

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Kajian peneliti sebelumnya ini digunakan sebagai pembandingan antara penelitian yang sudah dilakukan dan yang akan dilakukan oleh peneliti.

Penelitian yang berkaitan dengan topik yang diangkat oleh peneliti dilakukan oleh [3]. Adapun tujuan penelitian ini adalah untuk membuat sebuah sistem informasi persuratan yaitu pengarsipan surat masuk dan surat keluar yang dapat membantu dinas pekerjaan umum kota Semarang dalam kegiatan surat menyurat yang ditangani oleh bagian sekretariat. Sistem informasi ini berbasis dekstop yang dibuat dengan menggunakan diagram *Unified Modeling Language* (UML), yaitu : *use case diagram* dan *class diagram*. Hasil dari pengembangan sistem informasi berbasis desktop ini telah dapat mencakup pengelolaan surat, yaitu surat masuk dan surat keluar, sehingga kegiatan operasional akan lebih efektif dan efisien dibandingkan dengan sistem manual yang membutuhkan lebih banyak proses pengerjaan, waktu, dan tenaga. Sistem yang telah dibuat ini lebih berfungsi untuk menjadikan data surat yang ada pada Dinas Pekerjaan Umum Kota Semarang lebih terstruktur dan tersusun rapi sehingga mempermudah dalam penyimpanan dan pencarian dokumen.

Peneliti terkait juga dilakukan oleh [4]. Penelitian ini bertujuan untuk merancang sistem yang dapat membantu Masyarakat dalam pengajuan surat secara *online* dan memudahkan pengarsipan. Sistem ini dirancang dengan metode *waterfall* dan pemodelan *Unified Modelling Language* (UML) kemudian diimplementasikan menggunakan pemrograman *web*, *basisdata* PHP MySQL. *Black Box Testing* digunakan sebagai pengujian fungsional sistem. Hasil penelitian ini adalah telah dikembangkannya sistem informasi pengarsipan dokumen dan permohonan surat *online* di tingkat kecamatan yang diberi nama Sipadu. Sebanyak 71% pengguna menyatakan sistem mudah digunakan berdasarkan *Usability Testing*.

Peneliti selanjutnya yang terkait juga dilakukan oleh [5]. Tujuan dari penelitian ini adalah mengembangkan sistem informasi penelitian dan Pengabdian kepada Masyarakat (PkM) pada LPPM Universitas Dinamika Bangsa, yang dimana sistem ini dapat mengelola data buku ajar, data hki, data luaran lainnya, data pertemuan ilmiah, data publikasi,

data penelitian, data PkM, verifikasi dan penilaian. Untuk mengembangkan sistem informasi dalam penelitian ini menggunakan metode *waterfall* dan pemodelan rancangan *usecase diagram*. Hasil dari penelitian ini adalah terbentuk sebuah sistem informasi, yang dimana sistem ini dapat mengelola data buku ajar, data HKI, data luaran lainnya, data pertemuan ilmiah, data publikasi, data penelitian, data PKM, verifikasi dan penilaian. Sistem ini telah diujicoba oleh peneliti dan hasilnya sistem dapat berjalan dengan baik sesuai dengan tujuannya.

Peneliti selanjutnya yang terkait juga dilakukan oleh [6]. Tujuan dari penelitian ini adalah untuk membuat aplikasi yang dapat memudahkan kedua pihak pemohon dan yang pihak melayani pembuatan surat keterangan usaha. Adapaun metode yang digunakan untuk merancang sistem tersebut adalah metode *waterfall* dan perancangan model *usecase diagram*, *flowchart*, *sequence diagram* dan *activity diagram*. Hasil dari penelitian ini adalah aplikasi ini dapat membantu para warga kelurahan Karangpawitan untuk membuat surat keterangan usaha secara *online* dan membantu pegawai dalam menangani proses pelayanan pembuatan surat.

