

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Penelitian yang dilakukan oleh peneliti lain yang berkaitan dengan penelitian ini dijadikan sebagai bahan pertimbangan, seperti pada penelitian yang dilakukan oleh Ketut Jaya Atmaja dan Ni Kade Ayu Nirwana pada tahun 2020 membahas mengenai pemberian fasilitas bidang akademik yang salah satunya dilakukan oleh Biro Administrasi dan Akademik (BAAK). Pelayanan administrasi ini masih dilakukan dengan datang langsung ke BAAK yang kemudian akan dicatat dan diproses sesuai dengan pengajuan dari mahasiswa sehingga menimbulkan beberapa kendala seperti BAAK yang harus melayani dan mencatat kebutuhan mahasiswa satu persatu sehingga kurang efisien dalam waktu, terutama jika BAAK sedang mengerjakan tugas administrasi atau layanan lain yang memiliki tingkat urgensi lebih tinggi. Dari sisi mahasiswa hal ini kurang efisien dalam waktu dan mahasiswa sendiri tidak dapat memantau perkembangan proses pengajuan. Hasil dari penelitian ini terbentuklah sistem yang terkomputerisasi yang dapat mengakomodir proses pengajuan pindah kelas, semester pendek, surat pindah kampus, surat cuti, surat masih kuliah dan surat pengantar survey dalam lingkungan STMIK STIKOM sehingga membantu BAAK dalam hal merekap pengajuan yang dilakukan mahasiswa[4].

Penelitian selanjutnya dilakukan oleh Fatoni pada tahun 2020 membahas bagaimana seorang karyawan dan dosen harus melakukan serangkaian alur proses pengajuan cuti dan izin dengan menyampaikan pengajuan tersebut secara tertulis menggunakan formulir permohonan cuti. Dalam penerapan cara manual ini memiliki beberapa kelemahan yaitu kurang efisien, dan membutuhkan waktu yang lama karena belum terkomputerisasi. Berdasarkan kondisi tersebut diperlukan sebuah sistem yang dapat membantu aktifitas pengaturan permohonan cuti dan izin staf karyawan dan dosen. Metode yang digunakan dalam pengembangan sistem pada penelitian ini adalah metode *Rapid Application Development* (RAD). Hasil dari penelitian yang telah dilakukan ini berupa sistem Informasi pengajuan cuti dan izin berbasis web yang dapat digunakan oleh staf karyawan, dosen dan pimpinan di lingkungan Universitas Bina Darma dalam pengajuan cuti atau izin. Staf karyawan, dosen dan pimpinan mempunyai hak yang sama dalam proses permohonan cuti di sistem ini [5].

Penelitian selanjutnya dilakukan oleh Nurhayati pada tahun 2020 membahas mengenai *website* yang dimiliki oleh kampus tidak mencakup keseluruhan kegiatan akademik dan administrasi mahasiswa. Sebagai perguruan tinggi, sebaiknya sistem Informasi yang dimiliki memuat seluruh Informasi yang juga mencakup keseluruhan kegiatan akademik dan administrasi mahasiswa untuk menunjang kegiatan akademik maupun non-akademik mahasiswa. Informasi yang tersedia meliputi profil kampus, kegiatan akademik secara umum, forum mahasiswa dan alumni belum mencakup dalam Informasi yang spesifik bagi mahasiswa yang sudah terdaftar, bahkan masih banyak yang manual. Apalagi tidak semua mahasiswa meluangkan waktu untuk mengunjungi *website*, bahkan ada kemungkinan mahasiswa tidak mengetahui cara mengakses *website* kampus untuk melihat Informasi terbaru. Penelitian ini telah menghasilkan aplikasi yang memiliki fitur untuk membantu dalam proses pelayanan administrasi mahasiswa ditambah dengan adanya fitur SMS Gateway yang memberikan Informasi kepada mahasiswa terkait Informasi kegiatan akademik dan juga administrasi mahasiswa. Dengan adanya sistem ini dapat mengurangi waktu tunggu mahasiswa dalam pelayanan akademik dan mengurangi panjangnya antrean pelayanan [6].

