. Metode Pengujian Sistem

Terkait dengan pengujian program, pengujian yang akan digunakan adalah pengujian *black-box*. Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan rekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut [10] :

1. Fungsi-fungsi yang tidak benar atau hilang,
2. Kesalahan *interface*,
3. Kesalahan dalam struktur data atau akses *database* eksternal,
4. Kesalahan kinerja,
5. Inisialisasi dan kesalahan terminal.

Pengujian *black-box* cenderung diaplikasikan selama tahap akhir pengujian. Karena pengujian *black-box* memperhatikan struktur struktur kontrol, maka perhatian berfokus pada domain informasi.

Pengujian di desain untuk menjawab pertanyaan-pertanyaan berikut [11].

1. Bagaimana validitas fungsional diuji?
2. Kelas *input* apa yang akan *test case* menjadi baik?
3. Apakah sistem sangat sensitif terhadap harga *input* tertentu?
4. Bagaimana batasan dari suatu data diisolasi?
5. Kecepatan data apa dan volume data apa yang dapat ditolelir oleh sistem?
6. Apa pengaruh kombinasi tertentu dari data terhadap operasi sistem?

Pengujian *black-box* ini terdapat beberapa proses. Proses-proses yang ada dalam pengujian ini diantaranya :

- a. Menganalisa kebutuhan dan spesifikasi dari perangkat lunak.
- b. Pemilihan jenis *input* yang memungkinkan menghasilkan *output* dengan benar serta jenis *input* yang memungkinkan *output* salah pada perangkat lunak yang sedang diuji.
- c. Menentukan *output* untuk satu jenis *input*.
- d. Pengujian dilakukan dengan *input-input* yang telah benar-benar diseleksi.

- e. Melakukan pengujian
- f. Perbandingan *output* yang dihasilkan dengan *output* yang diharapkan.
- g. Pemeliharaan (*maintenance*)
Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Pada tahap ini dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

C. Uji Usability

Uji usability digunakan untuk mengetahui suatu ukuran kualitas pengalaman pengguna ketika berinteraksi dengan produk atau sistem. Uji usability pada penelitian ini dilakukan berdasarkan hasil pengisian instrument oleh responden. Instrument terbagi menjadi 22 butir (item), dimana 22 butir tersebut dikelompokkan menjadi lima (5) variabel yang digunakan dalam mengukur usability, perlu menggunakan pertanyaan-pertanyaan yang memiliki komponen-komponen tertentu untuk menggali pendapat responden, antara lain *Learnability*, *Efficiency*, *Memorability*, *Errors*, dan *Satisfaction* [12].

D. Tools/ Alat Bantu Penelitian

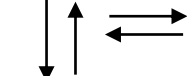
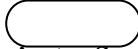



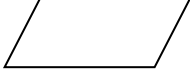

1. Flowchart

Flowchart adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir (*flowchart*) digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi [13]. Ada beberapa jenis-jenis *flowchart* diantaranya :

- a. Bagan alir sistem (*system flowchart*)
- b. Bagan alir dokumen (*document flowchart*)
- c. Bagan alir skematik (*schematic flowchart*)
- d. Bagan alir program (*program flowchart*)
- e. Bagan alir proses (*process flowchart*)

Simbol-simbol dalam *flowchart* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol Flowchart

No	Simbol	Keterangan
1	 <p>Flow Direction Symbol</p>	Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.
2	 <p>Terminator Symbol</p>	Yaitu simbol untuk permulaan (<i>start</i>) atau akhir (<i>end</i>) dari suatu kegiatan.
3	 <p>Processing Symbol</p>	Yaitu simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.
4	 <p>Manual Operation Symbol</p>	Yaitu simbol yang menunjukkan pengolahan data yang tidak dilakukan oleh komputer.
6	 <p>Decision Symbol</p>	Yaitu simbol untuk pemilihan proses berdasarkan kondisi yang ada.
7	 <p>Input-Output Symbol</p>	Yaitu simbol yang menyatakan proses input dan output tanpa tergantung dari jenis peralatannya.
6	 <p>Document Symbol</p>	Yaitu simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

2. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah hasil kerja dari konsorium berbagai organisasi yang berhasil dijadikan sebagai standar baku dalam *Object Oriented Analysis* dan *Design* (OOAD). Macam-





macam dari *Unified Modeling Language* (UML) antara lain [11] : *use case diagram*, *sequence diagram* dan *class diagram*.

a. Use Case

Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem tersebut *scenario* sedangkan pengguna disebut *actor*. *Actor* adalah sebuah peran yang biasa dimainkan oleh pengguna dalam interaksinya dengan sistem. Model *use case* adalah bagian dari model *requirement*. Definisi lain *use case* adalah abstraksi dari interaksi antara sistem dan *actor*. *Use case* dibuat berdasarkan keperluan *actor*.