Peneliti selanjutnya yang terkait juga dilakukan oleh [7]. Tujuan dari penelitian ini adalah untuk membangun aplikasi yang dapat memudahkan pimpinan perguruan tinggi memonitoring kegiatan penelitian dan pengabdian kepada Masyarakat. Sistem ini menghasilkan informasi berupa tabel dan grafik serta laporan kegiatan penelitian dan pengabdian kepada masyarakat per tahun. Adapaun metode yang digunakan untuk merancang sistem tersebut adalah metode *waterfall* dan perancangan model *usecase diagram*. Dalam penelitian ini telah dilakukan pengujian menggunakan *black box* yang dapat diketahui bahwa dari 149 item pengujian, telah berhasil 22 item hasilnya tidak valid dan 127 hasil valid, sehingga dapat diketahui tingkat keberhasilan aplikasi yang dibuat adalah 85,2%.

Pada penelitian ini akan dibangun sistem informasi permohonan surat tugas dan surat keterangan berbasis *web* di P3M Politeknik Negeri Cilacap menggunakan bahasa pemrograman PHP, metode pengembangan sistem dengan model *prototype*. Perbedaan dari penelitian yang dilakukan sebelumnya adalah sistem yang akan dibuat ini menggunakan tanda tangan digital bertipe png, jpeg, dan jpg sebagai persetujuan permohonan surat serta menggunakan *email* sebagai notifikasi untuk pemberitahuan penyetujuan dan hasil permohonan surat.

2.2 Landasan Teori

2.2.1 P3M

Pusat Penelitian dan Pengabdian kepada Masyarakat atau yang dikenal dengan sebutan P3M ini bertempat di Politeknik Negeri Cilacap yang memiliki tugas untuk memfasilitasi atau mengakomodir dosen dalam melaksanakan tridharma perguruan tinggi dalam hal penelitian dan pengabdian kepada masyarakat. Strategi P3M dalam usahanya untuk penguatan penelitian dan pengabdian, P3M Politeknik Negeri Cilacap melaksanakan berbagai program dan bantuan dalam bentuk kegiatan atau *reward* berdasarkan kinerja yang telah dicapai oleh peneliti atau dosen[1]. Berikut adalah program kegiatan yang diupayakan oleh P3M : Sistem pendukung keputusan memiliki sifat khas atau karakteristik dalam membantu mengambil keputusan yaitu:

1. Pelatihan penulisan artikel ilmiah nasional, mini *workshop medeley's* dan sosialisasi penyusunan laporan keuangan.
2. Pelatihan pemanfaatan hasil penelitian dan pengabdian kepada masyarakat yang berpotensi paten.
3. Pelatihan pengelolaan dan akreditasi jurnal elektronik.
4. Bantuan publikasi pada jurnal nasional terakreditasi maupun internasional berputasi.
5. Bantuan seminar nasional dan internasional bereputasi.
6. Seminar hasil penelitian dan pengabdian kepada masyarakat.
7. Menyelenggarakan seminar Bersama; Seminar Nasional Sentrinov, ICAST (seminar bersama antar Politeknik seluruh Indonesia) dan WiNCo (seminar bersama antar perguruan tinggi di Cilacap).

2.2.2 Penelitian Kepada Masyarakat

Penelitian merupakan kegiatan penemuan, penciptaan, dan pengembangan Ilmu Pengetahuan, Teknologi dan Seni. Pengabdian kepada masyarakat merupakan kegiatan penerapan Ilmu Pengetahuan, Teknologi dan Seni yang meliputi kegiatan pengembangan, penyebarluasan dan pembudayaan Ilmu Pengetahuan, Teknologi dan Seni[8].

2.2.3 Hak atas Kekayaan Intelektual

Hak kekayaan intelektual adalah hak yang timbul bagi hasil olah pikir otak yang menghasilkan suatu produk atau proses yang berguna untuk manusia. Pada intinya hki adalah hak untuk menikmati

secara ekonomi hasil suatu kreativitas intelektual. Hki dibagi menjadi 2 bagian, yaitu hak cipta dan hak kekayaan industri yang mencakup paten, desain industri, merek, penanggulangan praktik persaingan curang, desain tata letak sirkuit terpadu, dan rahasia dagang[9].

2.2.4 Surat

Surat merupakan salah satu sarana komunikasi secara tertulis yang berbentuk lembaran kertas yang memuat suatu informasi yang hendak disampaikan oleh seseorang kepada orang lain untuk menyampaikan informasi yang memiliki keperluan kegiatan tertentu. Informasi tersebut dapat berupa pemberitahuan, pertanyaan, laporan, peringatan[10].