Penelitian selanjutnya dilakukan oleh Hidayat Chandra, Tajuddin Abdillah dan Roviana pada tahun 2021 membahas mengenai pengelolaan administrasi kemahasiswaan. Administrasi kemahasiswaan berfungsi untuk melayani segala keperluan administrasi yang berkaitan dengan mahasiswa. Administrasi kemahasiswaan terdiri dari surat keterangan aktif kuliah, surat rekomendasi, surat tugas, surat dispensasi, surat persetujuan pindah, surat pengantar cuti, surat pengantar beasiswa, surat keterangan lulus, surat rekomendasi penelitian, surat keterangan bebas perlengkapan dan surat keterangan bebas perpustakaan. Dalam proses pelayanan administrasi mahasiswa, Fakultas Teknik Universitas Gorontalo memiliki beberapa kendala diantaranya yaitu tidak adanya Informasi mengenai *progress* pengurusan administrasi kemahasiswaan sehingga mahasiswa tidak mengetahui pengurusan administrasi sudah selesai atau belum. Selain itu dalam pembuatan hasil laporan agak terlambat karena pada saat pencarian data membutuhkan waktu yang agak lama. Hasil penelitian ini adalah sistem pengelolaan administrasi mahasiswa menggunakan metode *prototype* yang membantu proses pengurusan dan pembuatan laporan administrasi menjadi lebih mudah, selain itu juga membantu mahasiswa untuk memonitoring proses pengurusan administrasi yang diajukan [7].

Penelitian selanjutnya dilakukan oleh Sutriyanto pada tahun 2021 yang membahas pengajuan pengunduran diri mahasiswa di Politeknik Negeri Cilacap. Proses pengajuan pengunduran yang dilakukan oleh mahasiswa masih dilakukan dengan cara manual dengan mengambil formulir di BAAK kemudian mengisi dan menandatangani formulir serta meminta orangtua, dosen wali dan ketua jurusan untuk turut serta menandatangani. Proses ini membutuhkan waktu yang cukup lama oleh karena itu dilakukan penelitian ini untuk mempercepat proses pengajuan pengunduran diri dengan membuat sistem menggunakan metode *prototyping*. Sistem ini mengelola pengajuan mahasiswa dan mencetak surat permohonan yang diajukan oleh mahasiswa. Penelitian ini menghasilkan sistem pengajuan pengunduran diri mahasiswa yang dapat membantu dan mempercepat proses pengunduran diri mahasiswa [8].

Pada penelitian ini, peneliti bermaksud mengembangkan Sistem Informasi Pengunduran Diri Mahasiswa Politeknik Negeri Cilacap yang menangani proses pengunduran diri mahasiswa. Sistem ini akan dibangun dengan berbasis *website* menggunakan bahasa pemrograman PHP native dan metode pengembangan sistem metode SDLC dengan model *waterfall*. Alur kerja sistem dimulai dengan mahasiswa yang melakukan pengajuan permohonan pengunduran diri kemudian menunggu dosen wali, bagian perpustakaan, bagian keuangan dan ketua jurusan untuk mengkonfirmasi. Sistem ini akan dilengkapi dengan tambahan fitur pengelolaan data master seperti data mahasiswa, data jurusan dan data pegawai, pengelolaan surat permohonan, pengelolaan surat keputusan dan transkrip nilai, pengarsipan laporan serta surat, monitoring dan verifikasi pengajuan serta notifikasi WhatsApp yang berguna untuk memberikan notifikasi sistem terkait pengajuan pengunduran diri mahasiswa kepada pihak yang terlibat sehingga pihak tersebut tidak harus berulang kali membuka sistem untuk bisa melihat informasi di dalamnya. Dengan adanya sistem ini akan mempermudah mahasiswa dalam memonitoring pengajuan pengunduran diri sehingga membuat proses pengajuan menjadi lebih efisien. Selain itu, kelebihan sistem ini adalah menampilkan surat keputusan dan transkrip nilai mahasiswa selama perkuliahan sehingga mahasiswa tidak perlu datang ke kampus lagi untuk mengambilnya. Berikut adalah tabel penelitian-penelitian terdahulu:

