



BAB II

DASAR TEORI

BAB II DASAR TEORI

2.1 Sistem Infomasi

Sistem adalah sekumpulan dari elemen-elemen yang melakukan interaksi satu sama lain dengan pola teratur sehingga membentuk suatu totalitas untuk menyelesaikan suatu masalah tertentu. Berdasar dari beberapa pendapat ahli yang telah dikemukakan di atas, dapat ditarik sebuah kesimpulan bahwa sistem adalah kumpulan bagian atau beberapa subsistem yang dirancang dan disatukan untuk mencapai suatu tujuan tertentu [2]. Mulyanto mengemukakan dalam bukunya yang berjudul Sistem Informasi Konsep dan Aplikasi, Mulyanto menjelaskan karakteristik sistem yang memiliki beberapa komponen yang mendukung sistem, antara lain:

a. Komponen Sistem (*System Components*)

Suatu sistem tidak mungkin ada dalam lingkungan yang kosong, tetapi suatu sistem ada dan memiliki fungsi di dalam lingkungan yang berisi sistem lainnya. Suatu sistem juga terdiri dari beberapa bagian yang saling berinteraksi satu sama lain dan melakukan kerja sama dalam membentuk satu kesatuan. Jika sebuah sistem merupakan salah satu dari bagian dari sistem lain yang lebih besar, maka sebuah sistem tersebut akan disebut dengan subsistem, sedangkan sistem lain yang lebih besar tersebut merupakan lingkungannya.

b. Batasan Sistem (*System Boundary*)

Batas dari suatu sistem adalah pemisah atau pembatas antara sistem tersebut dengan sistem lain atau dengan lingkungan luarnya.

c. Lingkungan (*Environment*)

Lingkungan adalah apapun di luar batas dari sebuah sistem yang dapat mempengaruhi operasi dari sistem tersebut, baik pengaruh yang merugikan ataupun yang menguntungkan. Pengaruh yang merugikan ini tentunya harus ditahan dan dikendalikan sehingga tidak mengganggu keberlangsungan sistem. Sedangkan lingkungan yang menguntungkan harus dijaga agar dapat mendukung keberlangsungan operasi dari sistem tersebut.

d. Penghubung antar Komponen (*Interface*)

Penghubung antar komponen adalah medium antara satu subsistem dengan subsistem yang lainnya. *Interface* inilah yang akan menjadi medium yang digunakan *input* (masukan) hingga *output* (keluaran). Dengan subsistem yang lain membentuk satu kesatuan.

e. Masukan (*Input*)

Masukan atau data input adalah data yang dimasukkan ke dalam suatu sistem. Masukan

tersebut dapat berupa masukan perawatan (*maintenance input*), yaitu bahan yang dimasukkan agar sistem tersebut dapat beroperasi dan masukan sinyal (*signal input*), yang merupakan masukan yang diproses untuk mendapatkan keluaran.

f. Pengolahan (*processing*)

Pengolahan (*processing*) adalah bagian dari suatu sistem yang melakukan perubahan dari input untuk menjadi output yang sesuai dengan tujuan dari sistem.

g. Tujuan (*Goal*) dan Sasaran (*Objective*)

Sebuah sistem pasti mempunyai sasaran (*objective*) atau tujuan (*goal*). Jika suatu sistem tidak mempunyai tujuan, maka operasi dari sistem tersebut tidak akan ada gunanya. Tujuan inilah yang mengarahkan kemana suatu sistem tersebut berjalan. Tanpa adanya tujuan yang mengarahkan sistem, maka suatu sistem menjadi tidak terarah dan tidak terkendali.

h. Keluaran (*Output*)

Keluaran atau *output* adalah hasil dari pemrosesan suatu sistem. *Output* dapat berupa informasi untuk selanjutnya digunakan sebagai masukan pada sistem lain atau hanya sebagai keluaran akhir.

i. Mekanisme Pengendalian dan Umpan Balik

Mekanisme pengendalian (*control mechanism*) diwujudkan dengan menggunakan umpan balik (*feedback*), yang mencuplik keluaran. Umpan balik ini digunakan untuk *mengendalikan* baik masukan maupun proses. Tujuannya adalah untuk mengatur agar sistem berjalan sesuai dengan tujuan.

McLeod (2004) dalam Yakub (2012: 8) menyatakan bahwa “Informasi (*information*) adalah data yang diolah menjadi bentuk lebih berguna dan lebih berarti bagi yang menerimanya”. Informasi juga disebut data yang diproses atau data yang memiliki arti. Informasi merupakan data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan [3].