2.2.5 Web

Web adalah jaringan komputer yang terdiri dari kumpulan situs internet berupa sejumlah halaman yang memiliki topik saling berkaitan dengan menawarkan teks, grafik, sumber daya animasi melalui *hypertext transfer protocol*[11].

2.2.6 PHP

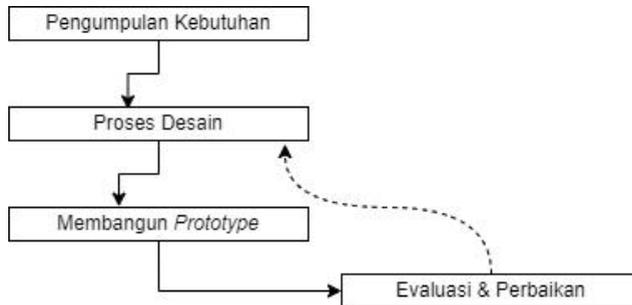
PHP adalah bahasa pemrograman *script* yang paling banyak dipakai untuk memprogram situs *web* dinamis. Pada awal dibuat PHP bernama FI (*Form Interpreted*), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data form dari *web*. Rasmus sang pencipta merilis kode sumber untuk umum, dengan perilsan kode sumber ini menjadi *open source* dan banyak *programmer* yang tertarik untuk ikut mengembangkan PHP[10].

2.2.7 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak merupakan serangkaian proses yang amat panjang untuk membuat atau menciptakan suatu perangkat lunak yang berkualitas, bukan merupakan cabang ilmu komputer yang mempelajari tentang *technical coding*[12].

2.2.8 Metode Pengembangan Sistem

Dalam merancang dan membangun Sistem Informasi Pengajuan Permohonan Surat Tugas Berbasis *Web* di P3M Politeknik Negeri Cilacap ini menggunakan metode *prototype*. Metode *prototype* diawali mengumpulkan kebutuhan [2].



Gambar 2. 1 Metode *Prototype*.

Untuk tahapan pengembangannya sebagai berikut :

1. Analisis Kebutuhan

Dalam rangka melakukan pengembangan sistem diperlukan penilaian kebutuhan awal dan analisa tentang ide atau gagasan untuk membangun ataupun mengembangkan sistem. Analisa dilakukan untuk mengetahui komponen apa saja yang sedang berjalan dan mengumpulkan informasi yang dibutuhkan pengguna akhir yang meliputi biaya dan maanfaat sistem yang akan dibangun. Analisa kebutuhan sistem mendefinikasikan kebutuhan sistem yang berupa :

- a. *Input* sistem
- b. *Output* sistem
- c. Proses yang berjalan dalam sistem
- d. Basisdata yang digunakan

2. Desain Sistem

Desain diperlukan dengan tujuan bagaimana sistem akan memenuhi tujuannya dibuat atau dicipatakan. Desain sistem terdiri dari kegiatan dalam mendesain yang hasilnya sebuah spesifikasi dari sistem. Bagian dari desain sistem dapat berupa konsep desain *interface*, proses dan data dengan tujuan menghasilkan spesifikasi sistem yang sesuai dengan

kebutuhan. Pada tahap ini penulis membuat rancangan sistem yang akan dikembangkan dengan membuat *flowchart*, *sequence diagram*, dan ERD sesuai dengan analisis yang dikerjakan.

3. Membangun *Prototyping*

Tahapan ini dilakukan setelah perancangan sistem telah selesai lalu pada tahap ini penulis membangun *prototyping* sesuai rancangan sistem yang dibuat dan yang diharapkan. Jika tidak sesuai maka perancangan sistem akan dibuat ulang dan jika sudah sesuai maka akan dilakukan tahapan selanjutnya yaitu pengujian *system*.

4. Pengujian Sistem

Pada tahap menguji sistem diharapkan sistem dapat berjalan sesuai dengan perencanaan dan desain awal yang dibuat. Pengujian sistem bertujuan menemukan kesalahan – kesalahan yang terjadi pada sistem untuk mengetahui apakah sistem sudah dapat berjalan dengan baik dan tidak terdapat *error*. Jika terdapat *error* maka akan dilakukan revisi sistem.