**Tabel 2. 1** Penelitian Terdahulu

No	Penulis dan Tahun	Hasil Penelitian	Perbedaan	Fitur Sistem
1	Ketut Jaya dan Ni Kade Ayu Nirwana (2020)	Sistem Informasi Manajemen Layanan Akademik di STMIK STIKOM Indonesia berbasis Website	1. Sistem melayani pengajuan cuti	1. Pengajuan pindah kampus 2. <i>History</i> proses pengajuan
2	Fatoni (2020)	Sistem Informasi Cuti dan Izin	1. Sistem melayani pengajuan cuti dan izin 2. Metode pengembangan RAD 3. Bahasa pemrograman <i>python</i> 4. Sistem melayani pengajuan cuti dan Izin karyawan	1. Pengajuan cuti 2. Cek status pengajuan
3	Nurhayati (2020)	Sistem Informasi Akademik	1. Melayani proses cuti mahasiswa 2. Perancangan sistem UML	1. Mengelola data jurusan 2. Melihat dan transkrip nilai 3. Notifikasi sistem SMS Gateway

4	Hidayat, Tajudin dan Roviana (2021)	Rancang Bangun Sistem Pengelolaan Administrasi Kemahasiswaan	<ol style="list-style-type: none"> <li>1. Metode pengembangan <i>prototype</i></li> <li>2. Perancangan sistem UML</li> </ol>	<ol style="list-style-type: none"> <li>1. Pengelolaan data mahasiswa</li> <li>2. Pengelolaan pengajuan surat pindah</li> <li>3. Pengelolaan data jurusan</li> <li>4. Monitoring proses pengajuan</li> </ol>
5	Sutriyanto (2021)	Sistem Pengunduran Diri Mahasiswa	<ol style="list-style-type: none"> <li>1. Metode pengembangan <i>prototype</i></li> <li>2. Perancangan sistem UML</li> </ol>	<ol style="list-style-type: none"> <li>1. Kelola data mahasiswa</li> <li>2. Kelola data wali dosen</li> <li>3. Kelola data ketua jurusan</li> <li>4. Kelola data kelas</li> <li>5. Tambah pengajuan</li> <li>6. Verifikasi pengajuan</li> <li>7. Cetak surat pengajuan</li> </ol>

## 2.2 Landasan Teori

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

### 2.2.1. Sistem Informasi

Sistem Informasi adalah suatu komponen yang bekerja sama untuk mengumpulkan, mengolah, menyimpan serta menyebarkan Informasi. Menurut O'Brien dan Marakas komponen sistem Informasi terdiri dari (*input*), memproses data menjadi Informasi, hasil (*output*) dan penyimpanan data. Sistem Informasi dapat diartikan sebagai serangkaian prosedur dimana data dikumpulkan, diproses menjadi Informasi dan didistribusikan kepada pengguna. Data yang disampaikan harus relevan

(*relevance*), tepat waktu (*timeliness*) dan akurat (*accurate*) agar dikatakan sebagai Informasi yang valid [9].

### **2.2.2. Pengunduran Diri**

Pengunduran diri adalah penarikan diri atas seseorang secara tetap dari pekerjaannya negara dalam berbagai bidang karena lanjut usia, ketidakmampuan, sakit dan lain sebagainya [10]. Pengunduran diri mahasiswa merupakan penarikan diri atau mundurnya mahasiswa untuk tidak melanjutkan studi belajarnya dikarenakan berbagai macam alasan dan faktor yang mengharuskan mahasiswa untuk keluar dari perguruan tinggi. Dengan kata lain, pengunduran diri dapat diartikan sebagai pemberitahuan secara sukarela oleh mahasiswa bahwa pihak yang bersangkutan berniat untuk berhenti dan mengakhiri masa studinya.

*Standard Operating Procedure* (SOP) alur pengajuan pengunduran diri yang ditetapkan oleh Bagian Administrasi Akademik dan Kemahasiswaan Politeknik Negeri Cilacap adalah meminta formulir pengunduran diri kepada BAAK, kemudian mengisi formulir pengajuan pengunduran diri mahasiswa dan meminta persetujuan berupa tanda tangan orang tua yang disertai dengan materai, tanda tangan dosen wali, bagian perpustakaan, bagian keuangan dan ketua jurusan serta menunjukkan bukti bebas tanggungan keuangan dan peminjaman buku dari perpustakaan kemudian mengembalikan formulir pengunduran diri ke Bagian Administrasi Akademik dan Keuangan untuk di proses dan dibuatkan surat keputusan pengunduran diri. Jika surat keputusan pengunduran diri telah di setujui oleh direktur maka mahasiswa tersebut sudah resmi keluar dari Politeknik Negeri Cilacap.