Berdasarkan definisi diatas maka dapat disimpulkan bahwa *Use Case* adalah kontruks untuk mendeskripsikan bagaimana sistem akan terlihat dimata pengguna potensial yang terdiri dari sekumpulan *scenario* dan *actor*. Sedangkan *use case diagram* memfasilitasi komunikasi diantara analis dan pengguna serta analis dan klien. Simbol *Use Case* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol Use Case Diagram







No.	Simbol	Nama Simbol dan Keterangan
1.		<i>Actor</i> , Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<i>Include</i> , Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
3.		<i>Association</i> , yang menghubungkan antara objek satu dengan objek lainnya
4.		<i>Use Case</i> , Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.

b. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah obyek dan *message* yang diletakan antara obyek-obyek didalam *use case*.

Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*. *Sequence Diagram* menambahkan dimensi waktu pada interaksi diantara obyek. Simbol-simbol yang dipakai dalam pembuatan *sequence diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol Sequence Diagram

GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
	<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
	<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari foem
	<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
	<i>A focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message
	<i>A message</i>	Menggambarkan Pengiriman Pesan

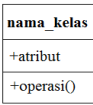

c. Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas. Sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas [14]. Diagram kelas dibuat agar *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

Class diagram membantu dalam visualisasi struktur kelas – kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak. Class diagram memperlihatkan hubungan antar kelas dan penjelasan detail tiap – tiap kelas di dalam model desain (dalam logical view) dari suatu sistem. Selama proses analisis, class diagram memperlihatkan aturan – aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama proses analisis, *class diagram* memperlihatkan aturan – aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap desain, class diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. Berikut beberapa simbol dari *class diagram* dapat dilihat pada Tabel 2.4 :

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> .

2.1.3 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan proyek yang berisi data dan operasi yang diperlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara

bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis [14].

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus menjadi kelompok data dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut, sifat, komponen lainnya, dan dapat berinteraksi satu sama lain.

2.1.4 Basis Data

Basis data adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file lain sehingga membentuk satu bangun data untuk penginformasian suatu perusahaan instansi, dalam bahasan tertentu.

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi *user* serta menggunakan perintah standar *Structured Query Language* (SQL). MySQL memiliki dua bentuk lisensi, yaitu *Free Software* dan *Shareware*. MySQL yang biasa kita gunakan adalah MySQL *Free Software* yang berada dibawah Lisensi *General Public License* (GPL) [15].

MySQL juga dapat didefinisikan sebagai sebuah database server, dapat juga berperan sebagai *client* sehingga sering disebut *database client* / server yang *open source* dengan kemampuan dapat berjalan baik di Operasi Sistem maupun dengan *Platform Windows* maupun *Linux*.

MySQL dikembangkan oleh sebuah perusahaan Swedia bernama MySQL AB, yang pada saat itu bernama TcX Data Konsult AB sekitar tahun 1994-1995. MySQL sudah ada sejak 1979. MySQL termasuk jenis *Relation Database Management System* (RDBMS) digunakan oleh banyak portal-portal internet sebagai basis data dari informasi yang ditampilkan pada situs *web*. Kepopuleran MySQL dimungkinkan karena kemudahannya untuk digunakan, cepat secara kinerja *query*, dan mencukupi untuk kebutuhan basis data perusahaan-perusahaan skala menengah dan kecil. Istilah seperti tabel, baris, dan kolom tetap digunakan dalam MySQL. Sebuah basis data yang terdapat pada MySQL mengandung satu atau beberapa tabel yang terdiri dari sejumlah baris dan kolom.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah suatu bahasa (*language*) yang digunakan untuk mengakses data di dalam sebuah database relasional. SQL sering juga disebut

dengan istilah query, dan bahasa *SQL* secara praktiknya digunakan sebagai bahasa standar untuk manajemen database relasional. Hingga saat ini hampir seluruh server database atau software database mengenal dan mengerti bahasa *SQL*. Dalam penggunaan *SQL* terdapat beberapa perintah yang berguna untuk mengakses dan memanajemen data yang terdapat dalam database. Secara garis besar, *SQL Server* mempunyai 3 (Tiga) jenis perintah *SQL* yaitu :

1.) *Data Definition Language (DDL)*

DDL adalah sub perintah dari bahasa SQL yang digunakan untuk membangun kerangka sebuah database, dalam hal ini database dan table. Terdapat tiga perintah penting dalam DDL, yaitu :

- a. CREATE: perintah ini digunakan untuk membuat, termasuk di dalamnya membuat database baru, tabel baru view baru, dan kolom baru. Contoh: CREATE DATABASE nama_database.
- b. ALTER: perintah ALTER berfungsi untuk mengubah struktur tabel yang telah dibuat. Mencakup di dalamnya mengubah nama tabel, menambah kolom, mengubah kolom, menghapus kolom, dan memberikan atribut pada kolom. Contoh: ALTER TABLE nama_tabel ADD nama_kolom datatype.
- c. DROP: perintah DROP berfungsi untuk menghapus database atau tabel. Contoh: DROP DATABASE nama_database.

