



BAB II

TINJAUAN PUSTAKA DAN

LANDASAN TEORI

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Beberapa penelitian yang relevan dilakukan oleh beberapa peneliti antara lain : penelitian dari Khuswatul Nur Fadhillah, Sugeng Wahyudiono, dan Sri Waluyo pada tahun 2017 dengan judul “Sistem Informasi Pelatihan di Balai Latihan Kerja Kabupaten Magelang”. Sistem ini dikembangkan karena penyampaian informasi pelatihan terbatas dan tidak menyeluruh sehingga yang mendaftar pelatihan hanya orang di sekitar lingkungan BLK. Metode pengembangan perangkat lunak yang digunakan adalah pengembangan dengan metode *waterfall* yaitu membuat aplikasi pendaftaran BLK kabupaten Magelang berbasis web sesuai dengan rancangan yang telah dibuat dengan bahasa Pemrograman PHP dan *database MySQL* sehingga dalam penyampaian informasi pelatihan dapat menyeluruh dan mengatasi proses pendaftaran calon peserta pelatihan [5].

Penelitian lainnya dengan judul “Implementasi Metode MVC untuk Pembangunan Sistem Informasi Pelatihan Kerja : Studi Kasus UPT Pelatihan Kerja Singosari” yang dirancang oleh Usman Nurhasan, Betlian Fajrin, Defandy Fanny A, Fredo Vale Yuda Ughay. Berkurangnya daya tarik calon peserta pelatihan dan pencari pekerjaan untuk menggali informasi lebih dalam di UPT Pelatihan Kerja Singosari, maka dibuatlah sebuah sistem informasi berbasis website dan video profile mengenai kegiatan-kegiatan yang dilakukan dan fasilitas-fasilitas untuk mendukung kegiatan dari para peserta pelatihan di UPT Pelatihan Kerja Singosari dengan menggunakan metode MVC [6].

Penelitian selanjutnya juga pernah dilakukan oleh Sulistiyono, Agus Irawan, dan Syaiful Rizal dengan judul “Sistem Informasi Penerimaan Peserta Didik Baru Berbasis Web pada Balai Latihan Kerja (BLK) Kota Cilegon”. BLK Cilegon masih menggunakan media cetak seperti brosur dan pamflet, ataupun dengan menggunakan media sosial seperti facebook dalam menginformasikan layanan layanannya kepada masyarakat. Sistem ini dirancang untuk memudahkan masyarakat mengetahui informasi dengan mudah mengakses dan mendaftar tanpa harus datang ke Dinas Tenaga Kerja Kota Cilegon karena lokasinya yang jauh dari pusat kota. Metode pengembangan sistem yang dibuat

dalam penelitian ini pengembangan metode *waterfall* [7].

Penelitian yang akan dilakukan memiliki perbedaan dengan penelitian-penelitian sebelumnya. Perbedaan tersebut diantaranya dapat menginformasikan mengenai kode verifikasi saat melakukan pendaftaran melalui email, terdapat proses pencetakan kartu peserta tes, terdapat proses validasi pada berkas pendaftaran yang digunakan sebagai persyaratan pendaftaran, dan dapat memfilter nama calon peserta bagi yang lebih memilih mengikuti pelatihan program pemerintah (gratis) agar dapat mendaftar dirinya hanya satu kali saja. Tahap pengembangan sistem informasi ini menggunakan metode RAD (*Rapid Application Development*) yang dibangun menggunakan bahasa pemrograman PHP (*Hypertext Processor*) dan *MySQL* sebagai aplikasi *database* dengan penambahan fitur email *gateway*.

2.2 Landasan Teori

2.2.1. Sistem

Sekelompok komponen yang saling berhubungan dan saling bekerja sama untuk mencapai beberapa tujuan. Proses yang harus diikuti dalam pengembangan suatu sistem yang baik disebut ***Sistem Analysis And Design (SA&D)***. Proses SA&D ini didasarkan pada pendekatan sistem untuk mengatasi suatu masalah yg disebabkan oleh beberapa prinsip dasar.