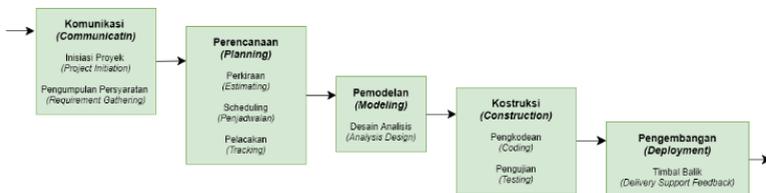
### **2.2.3. Notifikasi**

Notifikasi merupakan penyampaian pesan atau Informasi secara singkat melalui sebuah media. Sistem notifikasi digunakan sebagai sarana untuk penyampaian Informasi dari sebuah sistem kepada pengguna. Salah satu media notifikasi yang saat ini banyak digunakan adalah WhatsApp. Sebagai *platform* instan, WhatsApp dimanfaatkan sebagai program untuk menjalin komunikasi secara efektif antar pengguna. WhatsApp *Gateway* merupakan sistem aplikasi yang digunakan untuk mengirim pesan WhatsApp dari web atau perangkat lain ke perangkat WhatsApp. Penggunaan WhatsApp *Gateway* pada sistem ini digunakan sebagai notifikasi pengajuan pengunduran diri mahasiswa, sehingga pengguna akan langsung mengetahui jika ada mahasiswa yang melakukan

pengunduran diri dan melalui notifikasi ini mahasiswa dapat dengan mudah mengetahui dan memonitoring proses pengajuan yang sedang diajukan [11].

#### 2.2.4. Model Waterfall

Metode SDLC (*System Development Life Cycle*) yang digunakan dalam pengembangan sistem pada penelitian ini adalah model *waterfall* menurut Roger Pressman. Menurut Roger S. Pressman model *waterfall* sering dinamakan siklus hidup klasik, dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak. Metode SDLC model *waterfall* sangat cocok digunakan untuk pengembangan perangkat lunak yang jarang berubah-ubah. Model ini juga cenderung dipilih karena pendekatannya yang sistematis dan terurut dalam membangun sebuah perangkat lunak. Model *waterfall* dimulai dari tahap komunikasi, perencanaan, pemodelan, konstruksi dan terakhir penerapan. Model ini dinamakan *waterfall* karena setiap tahap harus dilakukan secara berurutan dan harus menunggu tahap sebelumnya selesai untuk bisa berlanjut pada tahap selanjutnya [3].



**Gambar 2. 1** Model *Waterfall* (Pressman, 2015)

Tahapan model *waterfall* yang dilakukan menurut Roger S. Pressman (2015) yaitu [3]:

1. Komunikasi (*communication*)

Komunikasi dengan pengguna merupakan langkah awal dari model *waterfall*. Komunikasi ini merupakan langkah yang penting karena berkaitan dengan pengumpulan Informasi tentang kebutuhan pengguna. Tahap ini juga dilakukan inisiasi proyek seperti menganalisis masalah yang ada dan tujuan yang akan dicapai.

2. Perencanaan (*planning*)

Tahap selanjutnya yaitu penetapan rencana yang akan dilakukan untuk pengerjaan sistem. Proses yang dilakukan yaitu memperkirakan kebutuhan-kebutuhan apa saja yang dibutuhkan

oleh sistem dan proses penjadwalan untuk pengerjaan sistem.

3. **Pemodelan (*modeling*)**

Tahap pemodelan merupakan tahap menerjemahkan kebutuhan pengguna ke dalam perancangan sistem sebelum memasuki tahap pengkodean. Proses ini berfokus pada perancangan struktur data, representasi antarmuka dan algoritma prosedural. Tahap ini mentranslasi kebutuhan sistem dari tahap kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

4. **Konstruksi (*construction*)**

Tahap *coding* merupakan penerjemahan desain dalam bahasa yang dipahami oleh komputer. Pada tahap ini penggunaan komputer akan dimaksimalkan dalam pengerjaan perangkat lunak. Hasil dari tahap ini adalah program yang diciptakan sesuai dengan desain yang telah dibuat pada tahap pemodelan sistem. Setelah tahap pengkodean selesai, maka dilakukan proses *testing* terhadap sistem yang sudah selesai dibuat. *Testing* ini dilakukan untuk mengetahui kesalahan-kesalahan yang muncul pada sistem untuk kemudian dilakukan perbaikan sebelum di serahkan kepada pengguna sehingga dapat meminimalisir terjadinya *error* dan keluaran yang tidak diharapkan

