



BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian dilakukan oleh R. P Dhaniawaty dan E Susilawati tahun 2018 [3]. Tujuan sistem ini yaitu untuk membantu dosen kemahasiswaan dalam mengelola data pengurus himpunan mahasiswa sistem informasi, mengintegrasikan jadwal program kerja dan membuat laporan bulanan program kerja. Informasi yang disajikan pada sistem ini adalah program kerja beserta jadwalnya, data kepengurusan, laporan pengajuan proposal dan laporan pertanggungjawaban program kerja. Metode penelitian yang digunakan adalah menggunakan model pengembangan sistem informasi *prototype*.

Penelitian dilakukan oleh S.W Ahmad Taufiqi Muhsinin tahun 2020 [4] menggunakan metodologi waterfall. Sistem ini bertujuan untuk membantu manajemen dokumen organisasi yang meliputi pengurus organisasi, pendataan dan penjadwalan kegiatan, serta data dokumen dan inventaris. Tahapan dalam sistem ini yaitu mengumpulkan data terkait sistem, perancangan desain antarmuka aplikasi dan perancangan aplikasi dengan basis bahasa pemrograman PHP. Pada penelitian ini belum sampai pada tahap pengujian.

Penelitian lainnya juga dilakukan W Setiyaningsih tahun 2021 [5] bertujuan untuk mengoptimalkan pelaporan kinerja harian dan bulanan oleh pegawai honorer. Pelaporan kinerja tersebut harus dilaporkan kepada pimpinan. Sistem ini diharapkan dapat memudahkan dalam mengedit laporan dan sebagai arsip laporan dalam satu bulan sehingga tenggat waktu pengumpulan pelaporan dan pemantauan oleh pimpinan dapat dilaksanakan dengan lebih cepat. Metode penelitian pada penelitian ini adalah dengan menggunakan metode pengembangan *waterfall*.

Penelitian lain dilakukan oleh Budi Supriadi dan Lukman Bachtiar tahun 2018 [6] memiliki tujuan yaitu memudahkan Sekretaris dalam melakukan pengelolaan surat masuk dan surat keluar, *setting* kategori surat, *setting* pegawai dan admin serta laporan surat masuk dan laporan surat keluar. Metode-metode pengembangan sistem perangkat lunak ini sebuah web editor *Adobe Dreamweaver CS6*, bahasa pemrograman Javascript, CSS, HTML, PHP dan MySQL. Dengan

adanya sistem tersebut diharapkan dapat diperoleh hasil yang mudah dan tepat waktu dalam pencarian surat, pencatatan surat dan disposisi surat.

Penelitian dilakukan oleh Dhia Farida Ariefni dan Mercurius Broto Legowo tahun 2018 [7] bertujuan untuk menerapkan konsep *monitoring* dan evaluasi karena kegiatan mahasiswa sangat banyak dan bervariasi serta membutuhkan aspek finansial. Pembuatan sistem ini juga sebagai arsip data atau file terkait seluruh kegiatan mahasiswa guna memudahkan Kaprodi untuk mengetahui data tentang kegiatan mahasiswa tersebut.