Mengembangkan suatu sistem perlu membedakan unsur-unsur dari sistem yang membentuknya. Berikut ini karakteristik sistem yang dapat membedakan suatu sistem dengan sistem lainnya, antara lain [8] :

1. Batasan (*Boundary*) : Penggambaran dari suatu elemen/unsur mana yang termasuk di dalam sistem dan mana yang di luar sistem.
2. Lingkungan (*Environment*) : Segala sesuatu di luar sistem, lingkungan menyediakan asumsi, kendala, dan *input* terhadap suatu sistem.

3. Masukan (*Input*) : Sumber daya (data, bahan baku, peralatan, energi) dari lingkungan yang dikonsumsi dan dimanipulasi oleh suatu sistem.
4. Keluaran (*Output*) : Sumber daya atau produk (informasi, laporan, dokumen, tampilan di layar komputer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.
5. Komponen (*Components*) : Kegiatan-kegiatan atau proses dalam suatu sistem yang mentransformasikan *input* menjadi bentuk setengah jadi ataupun *output*. Komponen ini bisa subsistem dari sebuah sistem.
6. Interface : Tempat dimana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.
7. Penyimpanan (*Storage*) : Area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga diantara komponen sistem yang memungkinkan komponen tersebut bekerja dengan berbagai tingkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama.

2.2.2. Informasi

Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang. Informasi itu sendiri memiliki ciri-ciri seperti berikut (Davis, 1999) [8] :

1. Benar atau salah. Dalam hal ini, informasi berhubungan dengan kebenaran terhadap kenyataan. Jika penerima informasi yang salah mempercayainya, efeknya seperti kalau informasi itu benar.
2. Baru. Informasi benar-benar baru bagi si penerima.
3. Tambahan. Informasi dapat memperbaharui atau memberikan perubahan terhadap informasi yang telah ada.
4. Korektif. Informasi dapat digunakan untuk melakukan koreksi terhadap informasi sebelumnya yang salah atau kurang benar.
5. Penegas. Informasi dapat mempertegas informasi yang telah ada sehingga keyakinan terhadap informasi semakin meningkat.

2.2.3. Sistem Informasi

Ada beberapa pengertian sistem informasi, antara lain yaitu :

Menurut Sutabri (2012:38) “sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu.

Menurut Pratama (2014:10) “sistem informasi merupakan gabungan dari empat bagian utama mencakup perangkat lunak (*software*), perangkat keras (*hardware*), infrastruktur, dan Sumber Daya Manusia (SDM) yang terlatih.

Menurut Al-Bahra Bin Ladjamudin kualitas dari suatu informasi tergantung dari 3 hal yaitu [9] :

1. Relevan (*Relevancy*)
Informasi tersebut benar-benar terasa manfaatnya bagi yang membutuhkan.
2. Akurat (*Accurancy*)
Suatu informasi dikatakan berkualitas jika seluruh kebutuhan informasi tersebut telah tersampaikan (*Completeness*), seluruh pesan telah benar/sesuai (*Correctness*), serta pesan yang disampaikan sudah lengkap atau hanya sistem yang diinginkan oleh user.
3. Tepat Waktu (*Timeliness*)
Proses dapat diselesaikan dengan tepat waktu laporan-laporan yang dibutuhkan dapat disampaikan tepat waktu.

Dari beberapa pendapat diatas, disimpulkan sistem informasi adalah pengiriman data melalui fasilitas sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi [10].

2.2.4. Pendaftaran Pelatihan Kerja

A. Pendaftaran

Menurut Kamus Besar Bahasa Indonesia (2008:306) “Pendaftaran adalah Pencatatan, nama, alamat, dan sebagainya

dalam daftar; perihal mendaftar (mendaftarkan)” [11].

Menurut Magdalena, Rachman (2017:39), “pendaftaran adalah proses pencatatan identitas pendaftar kedalam sebuah media penyimpanan yang digunakan dalam proses pendaftaran” [11].