5. **Penerapan (*deployment*)**

Tahap ini dapat dikatakan sebagai tahap final dalam model *waterfall* yaitu pengimplementasian sistem kepada pengguna atau *client*. Setelah melalui tahapan-tahapan tersebut sistem akan diserahkan kepada pengguna yang selanjutnya harus dilakukan pemeliharaan secara berkala terhadap sistem agar dapat berjalan dan berkembang sesuai fungsinya.

### **2.2.5. Pengujian (*Testing*)**

Pengujian sistem dilakukan untuk mengetahui kesalahan yang muncul ketika program selesai dibuat untuk selanjutnya dapat diperbaiki. Pengujian ini secara fungsional dilakukan untuk meminimalkan kesalahan dan memastikan bahwa semua bagian lulus pengujian dan hasil *output* sesuai dengan keinginan pengguna. Pengujian yang dilakukan pada penelitian ini adalah pengujian black-box untuk memperlihatkan fungsi perangkat lunak dapat beroperasi dengan baik [12].

Pengujian *black-box* didasarkan pada detail dari sistem seperti antarmuka, fungsi dan kesesuaian alur dengan permintaan pengguna. Berbeda dengan pengujian *white-box* yang menguji kode program,

pengujian *black-box* tidak dapat menguji bahkan melihat kode program. Selain itu, *black-box testing* juga menguji *input* serta *output* yang dihasilkan dengan menggunakan semua persyaratan fungsional suatu program. Tidak seperti pengujian *white-box*, yang dilakukan pada saat awal proses pengujian, pengujian *black-box* cenderung diaplikasikan pada tahap akhir dikarena pengujian *black-box* memperhatikan struktur - struktur kontrol, sehingga perhatian berfokus pada domain Informasi. Pengujian *black-box* dapat menemukan kesalahan dengan menggunakan kategori sebagai berikut [12]:

1. Fungsi yang tidak benar atau hilang
2. Kesalahan antarmuka
3. Kesalahan struktur data atau akses *database* eksternal
4. Kesalahan kinerja
5. Kesalahan inisialisasi ataupun terminal

Pengujian *black-box* lebih baik dilakukan pada tahap akhir. Penguji harus merancang skenario pengujian yang dapat memberitahu mengenai ada atau tidaknya kesalahan melalui beberapa proses yang ada, diantaranya yaitu [12]:

1. Menganalisa kebutuhan dan spesifikasi dari perangkat lunak
2. Pemilihan jenis *input* yang menghasilkan *output* dengan benar serta *input* yang mungkin menghasilkan *output* yang salah
3. Menentukan *output* untuk setiap *input-an*
4. Pengujian dilakukan dengan *input-input* yang benar-benar telah diseleksi
5. Melakukan pengujian
6. Perbandingan *output* yang dihasilkan dengan *output* yang diharapkan
7. Menentukan fungsional yang seharusnya ada pada perangkat lunak

#### **2.2.6. Basis Data**

Basis data atau *database* adalah kumpulan data yang saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi. Dengan kata lain, *database* merupakan kumpulan file yang mempunyai kaitan antara satu dengan yang lain sehingga membentuk satu bangun data untuk mengInformasikan dalam bahasa tertentu [13].

*Database* membuat penyimpanan dan pengelolaan data menjadi lebih efisien. *Database* berwujud tabel yang terdiri dari kolom dan baris yang memuat atribut dan nilai tertentu. Banyaknya jumlah kolom dan baris dalam suatu *database* tergantung pada jumlah kategori atau jenis

Informasi yang perlu disimpan. Fungsi *database* adalah untuk menghindari data ganda yang tersimpan. Suatu *database management system* (DBMS) dapat diatur supaya bisa mengenali duplikasi data ketika diinput. Selain itu, *database* juga berfungsi untuk memudahkan proses akses, menyimpan, pembaharuan, penghapusan data, pengelompokan data dan menjadi alternatif terkait masalah penyimpanan ruang dalam suatu aplikasi. DBMS adalah perangkat lunak yang memungkinkan pemakai untuk mendefinisikan, mengelola dan mengontrol akses ke basis data. DBMS yang mengelola basis data *relational* disebut dengan *Relational DBMS* (RDMS) [13].