5. Implementasi

Setelah *prototype* diterima oleh pengguna tahap selanjutnya adalah implementasi sistem, tahap ini merupakan implementasi sistem yang sudah siap dioperasikan dan akan diadakan proses pembelajaran terhadap sistem baru.

2.2.9 Metode Pengujian Sistem

Terkait dengan pengujian program, pengujian yang akan digunakan adalah pengujian *black-box*. Pengujian *black-box* adalah metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal. Metode uji coba *black-box* memfokuskan pada keperluan fungsional dari *software* untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat – syarat fungsional suatu program. Pengujian *black-box* digunakan untuk memperlihatkan bahwa fungsi – fungsi perangkat lunak adalah operasional, bahwa *input* dapat diterima dengan baik dan *output* dihasilkan dengan tepat, dan integritas informasi eksternal (seperti file data). Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut[13] :

1. Fungsi - fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses *database* internal.

4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminal.

2.2.10 Tools Alat Bantu Penelitian

A. *Flowchart*

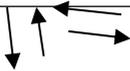
Flowchart adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam sebuah program atau prosedur sistem secara logika. Bagan alir (*flowchart*) digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. *Flowchart* adalah penggambaran secara grafik dari langkah-langkah dan urutan - urutan prosedur dari suatu program[14].

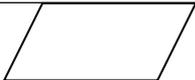
Ada beberapa jenis-jenis *flowchart* diantaranya :

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

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

Tabel 2. 1 Simbol *Flowchart*

No	Simbol	Nama	Keterangan
1		<i>Terminator Symbol</i>	Yaitu simbol untuk permulaan (<i>start</i>) atau akhir (<i>end</i>) dari suatu kegiatan (<i>flowchart</i>).
2		<i>Flow Direction Symbol</i>	Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain atau mempresentasikan alur kerja.
3		<i>Processing Symbol</i>	Yaitu simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.

4		<i>Manual Operation Symbol</i>	Yaitu simbol yang menunjukkan pengolahan data yang tidak dilakukan oleh komputer.
5		<i>Decision Symbol</i>	Yaitu simbol untuk pemilihan proses berdasarkan kondisi yang ada.
5		<i>Input-Output Symbol</i>	Yaitu simbol yang menyatakan proses input dan <i>output</i> tanpa tergantung dari jenis peralatannya.
6		<i>Document Symbol</i>	Yaitu simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak ke kertas.

2.2.11 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah sebuah bahasa permodelan yang telah menjadi standar dalam industri *software* untuk visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak. Macam-macam dari *Unified Modeling Language (UML)* antara lain: *use case diagram*, *sequence diagram* dan *class diagram*[15].

a. Usecase Diagram

Usecase 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 skenario 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 berdasarkan keperluan *actor*. Berdasarkan definisi diatas maka dapat disimpulkan bahwa *use case* adalah mendeskripsikan bagaimana sistem akan terlihat di mata pengguna potensial yang terdiri

dari sekumpulan *scenario* dan *actor*. Sedangkan *use case* diagram memfasilitasi komunikasi antara analis dan pengguna serta analisis dan klien[16]. Simbol *Use case* dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Simbol *Usecase* Diagram

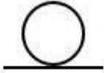
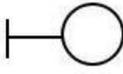
No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Menyepesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>not independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menyepesifikasikan bahwa <i>use case</i> sumber secara eksplisit.

5		<i>Extend</i>	Menyepesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association,</i>	Yang menghubungkan antara objek satu dengan objek lainnya
7		<i>System</i>	Menyepesifikasikan paket yang menampilkan sistem secara terbatas
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerjasama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

b. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antara objek didalam dan disekitar sistem berupa message yang digambarkan terhadap waktu. Sequence diagram yang digunakan juga menggambarkan berbagai aliran aktifitas dalam sistem yang sedang dirancang, bagaimana masing – masing aliran berawal dan berakhir[17]. Berikut adalah elemen-elemen dari *sequence diagram*.