2) *Data Manipulation Language (DML)*

DML adalah sub perintah dari bahasa SQL yang digunakan untuk memanipulasi data dalam database yang telah dibuat. Terdapat 4 (Empat) perintah penting dalam DML, yaitu :

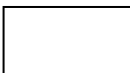

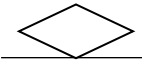

- a. INSERT: perintah ini digunakan untuk memasukkan data baru ke dalam sebuah tabel. Perintah ini tentu saja bisa dijalankan ketika database dan tabel sudah dibuat. Contoh: INSERT INTO nama_tabel VALUES (data1, data2, dst...);
- b. SELECT: perintah ini digunakan untuk mengambil dan menampilkan data dari tabel atau bahkan dari beberapa tabel dengan penggunaan relasi. Contoh: SELECT nama_kolom1, nama_kolom2 FROM nama_tabel;
- c. UPDATE: perintah update digunakan untuk memperbaharui data pada sebuah tabel. Contoh: UPDATE nama_tabel SET kolom1=data1, kolom2=data2,... WHERE kolom=data;

- d. DELETE: perintah delete digunakan untuk menghapus data dari sebuah tabel. Contoh: DELETE FROM nama_tabel WHERE kolom=data;

A. *Entity Relationship Diagram (ERD)*




ERD adalah suatu rancangan atau bentuk hubungan suatu kegiatan di dalam sistem yang berkaitan langsung dan mempunyai fungsi di dalam proses tersebut. ERD adalah suatu pemodelan dari basis data relasional yang didasarkan atas persepsi di dalam dunia nyata, dunia ini senantiasa terdiri dari sekumpulan objek yang saling berhubungan antara satu dengan yang lainnya. Suatu objek disebut *entity* dan hubpeungan yang dimilikinya disebut *relationship*. Suatu *entity* bersifat unik dan memiliki atribut sebagai pembeda dengan entity lainnya [14]. Berikut merupakan simbol-simbol dari ERD dapat dilihat pada Tabel 2.5 :

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.	Link		Garis sebagai penghubung antara himpunan, relasi, dan himpunan entitas dengan atributnya

ERD memiliki derajat relasi atau biasa disebut kardinalitas. Kardinalitas menjelaskan batasan jumlah keterhubungan satu entity dengan entity lainnya seperti pada Tabel 2.6.

Tabel 2.6 Macam-Macam Kardinalitas

No.	Simbol	Nama	Keterangan
1.		Relasi Satu ke Satu (<i>One to One</i>)	Relasi yang menunjukkan bahwa setiap himpunan entitas berhubungan dengan tepat satu himpunan entitas lainnya
2.		Relasi Satu ke Banyak (<i>One to Many</i>)	Relasi yang menunjukkan bahwa hubungan antara entitas pertama dengan entitas kedua adalah satu banding banyak, begitu pula sebaliknya
3.		Relasi Banyak ke Banyak (<i>Banyak to Many</i>)	Relasi yang menunjukkan bahwa setiap himpunan entitas boleh berhubungan dengan banyak himpunan entitas lainnya dan sebaliknya

2.1.5 Registrasi dan Heregistrasi

Registrasi yaitu calon mahasiswa yang telah dinyatakan diterima sebagai mahasiswa baru wajib melakukan registrasi sesuai dengan jadwal dan ketentuan yang telah ditetapkan. Calon mahasiswa yang telah melaksanakan registrasi, memiliki hak penuh sebagai mahasiswa, yaitu hak untuk mengikuti semua kegiatan akademik dan menggunakan semua fasilitas yang disediakan bagi mahasiswa. Calon mahasiswa yang tidak melaksanakan registrasi sesuai dengan waktu yang ditentukan, hak sebagai mahasiswa dinyatakan gugur.

Heregistrasi yaitu setiap awal semester gasal, mahasiswa yang akan aktif mengikuti kegiatan akademik wajib melakukan heregistrasi sesuai dengan jadwal dan ketentuan yang telah ditetapkan. Mahasiswa yang telah melakukan heregistrasi, memiliki hak penuh sebagai mahasiswa yaitu hak untuk mengikuti semua kegiatan akademik dan menggunakan semua fasilitas bagi mahasiswa yang disediakan, sebagaimana ketentuan yang berlaku pada semester yang berjalan.