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*. Kepopuleran MySQL dimungkinkan karena kemudahannya untuk digunakan, cepat secara kinerja *query*, dan mencukupi untuk kebutuhan basis data perusahaan-perusahaan skala menengah dan kecil. Dalam penggunaan *SQL* terdapat beberapa perintah yang berguna untuk mengakses dan manajemen data yang terdapat dalam *database*. Secara garis besar, *SQL Server* mempunyai 3 (Tiga) jenis perintah *SQL*, yaitu [13]:

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

Sub perintah dari Bahasa SQL yang digunakan untuk membangun kerangka sebuah *database*. Dalam Bahasa ini dapat membuat tabel baru, membuat indeks, menentukan struktur dan penyimpanan tabel. Berikut adalah perintah yang digunakan dalam DDL, diantaranya [14]:

##### 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 ini 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 datatype.

c. *DROP*

Perintah ini berfungsi untuk menghapus *database* atau tabel.  
Contoh: *DROP DATABASE* nama\_database.

2. *Data Manipulation Language (DML)*

Bahasa yang berguna untuk melakukan manipulasi data pada suatu basis data. Manipulasi dapat berupa penambahan, penghapusan dan pengubahan data pada suatu basis data. Berikut adalah perintah yang digunakan dalam DML, diantaranya yaitu [14]:

a. *INSERT*

Perintah ini digunakan untuk memasukan data baru ke dalam sebuah tabel. Perintah ini tentu dapat 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 menggunakan relasi. Contoh: *SELECT* nama\_kolom1, nama\_kolom2 *FROM* nama\_tabel;

c. *UPDATE*

Perintah ini digunakan untuk memperbarui atau mengupdate data pada sebuah tabel. Contoh: *UPDATE* nama\_tabel *SET* kolom1=data1, kolom2=data2,... *WHERE* kolom=data;

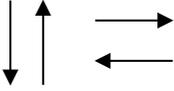
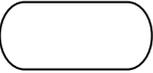
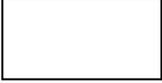
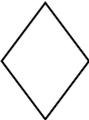
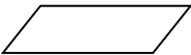
d. *DELETE*

Perintah ini digunakan untuk menghapus data dari sebuah tabel.  
Contoh: *DELETE FROM* nama\_tabel *WHERE* kolom=data;

### 2.2.7. Flowchart

*Flowchart* adalah bagan (*chart*) yang menunjukkan alir (*flow*) yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau panah. *Flowchart* berperan penting dalam pemutusan langkah dari pembuatan program agar lebih mudah untuk dipahami [15]. Simbol-simbol dalam *flowchart* dapat dilihat pada tabel 2.2 berikut:

Tabel 2. 2 Simbol Flowchart

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

### 2.2.8. Data Flow Diagram (DFD)

Data Flow Diagram merupakan gambaran perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi yang dapat digunakan untuk penggambaran analisa maupun rancangan sistem. Diagram ini menggambarkan secara rinci tentang proses yang terdapat pada diagram jenjang. Penggunaan DFD ini membantu untuk memahami sistem secara terstruktur dan jelas. Dalam DFD juga menggambarkan penyimpanan data serta proses mentransformulasikan data [16].

Fungsi dari adanya DFD adalah pemodelan yang mungkin untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. Menurut Pressman DFD memperlihatkan gambaran tentang masukan – proses – keluaran dari sebuah sistem. Dalam DFD terdapat 3 level yang harus diketahui, yaitu [16]:

#### 1. Diagram Konteks

Diagram konteks adalah sebuah diagram yang menggambarkan seluruh proses yang terdapat dalam sistem. Diagram konteks ini merupakan level tertinggi dalam DFD sehingga sebelum DFD dibuat, dibutuhkan diagram konteks. Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran-aliran data utama menuju dan dari sistem. Diagram ini sama sekali tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan. Diagram konteks hanya menggunakan 3 simbol, yaitu entitas eksternal, aliran data dan proses.