Perbedaan dari penelitian yang dilakukan adalah dalam membuat Sistem Informasi Pengelolaan dan Pelaporan Program Kerja Himpunan Mahasiswa Teknik Informatika Politeknik Negeri Cilacap yaitu menggunakan *framework* Codeigniter(CI). Selain itu terdapat notifikasi email sehingga proses persetujuan lebih cepat dilakukan.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Pengertian sistem informasi adalah sebuah implementasi dari penerapan teknologi terkomputerisasi yang memudahkan manusia dimana pekerjaan menjadi lebih mudah sehingga dapat mencapai tujuan tertentu. Kelebihan dari sistem informasi adalah sebagai penyedia data dan penyimpanan terbesar yang dapat memungkinkan penyelesaian proses kerja lebih cepat sehingga dapat membantu menyelesaikan permasalahan yang dihadapi [8]. Sistem informasi juga dapat diartikan media yang diimplementasikan secara teknologi guna keperluan pencatatan, penyimpanan serta penyebaran aktualisasi kebahasaan dan guna mendukung pembuatan inferensi [9]. Sistem informasi dapat didefinisikan sebagai kumpulan sistem-sistem yang saling terintegrasi satu dengan yang lainnya hingga bermanfaat bagi pemakainya. Data akan tersimpan pada suatu database yang mengakibatkan data menjadi teratur dan memudahkan penggunaanya [10]. Berdasarkan referensi pengertian sistem informasi maka dapat disimpulkan bahwa sistem informasi adalah sebuah implementasi dari teknologi terkomputerisasi yang dapat mencatat, menyimpan dan menyebarkan sebuah informasi dimana fungsinya untuk memudahkan penggunaanya.

2.2.2 Rekayasa Web

Rekayasa *web* merupakan proses yang digunakan guna menciptakan pelaksanaan web berkualitas tinggi. Proses awal rekayasa web yaitu menentukan langkah penyelesaian melalui aplikasi *web*. Aplikasi web merupakan aplikasi yang diimplementasikan oleh insinyur *web* dan pengembang non teknis yang hasilnya berupa aplikasi berbasis *web* [11]. Tahapan-tahapan dalam rekayasa *web* adalah [12]:

1. *Customer communication*
Langkah paling efektif dalam memahami apa saja yang diinginkan oleh user adalah melalui komunikasi yang baik. Setelah mengetahui apa yang diinginkan oleh user maka dapat dibuat rumusan masalah terhadap sistem yang akan dibangun (*formulation*). Melakukan pengembangan sistem sesuai dengan kebutuhan memerlukan perjanjian dengan user dapat dilakukan setelah merumuskan masalah (*negatitation*). Kemudian observasi terhadap sistem yang akan dibangun (*elicitation*).
2. *Planning*
Perencanaan teknis dan wawancara merupakan hasil dari kebutuhan dan informasi yang diperoleh dari user. Mengidentifikasi perangkat lunak dan perangkat keras yang dibutuhkan adalah bagian dari perencanaan teknis. Perkiraan biaya pada sistem yang akan dibangun (*estimation*), menganalisis resiko oleh pengembang sistem (*risk analysis*), membuat jadwal sistem (*scheduling*) dan memantau sistem yang sedang dikembangkan (*monitoring*).
3. *Modeling*
Perancangan tampilan halaman dengan memperhatikan jenis teks dan warna sehingga dapat disesuaikan dengan alur kerja sistem. Selain merancang tampilan, pada tahap ini juga menyelesaikan masalah antara user dan fungsinya didalam sistem.
4. *Implementation(Coding)*
Hasil perancangan aktivitas dituangkan dalam bentuk html untuk halaman web dan bentuk PHP untuk fungsi logika.
5. *Testing*
Kesalahan-kesalahan yang mungkin terjadi pada sistem dapat diketahui pada tahap ini. Masalah yang dapat dideteksi mulai dari kesalahan tampilan hingga fungsi sistem.
6. *Deployment*

Penilaian serta evaluasi terhadap sistem dan pengembang merupakan tahap akhir dari rekayasa web. Menyebarkan kuisioner kepada user untuk mendapat penilaian dari setiap kriteria yang ada didalamnya.

2.2.3 Rekayasa Perangkat Lunak

Pengertian rekayasa perangkat lunak merupakan satu bidang profesi yang mendalami cara-cara pengembangan aplikasi termasuk didalamnya pembuatan, pemeliharaan, manajemen organisasi pengembangan aplikasi dan sebagainya[13]. Definisi lain rekayasa perangkat lunak adalah proses kegiatan pengembangan aplikasi itu sendiri menggunakan perintah program sebuah personal komputer sehingga aplikasi bisa bekerja lebih efisien dan efektif untuk pengguna [14].