Sistem informasi adalah alat untuk menyajikan informasi sedemikian rupa sehingga bermanfaat bagi penerimanya. Tujuannya adalah untuk memberikan informasi dalam perencanaan, memulai, pengorganisasian, operasional sebuah perusahaan yang melayani sinergi organisasi dalam proses mengendalikan pengambilan keputusan [4].

2.2 Inventaris Barang

Inventaris barang adalah kegiatan pengelolaan persediaan segala sesuatu yang memiliki nilai finansial, berlaku untuk perorangan, perusahaan dan pemerintahan [5]. Sistem inventaris juga bisa diartikan sebagai suatu proses untuk mengelola serta mendata barang yang terdapat pada sebuah

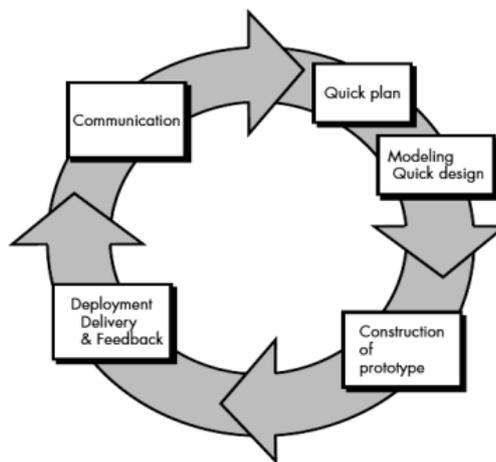
organisasi, instansi, maupun perusahaan yang bertujuan untuk membuat pendataan dengan terstruktur dan sistematis menggunakan sebuah teknologi agar bisa mengefisiensi waktu pada saat melakukan inventarisasi [6].

2.3 Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak (RPL) adalah suatu disiplin ilmu yang mengulas semua aspek produksi perangkat lunak, dimulai dari tahap awal yaitu *communication*, *requirements capturing* (analisa kebutuhan pengguna), *specification* (menentukan spesifikasi dari kebutuhan pengguna), *desain*, *coding*, *testing* hingga *maintenance* (pemeliharaan sistem) setelah digunakan[7]. Metode pengembangan sistem dan struktur data yang digunakan dalam penelitian ini adalah sebagai berikut.

A. Metode *Prototype*

Metode yang digunakan dalam pengembangan sistem yang dilakukan penulis yaitu metode *Prototype*. Metode *Prototype* akan menghasilkan sistem sebagai perantara antara *Developer* dan User karena dapat memudahkan User dalam memilih sistem yang sesuai dari apa yang diharapkan. Pembuatan model *software* yang sederhana dengan penggambaran dasar yang digunakan untuk dijadikan rancangan merupakan pengertian dari *prototype* [1]. Berikut ini merupakan alur dari metode *Prototype* yang ditunjukkan pada Gambar 2. 1.



Gambar 2. 1 Alur Metode *Prototype*

Metode *prototype* ini memiliki beberapa tahapan yang memiliki perannya masing-masing selama proses perancangan perangkat lunak yang bisa dijelaskan masing -masing tahapan tersebut pada penjelasan dibawah ini [8]:

1. *Communication*

Dimulai dengan tahap *communication*, tahapan ini bertujuan untuk mengidentifikasi berbagai kebutuhan aplikasi yang akan dirancang nantinya dengan melibatkan para client yang bersangkutan agar selama proses perancangan bisa memberikan hasil yang tepat sesuai keinginan *client* yang bersangkutan.

2. *Quick Plan*

Pada tahap *quick plan* ini perancang perangkat lunak akan melakukan perencanaan cepat sesuai dengan spesifikasi kebutuhan user berdasarkan data yang telah dikumpulkan pada tahap *communication* dengan merancang desain antarmuka yang dibutuhkan dan kebutuhan pendukung pada proses ini.

3. *Modelling Quick Design*

Pada tahap ini tim perancang akan membuat model *design* UML ataupun pemodelan yang dibutuhkan lainnya dengan waktu perancangan yang efektif untuk mendeskripsikan kebutuhan *client* berdasarkan analisis yang telah dilakukan sebelumnya.

4. *Construction of Prototype*

Selanjutnya pada tahap ini perancang akan memulai membangun perangkat lunak berdasarkan data yang telah dikumpulkan sebelumnya, proses pembangunan ini lebih berfokus terhadap aspek utama perangkat lunak dengan maksud pada proses selanjutnya perancang bisa dengan cepat mendapatkan *feedback* dari *client* tentang perangkat lunak yang dibuat.

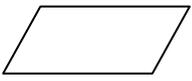
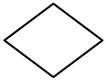
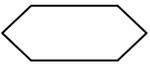
5. *Deployment Delivery & Feedback*

Dalam tahap ini *prototype* akan diserahkan kepada client untuk mendapatkan *feedback* dari hasil *prototype* tersebut, *feedback* tersebut akan digunakan sebagai landasan untuk memperbaiki *prototype* agar sesuai dengan *spesifikasi* kebutuhan *client*.