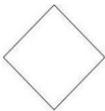
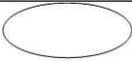
Tabel 2. 3 Simbol *Sequence Diagram*

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan.
3		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari <i>form</i> .
4		<i>Control Class</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel.
5		<i>A Focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya <i>massage</i> .
6		<i>A Message</i>	Menggambarkan Pengiriman Pesan.

a. *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram atau disebut dengan ERD merupakan bentuk jaringan yang disimpan secara abstrak dengan menggunakan susunan data. ERD adalah bentuk data konseptual, yang memiliki fokus pada struktur ERD dan data relasional. Entitas merupakan sesuatu hal yang unik, karena dalam entitas dapat dibedakan antar objeknya, sehingga antar objek yang satu dengan lainnya memiliki perbedaan. Atau dengan kata lain, entitas adalah objek yang bisa dikatakan mewakili atau melambangkan suatu hal di dalam dunia nyata. Dan setiap entitas pasti memiliki banyak atribut, atribut ini berfungsi untuk mendeskripsikan karakteristik yang terdapat dalam objek tersebut [18]. Simbol-simbol dalam ERD adalah sebagai berikut:

Tabel 2. 4 *Entity Relationship Diagram*

No	Simbol	Nama	Fungsi
1.		Entitas	Entitas merupakan data inti yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer.
2.		Relasi	Relasi yang menghubungkan antar entitas. Biasanya diawali dengan kata kerja.
3.		Atribut	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.

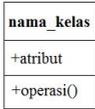
No	Simbol	Nama	Fungsi
4.		Atribut Kunci Primer	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan.
5.		Atribut Multinilai	<i>Field</i> yang butuh disimpan dalam entitas yang dapat memiliki nilai lebih dari satu.
6.		Alur	Alur memiliki fungsi untuk menghubungkan atribut dengan entitas dan entitas dengan relasi.
7.		Asosiasi/ <i>Association</i>	Penghubung antara relasi dan entitas dimana kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian, kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas.

d. *Class Diagram*

Class diagram adalah bentuk visual dari sebuah struktur atau sistem program pada jenis – jenis yang di bentuk dimana *class diagram* merupakan alur jalannya *database* pada sebuah sistem. *Class diagram* juga merupakan penjelasan proses *database* dalam suatu program dimana dalam sebuah proses sistem maka *class diagram* ini

harus ada. *Class diagram* bersifat statis atau yang biasa disebut dengan menggambarkan interaksi apa yang terjadi bukan apa yang terjadi jika mereka berinteraksi. Dalam diagram kelas kita bisa memilih fondasi buat diagram yang memperlihatkan aspek lain menurut sistem, misalnya status objek & kerja sama antar objek yang ditampilkan pada diagram yang dinamis. Diagram kelas hanya memperlihatkan kelas, bukan objek. Diagram objek memperlihatkan bahwa *instance* objek actual menurut kelas [19].

Tabel 2. 5 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> .
3.		Asosiasi berarah/ <i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
4.		Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).

5.		Kebergantungan / <i>Dependency</i>	Relasi antarkelas dengan makna kebergantungan antarkelas.
6.		Agregasi / <i>Aggregation</i>	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>).

2.2.12 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan projek yang berisi data dan operasi yang diperlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek dan pengujian berorientasi objek[20].

Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek – objek dunia nyata. Ada banyak cara untuk memodelkan objek seperti mulai dan abstraksi objek, kelas, hubungan antarkelas sampai abstraksi sistem. Berikut adalah beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek:

a. Kelas

Kelas adalah kumpulan objek – objek dengan karakteristik yang sama. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/metode), hubungan (*relationship*) dan arti. Suatu kelas dapat diturunkan dan kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru. Secara teknis, kelas adalah sebuah struktur tertentu dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode program yang menggunakan metodologi berorientasi objek.

b. Objek

Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status, atau hal – hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Secara teknis, sebuah objek saat program dieksekusi maka akan dibuat sebuah objek.

c. Metode

Operasi atau metode atau *method* pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek. Metode atau operasi dapat berasal dari *event*, aktivitas atau aksi keadaan, fungsi atau kelakuan dunia nyata. Contoh metode atau operasi misalnya *read*, *write*, *move*, *copy*, dan sebagainya. Kelas sebaiknya memiliki metode *get* dan *set* untuk setiap atribut agar konsep enkapsulasi tetap terjaga. Metode *get* digunakan untuk memberikan akses kelas lain dalam mengakses atribut, dan *set* adalah metode yang digunakan untuk mengisi atribut, agar kelas lain tidak mengakses atribut secara langsung.