2.2.3.1 UML (*United Modelling Language*)

UML adalah sebuah model perancangan sistem yang memiliki kelebihan dapat memudahkan developer sistem pada saat merancang sistem yang akan dibentuk karena sifatnya yang berorientasikan pada objek[15]. Terdapat beberapa diagram UML[16]: sebutkan

2.2.3.1.1 Use case

Use case merupakan gambaran berdasarkan fungsionalitas yang dibutuhkan berdasarkan sebuah sistem dan mempresentasikan sebuah sistem dan mempresentasikan sebuah hubungan antara *actor* dan sistem. Didalam *use case*, aktor digambarkan sebagai sebuah entitas berdasarkan manusia atau sebuah sistem yang melakukan pekerjaan pada sistem tersebut. Simbol-simbol pada *use case diagram* terdapat pada tabel 2.1:

Tabel 2. 1 Use Case

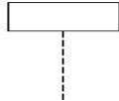

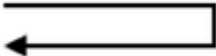


No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Menspesifikasikan himpunan tokoh yang pengguna mainkan saat berhubungan menggunakan <i>use case</i> .
2.		<i>Include</i>	<i>Use case</i> merupakan sumber secara eksplisit.
3.		<i>Extend</i>	<i>Use case</i> memperluas perilaku sumber pada titik yang diberikan.
4.		<i>Association</i>	Berguna untuk menghubungkan objek yang satu dengan objek yang lain
5.		<i>Use Case</i>	Deskripsi berdasarkan urutan aksi-aksi yang ditampilkan sistem yang membuat suatu output terukur bagi sebuah aktor



2.2.3.1.2 Sequence Diagram

Sequence diagram merupakan menggambarkan hubungan antar objek didalam dan pada kurang lebih sistem yang berupa *message* yang

digambarkan terhadap waktu. Pengertian lain dari *sequence diagram* merupakan grafik dua dimensi dimana objek ditunjukkan pada dimensi *horizontal* sedangkan *lifeline* ditunjukkan pada dimensi *vertical* [17]. Adapun simbol-simbol dari *sequence diagram* pada tabel 2.2:

Tabel 2. 2 *Sequence Diagram*



No	Gambar	Nama	Keterangan
		<i>LifeLine</i>	Objek antarmuka yang saling berhubungan.
		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat berita-berita tentang aktivitas yang terjadi.
		<i>Message (Return)</i>	Penggambaran hasil dari pengiriman pesan.
		<i>Message (return)</i>	Menyatakan arah kembali pesan.
		<i>Activation</i>	Menyatakan objek dalam keadaan aktif.





		<i>Message(Destroy)</i>	Pesan yang mewakili permintaan untuk menghancurkan siklus hidup target.
		<i>Actor</i>	Menspesifikasikan himpunan peran pengguna ketika berinteraksi dengan sistem.

2.2.3.1.3 Class Diagram

Class diagram adalah hubungan antar class yang sudah terdapat nama *class*, *attribute* dan *method* [18]. *Class diagram* memberikan gambaran interaksi antara tabel-tabel yang ada dalam *database*. Masing-masing *class* mempunyai atribut dan cara atau fungsi berdasarkan proses yang terjadi. Simbol-simbol pada *class diagram* antara lain pada tabel 2.3:

Tabel 2.3 *Class Diagram*

No	Gambar	Nama	Keterangan
1.		Operasi	Kelas pada struktur sistem.
2.		Antarmuka/ <i>interface</i>	Berfungsi untuk menambahkan kelas pada diagram.