B. *Flowchart*

“*Flowchart* dapat diartikan sebagai langkah langkah penyelesaian masalah yang di tuliskan dalam suatu simbol-simbol tertentu. Diagram alir ini akan menunjukkan alur di dalam program secara logika” [9]. Pada Tabel 2. 1 merupakan simbol-simbol yang digunakan saat membuat *flowchart*.

Tabel 2. 1 Simbol Simbol Dalam *Flowchart*

No	Simbol	Nama	Keterangan
1		<i>Terminator</i>	Awal atau akhir suatu program (Prosedur).
2		<i>Input/Output</i>	Proses input atau output terlepas dari jenis perangkat
3		<i>Process</i>	Proses operasional komputer
4		<i>Decision</i>	Untuk menunjukkan bahwa suatu kondisi tertentu mengarah pada dua kemungkinan, yaitu ya atau tidak
5		<i>Connector</i>	Koneksi penghubung proses ke proses lain pada halaman yang sama
6		<i>Offline Connector</i>	Koneksi penghubung dari satu proses ke proses lain di halaman lain
7		<i>Predefined Process</i>	Mewakili ketentuan penyimpanan untuk diproses untuk memberikan awal harga
8		<i>Punched Card</i>	Input berasal dari kartu atau output ditulis ke kartus
9		<i>Document</i>	Mencetak output dalam format dokumen (melalui printer)
10		<i>Flow</i>	Menyatakan jalannya arus suatu proses
11		<i>Manual Operation</i>	Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer

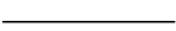
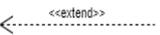
C. UML (*Unified Modeling Language*)

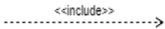
UML (*Unified Modeling Language*) merupakan pengganti dari metode analisis berorientasi *object* dan *design* berorientasi *object* (OOAD&D / *object oriented analysis and design*) yang dimunculkan sekitar akhir tahun 80-an dan awal tahun 90-an. UML merupakan gabungan dari metode Booch, Rumbaugh (OMT) dan Jacobson. Tetapi UML mencakup lebih luas daripada OOAD. Pada pertengahan saat pengembangan UML, dilakukan standarisasi proses dengan OMG (*Object Management Group*) dengan harapan UML bakal menjadi bahasa standar pemodelan pada masa yang akan datang (yang sekarang sudah banyak dipakai oleh berbagai kalangan) [10]. UML yang dipakai dalam penelitian ini adalah:

a. Use Case Diagram

Use case diagram merupakan pemodelan perilaku (*behavior*) dari suatu sistem yang dibutuhkan serta diharapkan pengguna. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem dan siapa saja atau aktor-aktor yang berhak menggunakan fungsi-fungsi tersebut [11]. Simbol-simbol yang digunakan pada *use case diagram* dapat dilihat pada Tabel 2. 2:

Tabel 2. 2 Simbol *Use Case Diagram*

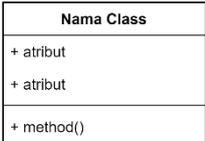
No	Simbol	Nama	Fungsi
1	 Actor	<i>Actor</i>	Digunakan untuk menjelaskan sesuatu atau seseorang yang sedang berinteraksi dengan sistem.
2	 Use Case	<i>Use Case</i>	Menggambarkan suatu perilaku dari sistem tanpa mengungkapkan struktur internal dari sistem tersebut.
3		<i>Assosiation</i>	Jalur komunikasi antar actor dengan <i>use case</i> yang saling berpartisipasi.
4		<i>Extend</i>	Penambahan perilaku ke dalam <i>use case</i> dasar yang tidak tahu tentang hal tersebut.
5		<i>Use case generalization</i>	Hubungan antara <i>use case</i> umum dengan <i>use case</i> yang lebih spesifik, yang mewarisi dan menambah fitur terhadapnya.

6		<i>Include</i>	Penambahan perilaku ke dalam <i>use case</i> dasar yang secara eksplisit menjelaskan penambahannya.
---	---	----------------	---

b. *Class Diagram*

Class Diagram menggambarkan serta deskripsi dari class, atribut dan objek serta hubungan satu sama lain. *Class diagram* dapat memberikan pandangan global atas sebuah system. Hal tersebut tercermin dari class yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa *class diagram*. *Class diagram* sangat membantu dalam visualisasi struktur kelas dari suatu *system*. Diagram ini umum digunakan pada pemodelan *system* berorientasi objek. *Class Diagram* berfungsi untuk menjelaskan tipe dari objek sistem dan hubungannya dengan objek yang lain. Simbol-simbol *class diagram* dapat dilihat pada Tabel 2. 3.