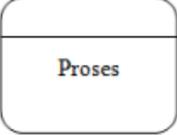
#### 2. Diagram Level 1

Diagram level 1 merupakan tahap lanjutan dari diagram 0, dimana pada diagram ini semua proses yang ada pada diagram konteks akan dirinci dengan lengkap sehingga lebih detail. Proses utama akan dipecah menjadi sub-proses sesuai dengan fungsinya masing-masing. Pada DFD level 1 ini akan digambarkan aliran data secara lebih kompleks di setiap prosesnya hingga terbentuklah *datastore* dan aliran data.

#### 3. Diagram Level 2

Diagram level 2 merupakan diagram aliran data yang mana menjadi rincian proses dari DFD level 1 yang tugasnya adalah menguraikan proses-proses yang ada dalam lingkup sistem. Setiap proses yang ada pada DFD level 1 dapat diperinci menjadi sebuah DFD lagi. Proses DFD nantinya akan berhenti hingga cukup mendetail. Simbol-simbol dalam DFD dapat dilihat pada tabel 2.3

**Tabel 2. 3** Simbol *Data Flow Diagram*

No	Simbol	Fungsi
1	 <b>Entitas Eksternal</b>	Simbol yang menunjukkan entitas luar yang berinteraksi langsung dengan sistem bisa dalam bentuk memberikan data ke sistem, menerima data dari sistem ataupun keduanya
2	 <b>Proses</b>	Simbol yang digunakan untuk menunjukkan kegiatan yang dilakukan oleh pengguna, komputer dari hasil suatu arus yang masuk ke dalam proses untuk dihasilkan arus data yang keluar dari proses atau untuk mengubah input menjadi output.
3	 <b>Data Store</b>	Simbol yang digunakan untuk menunjukkan simpanan data yang dapat berupa file atau <i>database</i> pada sebuah sistem
4	 <b>Alur Data</b>	Simbol yang menunjukkan data mengalir melalui sistem, dimulai dengan sebgaiain input dan diubah atau diproses menjadi output.

### 2.2.9. Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* atau diagram hubungan *entitas* merupakan diagram yang digunakan untuk perancangan suatu *database* dan menunjukan relasi antar objek atau entitas beserta atribut-atributnya secara detail. Dengan menggunakan ERD, sistem *database* yang sedang

dibentuk dapat digambarkan dengan lebih terstruktur dan rapi. Selain itu, membantu dalam menganalisis suatu *database* dengan lebih cepat dan mampu menjalankan relasi antar data yang mempunyai keterkaitan dengan objek yang dihubungkan dengan suatu relasi khusus [14].

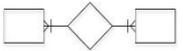
*Entity Relationship Diagram* digunakan untuk memodelkan struktur data serta hubungan antar data dan untuk menggambarkannya digunakan beberapa notasi serta simbol. Terdapat banyak komponen yang dimiliki oleh ERD, di mana komponen ini akan melakukan visualisasi pada dua Informasi yang dianggap penting. Informasi pertama merupakan entitas utama yang terdapat di dalam ruang lingkup suatu sistem, sedangkan Informasi kedua merupakan hubungan yang terdapat di berbagai entitas itu sendiri [14]. Simbol dalam ERD dapat dilihat pada tabel 2.4 berikut

**Tabel 2. 4 Entity Relationship Diagram**

No.	Nama	Simbol	Keterangan
1.	Entitas		Suatu kumpulan objek atau sesuatu yang dapat dibedakan atau dapat didefinisikan secara unik
2.	Atribut		Atribut merupakan Informasi yang diambil tentang sebuah entitas.
3.	Relasi		Himpunan relasi merupakan hubungan yang terjadi antara satu entitas atau lebih
4.	Link		Garis sebagai penghubung antara himpunan, relasi dan himpunan entitas dengan atributnya

ERD memiliki derajat relasi atau biasa disebut dengan kardinalitas. Kardinalitas menjelaskan batasan jumlah keterhubungan satu entitas dengan entitas yang lain. Kardinalitas yang terdapat pada ER-diagram dapat dilihat pada tabel 2.5 [17]

**Tabel 2. 5** Kardinalitas ER-Diagram

No.	Simbol	Nama	Keterangan
1.		Relasi Satu ke Satu ( <i>One to One</i> )	Relasi yang menunjukkan bahwa setiap entitas berhubungan dengan tepat satu 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 entitas boleh berhubungan dengan banyak entitas lainnya dan sebaiknya

**~Halaman ini sengaja dikosongkan~**