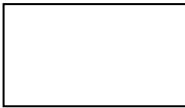
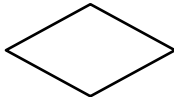
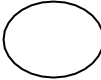
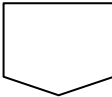
3.		Asosiasi	Relasi antara kelas dengan makna umum.
4.		Asosiasi berarah	Relasi antara kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
5.		Generalisasi	Relasi antara kelas dengan makna umum-khusus
6.		Agregasi	Hubungan antara kelas dengan makna semua bagian.





2.2.3.2 Flowchart

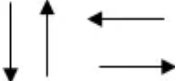


Flowchart atau bagan alir merupakan bagan (*chart*) yang menampakkan alir (*flow*) pada program atau mekanisme sistem secara logika. *Flowchart* sangat bermanfaat khususnya guna mendeskripsikan urutan proses yang pelaksanaannya memiliki berbagai pilihan atau percabangan. *Flowchart* membantu memahami urutan nalar yang rumit dan panjang. Dengan *flowchart* maka jalannya program dapat dikomunikasikan lebih mudah sehingga pembuatan *flowchart* merupakan mutlak khususnya guna melengkapi cara dokumentasi suatu *listing* program personal komputer yang rumit. Demi mendeskripsikan sebuah prosedur pemecahan terstruktur dan dipahami oleh oranglain terkhusus pada programmer yang bertugas mengimplementasikan program, maka diperlukan menggunakan alat bantu yang berbentuk diagram alir atau *flowchart*. Hal tersebut, dapat mendeskripsikan *flowchart* menjadi urutan menurut nalar suatu mekanisme pada pemecahan suatu kasus yang dituliskan berupa simbol-simbol tertentu misalnya persegi dan belah ketupat maupun menggunakan bentuk lain yang kemudian dihubungkan menggunakan garis-garis berarah. Setiap




Simbol dalam diagram ini, memperlihatkan langkah penyelesaian, sedangkan garis berpanah memperlihatkan urutan kejadian [19]. Berikut ini merupakan Simbol-simbol pada *flowchart* dan fungsinya disebutkan pada tabel 2.4 :



Tabel 2. 4 Simbol *Flowchart* [20]

No	Simbol	Nama	Keterangan
1.		<i>Terminal</i>	Awal atau akhir sebuah program
2.		<i>Output/Input</i>	Proses input atau output
3.		Proses	Proses operasional didalam komputer
4.		<i>Decision</i>	Suatu kondisi tertentu mengarah pada dua pilihan kemungkinan ya/tidak.
5.		<i>Connector</i>	Penghubung antar proses pada halaman yang sama
6.		<i>Offline Connector</i>	Penghubung antar proses pada halaman yang lain

7.		<i>Predefined Proses</i>	Simbol yang menunjukkan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
8.		<i>Document</i>	Simbol yang menunjukkan untuk mencetak laporan ke printer.
9.		<i>Manual input</i>	Simbol yang menunjukkan proses input yang dilakukan secara manual menggunakan online keyboard.
10.		<i>Display</i>	Simbol yang menunjukkan adanya <i>output</i> yang digunakan berupa layar

			(video, computer).
11.		<i>Flow Direction</i>	Simbol yang berfungsi untuk menyatakan jalannya alur sebuah proses
12.		<i>Manual</i>	Simbol yang menunjukkan proses diluar komputer atau kegiatan tanpa penggunaan komputer
13.		<i>Communication Link</i>	Simbol yang berfungsi untuk menyatakan bahwa adanya transisi suatu data/informasi dari satu lokasi ke lokasi lainnya

14.		<i>Keying Operation</i>	Simbol yang berfungsi untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard
15.		Offline Storage	Simbol yang menunjukkan bahwa data dalam Simbol ini akan disimpan ke suatu media tertentu.
16.		<i>Punched Card</i>	Simbol untuk menyatakan input berasal dari kartu atau output ditulis ke kartu.

17.		<i>Magnetic-tape unit</i>	Simbol untuk menyatakan input berasal dari pita magnetic atau output disimpan ke pita magnetic.
18.		<i>Disk Storage</i>	Simbol untuk menyatakan input berasal dari disk atau input disimpan ke disk.