Tabel 2. 3 Simbol-simbol *Class Diagram*

No	Simbol	Nama	Keterangan
1		<i>Class</i>	Himpunan objek-objek dari berbagai atribut yang me-miliki operasi yang sama..
2		<i>Association</i>	Relasi antar kelas dengan makna umum dan biasanya disertai multiplicity.
3		<i>Directed Association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain.
4		<i>Aggregation</i>	Mengindikasikan keseluruhan bagian relationship disebut sebagai relasi.
5		<i>Composition</i>	Relasi Composition terhadap class tempat dia bergantung.
6		<i>Dependency</i>	Menunjukkan operasi pada suatu class yang menggunakan class yang lain.

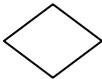
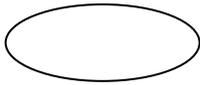
2.4 Basis Data

Menurut Begg & Connolly (2010:65), basis data merupakan suatu kumpulan data yang dirancang untuk menyimpan kebutuhan informasi dari sebuah organisasi secara logis. Pada pembelajaran SMK khususnya program kejuruan RPL, basis data merupakan ilmu yang mempelajari tentang hubungan antara kumpulan data yang disimpan tanpa *redundancy* atau perulangan yang menyebabkan data sama dengan yang lain untuk memenuhi berbagai kebutuhan aplikasi [12]. Dalam penelitian ini, digunakan *Entity Relationship Diagram* (ERD) untuk membantu dalam proses pembuatan basis data.

1. *Entity Relationship Diagram* (ERD)

Menurut Linda Marlinda (2004), “Model Entity Relationship merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan suatu persepsi bahwa real word terdiri dari objek-objek dasar mempunyai hubungan atau relasi antar objek-objek tersebut, relasi antar objek dilukiskan dengan menggunakan simbol-simbol grafis tertentu” [13]. Berikut ini simbol-simbol yang digunakan pada ERD yang dapat dilihat pada Tabel 2. 4.

Tabel 2. 4 Simbol-simbol *Entity Relationship Diagram* (ERD)

No	Simbol	Nama	Keterangan
1		Entitas	Merupakan suatu simbol untuk mewakili suatu objek dengan karakteristik sama yang dilengkapi oleh atribut.
2		Relasi	Simbol yang digunakan untuk menghubungkan beberapa entitas berdasarkan fakta pada suatu lingkungan. Digambarkan dengan simbol belah ketupat.
3		Atribut	Simbol yang menjelaskan karakteristik suatu entitas dan juga relasinya. Digambarkan dengan elips.
4		<i>Connection</i>	Simbol untuk merangkai keterkaitan antara notasinotasi yang berupa garis penghubung

			yang digunakan seperti entitas, relasi dan atribut.
--	--	--	---

2. MySQL

(Adani, 2020) berpendapat bahwa MySQL adalah sebuah DBMS (*Database Management System*) menggunakan perintah SQL (*Structured Query Language*) yang banyak digunakan saat ini dalam pembuatan aplikasi berbasis website. MySQL dibagi menjadi dua lisensi, pertama adalah *free software* dimana perangkat lunak dapat diakses oleh siapa saja. Dan kedua adalah *shareware* dimana perangkat lunak berpemilik memiliki batasan dalam penggunaannya. MySQL termasuk ke dalam RDBMS (*Relational Database Management System*). Sehingga, menggunakan tabel, kolom, baris, di dalam struktur databasenya. Jadi, dalam proses pengambilan data menggunakan metode *relational database*. Dan juga menjadi penghubung antara perangkat lunak dan *database server* [14].

2.5 Hypertext Preprocessor (PHP)

“PHP (*HyperText PreProcessor*) merupakan bahasa pemrograman yang di proses di server, Fungsi utama PHP dalam membangun website adalah untuk melakukan pengelolaan data dalam *database*”[15]. Menurut Arief (2011c:43) dalam Pemograman Web Dinamis Menggunakan PHP dan MySQL, PHP adalah bahasa *server side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis [16].

2.6 Laravel

Laravel adalah sebuah framework web berbasis PHP yang *open source* dan tidak berbayar, diciptakan oleh Tylor Otwell dan di peruntukan untuk pengembangan aplikasi web yang menggunakan pola MVC. Struktur pola MVC pada *Laravel* sedikit berbeda pada struktur pola MVC pada umumnya. Di *Laravel* terdapat *routing* yang menjembatani antara *request* dari user dan *controller*. Jadi controller tidak langsung menerima request tersebut. Pada penelitian ini penulis menggunakan *Laravel* dengan versi 10 [17].