d. Atribut

Atribut dari sebuah kelas adalah *variable* global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen – elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya. Atribut sebaiknya bersifat privasi untuk menjaga konsep enkapsulasi.

e. Pewarisan

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.

f. Antarmuka

Antarmuka 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. Sebuah kelas dapat mengimplementasikan lebih dari satu antarmuka dimana kelas ini akan mendeklarasikan metode pada antarmuka yang dibutuhkan oleh kelas itu sekaligus mendefinisikan isinya pada kode - kode program kelas itu.

g. *Reusability*

Pemanfaatan kembali objek yang sudah di definikasikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

h. Polimorfisme

Kemampuan suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

2.2.13 Database

Basis data adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file lain sehingga membentuk satu bangun data untuk menginformasikan 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 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 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 manajemen data yang terdapat dalam *database*[21]. 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.

```
CREATE TABLE `anggota_penelitian` (
  `id_anggota_penelitian` varchar(50) NOT NULL,
  `id_penelitian` varchar(50) NOT NULL,
  `nm_anggota` varchar(50) NOT NULL,
  `urutan` int(3) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- 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*

```
ALTER TABLE `anggota_penelitian`
ADD PRIMARY KEY (`id_anggota_penelitian`);
```

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.

```
INSERT INTO `val_ketua` (`id_val_ketua`, `id_pegawai`, `id_usulan`,
`tgl_val_ketua`) VALUES
```

- b. *SELECT* : perintah ini digunakan untuk mengambil dan menampilkan data dari tabel atau bahkan dari beberapa tabel dengan penggunaan relasi.

```
SELECT * FROM anggota_penelitian
```

- c. *UPDATE* : perintah *update* digunakan untuk memperbaharui data pada sebuah tabel.

```
UPDATE prodi SET nm_prodi = ALKS WHERE id_prodi = 1;
```

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

```
mysql_query($con, "delete from pengusul where id_pengusul = '1'");
```

2.2.14 Skala Likert

Skala Likert adalah skala yang digunakan untuk mengukur persepsi, sikap atau pendapat seseorang atau kelompok mengenai sebuah peristiwa atau fenomena social. Skala *likert* merupakan metode yang digunakan untuk keperluan penelitian atas jawaban responden terhadap suatu pernyataan. Dengan skala likert mempunyai gradasi dari sangat tidak setuju sampai sangat setuju[22]. Pada penelitian kuantitatif, jawaban item *instrument* dapat diberi skor seperti pada Tabel 2.6.

Tabel 2. 6 Tabel Skala Likert

Jawaban	Skor
Sangat Tidak Setuju (STS)	1
Tidak Setuju (TS)	2
Setuju (S)	3
Sangat Setuju (SS)	4

2.1.1 Usability

Usability adalah atribut kualitas yang mengukur seberapa mudah penggunaan suatu antarmuka (*interface*). “*Usability*” merujuk pada metode untuk meningkatkan kemudahan pemakaian selama proses perancangan. *Usability* memiliki lima aspek yaitu *learnability*, *efficiency*, *memorability*, *errors*, dan *satisfaction*. Pengukuran *usability* bergantung pada kemampuan pengguna menyelesaikan serangkaian tes [23]. Adapun nilai uji *usability* dapat dihitung menggunakan rumus:

$$Usability = \frac{\text{Nilai Perolehan}}{\text{Nilai Maximal}} \times 100\%$$

Keterangan : Nilai hasil uji usabilitas

np : Nilai perolehan dari hasil penilaian responden

nm : Nilai maksimal yang dapat diperoleh (jumlah responden x nilai maksimal skala likert x jumlah pertanyaan)

tegori yang dipakai dalam mengelompokkan uji *usability* ditunjukkan pada Tabel 2.7.

Tabel 2. 7 Kategori Intpretasi Predikat Sistem

Skor	Kualifikasi
0%-25%	Sangat Tidak Setuju
26% – 50%	Tidak Setuju
51% – 75%	Setuju
76% – 100%	Sangat Setuju

~Halaman Sengaja Dikosongkan~