2.2.3.3 *System Development Life Cycle (SDLC) (rpl)*

Metode SDLC merupakan salah satu metode yang memiliki ciri khas bahwa pengerjaan setiap fase wajib dikerjakan terlebih dahulu sebelum melanjutkan ke fase berikutnya. Kelebihan menggunakan metode SDLC adalah:

1. Urutan proses pengerjaan dengan metode ini menjadi lebih teratur pada setiap tahapannya.
2. Dari sisi user pula lebih menguntungkan karena bisa merencanakan dan menyiapkan semua kebutuhan data dan proses yang akan diperlukan.
3. Jadwal menjadi lebih jelas karena jadwal pada tiap proses dapat ditentukan secara pasti. Sehingga terlihat jelas sasaran penyelesaian pengembangan program dan progresnya [21]

Ada beberapa model pengembangan sistem pada metode SDLC, salah satunya adalah waterfall. Metode waterfall adalah sebuah pengembangan sistem dimana antar fase dilakukan secara berurutan. Implementasi dalam metode ini yaitu harus menyelesaikan tahap

pertama sebelum melanjutkan ke tahap selanjutnya [22]. Adapun tahapan dalam waterfall:

1. *Requirement Analysis and Definition* (Analisis dan Definisi Kebutuhan)

Pada tahap ini akan melakukan wawancara langsung maupun tidak langsung yang bertujuan untuk mengetahui apa saja kebutuhan sistem dan kegunaannya.

2. *System and Software Design* (Desain Sistem dan Perangkat Lunak)

Tahap ini akan mulai membangun desain dengan mempelajari kebutuhan sistem dan kegunaannya yang telah didapat pada tahap sebelumnya. Desain yang dibuat seperti *Entity Relationship Diagram* (ERD), *Data Flow Diagram* (DFD), *Flowchart* dan *Sequence Diagram*.

3. *Implementation and Unit Testing* (Implementasi dan Pengujian Unit)

Design yang telah dibentuk selanjutnya akan dibuat kode-kode sehingga menjadi perangkat lunak untuk dapat dikembangkan.

4. *Integration and System Testing* (Integrasi dan Pengujian Sistem)

Seluruh perangkat lunak yang telah dikembangkan sebelumnya akan diuji apakah sistem dapat berjalan dengan baik dengan mendeteksi kesalahan pada sistem.

5. *Operation and Maintenance* (Operasi dan Pemeliharaan)

Pada tahap yang terakhir ini akan menjalankan sistem jadi. Selain menjalankan sistem penulis akan melakukan pemeliharaan dan perbaikan sistem. Hal ini bertujuan agar sistem terus berjalan dengan baik.

2.2.3.4 *Blackbox Testing*

Pengujian *blackbox testing* merupakan pengujian guna mengetahui apakah fungsi-fungsi masukan dan keluaran berdasarkan aplikasi sesuai spesifikasi yang dibutuhkan. Langkah dalam pengujian ini yaitu menciptakan masalah uji yang bersifat mencoba seluruh fungsi menggunakan aplikasi apakah sinkron dengan spesifikasi yang dibutuhkan. Keuntungan penggunaan *blackbox testing* adalah:

1. Penguji tidak perlu mempunyai pengetahuan mengenai bahasa pemrograman tertentu.

2. Pengujian dilakukan menurut sudut pandang pengguna, ini membantu guna mendeteksi hal yang bermakna ganda pada spesifikasi persyaratan.
3. Programmer dan tester keduanya saling bergantung satu sama lain.

Kekurangan penggunaan metode *black box testing* yaitu:

1. Uji kasus sulit dirancang tanpa spesifikasi yang jelas.
2. Kemungkinan mempunyai pengulangan tes yang telah dilakukan *programmer*.
3. Beberapa bagian *back end* tidak diuji sama sekali[23].

2.2.4 Pemrograman Berbasis Objek

Pemrograman berbasis objek merupakan suatu cara baru dalam berpikir serta berlogika dalam menghadapi masalah-masalah yang akan dicoba atasi menggunakan komputer, dimana setiap objek merupakan entitas tunggal yang memiliki kombinasi struktur data dan fungsi tertentu [24].

Kerangka pada pemrograman berbasis objek adalah [25]:

1. **Komentar**
Komentar merupakan bagian dari program yang berupa sisipan, hanya untuk sebuah penjelasan dan bukan bagian dari instruksi.
2. **Paket(Package)**
Package atau paket merupakan pengelompokkan file kelas terkait (karena jenis, fungsionalitas, atau alasan lain) di direktori yang sama, dimana setiap kelas memiliki statement (statement java dalam kode yang digunakan untuk membuat kelas) sebuah package merujuk ke direktori tersebut.
3. **Class**
Class merupakan desain yang mendefinisikan *variable* dan metode pada semua objek yang diberikan. *Class* digunakan untuk menyimpan isi program yang akan dieksekusi, berisi tipe data dan metode untuk mengeksekusi suatu program.
4. **Class Body**
Class body merupakan bagian program yang mendefinisikan tentang objek maupun method yang digunakan untuk menciptakan objek sebagai tempat mendeklarasikan kode-kode program.

5. **Data**
Data adalah sekumpulan informasi atau keterangan untuk menyelesaikan masalah tertentu.
6. **Method**
Method merupakan tindakan yang bisa dilakukan didalam class. Seluruh sifat serta fungsi dapat diterapkan pada *method*, seperti pendapat(*argument*) atau parameter, mengembalikan sebuah nilai dengan *keyword return* dll.
7. **Main method**
Main method adalah blok program yang dieksekusi pertama kali. Tanpa adanya main method program tidak dapat dieksekusi.

Keunggulan yang dihasilkan pada PBO diantaranya:

1. Objek-objek yang sudah digunakan bisa digunakan ulang (*reusable*) untuk kepentingan lain.
2. Sifatnya alami dikarenakan sifat objek dan perilaku akan disesuaikan dengan objek nyata di sekitar.
3. Programnya lebih mudah dikembangkan dan terstruktur [26].

2.2.5 DBMS (Data Base Management System)(basis data)

Data Base Management System adalah himpunan data yang saling berhubungan berdasarkan struktur tertentu dan disimpan dengan baik. Untuk menerima informasi yang bermanfaat berdasarkan himpunan data maka dibutuhkan suatu perangkat lunak guna memanipulasi data sehingga informasi dapat bermanfaat. DBMS adalah perangkat lunak yang dipakai guna menciptakan sebuah basis data yang berbasis komputerisasi. DBMS membantu pada pemeliharaan dan pengolahan himpunan data jumlah besar sehingga penggunaan DBMS tidak mengakibatkan kekacauan dan bisa digunakan pengguna sesuai kebutuhan [27].

2.2.5.1 Basis Data

Basis data merupakan sebuah himpunan data yang secara logis terkait dan dibuat guna memenuhi suatu kebutuhan informasi berdasarkan sebuah organisasi. Adapun empat tahap dalam pemrosesan data yaitu[28] :

1. **Input data**
Input data adalah suatu proses memasukkan data ke dalam personal komputer dan data tersebut didapatkan dari dokumen yang telah tersedia di perusahaan.
2. **Pemrosesan data**

Pemrosesan data merupakan aktivitas mentransformasikan input sebagai hasil atau mengganti data sebagai informasi. Didalam pemrosesan data juga terdapat aktivitas lain seperti perhitungan, perbandingan, peringkasan, pemilahan dan pemanggilan.

3. Penyimpanan data

Penyimpanan data merupakan suatu proses pada data atau transaksi yang telah diolah disimpan agar dapat diakses dengan mudah dan efisien. Adapun jenis-jenis penyimpanan data yaitu penambahan, penghapusan, pembaruan, dan perubahan.

4. *Output*

Dari data atau transaksi yang telah diproses dan disimpan, maka akan menerima suatu hasil berupa informasi yang akan digunakan menjadi dasar pada pengambilan suatu keputusan

Database dikelola dengan cara menuliskan sebuah kode perintah berupa *query* SQL yang harus dimasukan sesuai kebutuhan dan hal tersebut terbagi menjadi 2 yaitu, DDL (*Data Definition Language*) dan DML (*Data Manipulation Language*). Berikut dibawah ini merupakan pengertian DDL dan DML:

1. DDL (*Data Definition Languages*)

DDL digunakan sebagai perintah yang berhubungan dengan pendefinisian dari suatu struktur database. Adapun perintah-perintah dasar pada DDL:

a. *Create*

Perintah dalam DDL untuk membuat database serta tabel. Contoh perintah *create*, “*CREATE TABLE data_proker(id INTEGER PRIMARY KEY, kode_proker VARCHAR(20) NULL, nama_proker VARCHAR(225) NOT NULL, nama_divisi VARCHAR(40) NOT NULL, tanggal_pelaksanaan DATE NULL, status VARCHAR(20) NOT NULL)*”.

b. *Rename*

Perintah yang digunakan untuk mengganti nama tabel. Contoh perintah *rename*, “*RENAME TABLE DATA_PROKER TO DATA_SEMUA_PROKER;*”

c. *Alter*

Perintah yang digunakan untuk mengedit atau mengubah. Contoh perintah *alter*, “*Alter TABLE proker Drop Column nama_divisi;*”

d. *Drop*

Perintah yang digunakan untuk menghapus tabel maupun database. Contoh perintah *drop*, “*Drop Table proker;*”

2. DML (*Data Management Languages*)
DML merupakan perintah yang digunakan untuk melakukan pengelolaan pada database seperti membuat baru, menghapus, menambah dan menampilkan. Berikut ini adalah perintah pada DML :
 - a. *Insert*
Perintah yang berfungsi untuk memasukkan data ke dalam database. Contoh perintah *insert*, “ *INSERT INTO* proker(id, kode_proker, nama_proker, tanggal_pelaksanaan) *values*(1234, “ProIH”, “*Study Excursie*”, “2021-04-13”);”.
 - b. *Update*
Perintah yang digunakan untuk pembaruan data. Contoh perintah dari *update*, “*Update* proker *Set* nama_proker=“*Study Excursie*” *Where* id= 1234;”.
 - c. *Select*
Perintah yang berfungsi untuk menampilkan data. Contoh perintah *select*, “*Select * From* proker;”.
 - d. *Delete*
Perintah yang digunakan untuk menghapus data. Contoh perintah dari *delete*, “*Delete From* proker *Where* nama_proker=“*Study Excursie*”;”.

2.2.6 Usability dan Evaluation

- a. Usability adalah suatu ukuran, dimana pengguna dapat mengakses fungsionalitas sebuah sistem dengan efektif, efisien, serta memuaskan dalam mencapai tujuan tertentu. Pengukuran usability, dilakukan menggunakan serangkaian kuesioner yang dapat mengolah data yang berhubungan dengan efektifitas, efisiensi dan kepuasan dalam penggunaan suatu sistem informasi. Kegunaan perangkat lunak dapat didefinisikan sebagai kualitas dalam aplikasi mengenai seberapa produktif dan mendukung akan kebutuhan pengguna.
- b. Evaluation adalah suatu proses yang sistematis dan berkelanjutan untuk menentukan kualitas dari sesuatu berdasarkan pertimbangan dan kriteria dalam rangka mengambil suatu keputusan. Evaluation merupakan gambaran kualitas sesuatu tentang nilai dan arti [29].