Menurut Ana dan Suharso (2019:9), “pendaftaran : pencatatan nama, alamat, dan sebagainya dalam daftar; perihal mendaftar (mendaftarkan)” [10]. Dapat disimpulkan pendaftaran adalah proses pencatatan identitas pendaftar kedalam sebuah media penyimpanan yang digunakan dalam proses pendaftaran.

B. Pelatihan Kerja

Menurut Tambunan (2016:354) Pelatihan Kerja adalah keseluruhan kegiatan untuk memberi, memperoleh, meningkatkan serta mengembangkan kompetensi kerja, produktivitas, disiplin sikap dan etos kerja pada tingkat keterampilan dan keahlian tertentu sesuai dengan jenjang dan kualifikasi jabatan atau pekerjaan [10].

C. Email Gateway

Email gateway ini memungkinkan seseorang untuk memulai sebuah skrip dengan mengirim pesan email ke alamat yang ingin kita tuju [12].

2.2.5. Rekayasa Perangkat Lunak

Perangkat lunak adalah perintah/intruksi (program komputer) yang jika diinstruksikan sesuai perintah maka akan menjalankan intruksi tersebut sesuai dengan yang diinstruksikan. Struktur data yang memungkinkan program memanipulasi data dan sebuah informasi secara proporsional. Dokumen yang menggambarkan operasi dan kegunaan program. Sedangkan yang dimaksud dengan Rekayasa Perangkat Lunak adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu *communication*, *requirements capturing* (analisa kebutuhan pengguna), *specification* (menentukan spesifikasi dari kebutuhan pengguna), desain, coding, testing sampai *maintenance* (pemeliharaan sistem) setelah digunakan.

2.2.6. SDLC (System Development Life Cycle)

System Development Life Cycle merupakan salah satu metode atau cara dalam pengembangan sebuah sistem informasi yang populer dan dapat digunakan pada saat sebuah sistem informasi tersebut pertama kalinya dilakukan pengembangan sebuah sistem informasi. Selanjutnya

berkembang dengan adanya beberapa model dan *framework* dalam pengembangan sistem informasi. Salah satu model yang dapat digunakan dalam pengembangan sistem informasi adalah *Rapid Application Development* (RAD) [13].

Menurut James Martin "*Rapid Application Development* (RAD) merupakan pengembangan siklus yang dirancang untuk memberikan pengembangan yang jauh lebih cepat dan hasil yang lebih berkualitas tinggi daripada yang dicapai dengan siklus hidup tradisional. Pendekatan untuk membangun sistem komputer yang menggabungkan *Computer Assisted Software Engineering* (CASE) tools dan teknik, *userdriven prototyping*. RAD meningkatkan kualitas sistem secara drastis dan mengurangi waktu yang diperlukan untuk membangun sistem [4]. Berikut tahapan-tahapannya :

1. Tahap *Requirements Planning*

Tahap ini lebih difokuskan pada kebutuhan pengembangan Sistem Informasi Pengelolaan Pendaftaran Pelatihan Kerja dengan cara mencari informasi mengenai fitur dan kebutuhan sebuah sistem yang akan digunakan oleh pihak BLK Cilacap dengan melalui tahapan wawancara dengan Pegawai bagian pendaftaran BLK Cilacap.

2. Tahap *User Design*

Tahap ini memastikan desain sistem sesuai dengan kebutuhan pengguna berdasarkan data pada tahap sebelumnya. Secara garis besar pada tahap ini keaktifan user yang terlibat menentukan untuk mencapai tujuan karena pada proses ini melakukan proses desain dan melakukan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian desain antara *user* dan desainer sistem. Membuat rancangan dengan *usecase diagram*, *activity diagram*, *sequence diagram*, serta perancangan *input* dan *output* melalui komunikasi antara *user* dengan desainer sistem.

3. Tahap *Construction*

Melakukan implementasi dan pengujian dilakukan berulang secara interaktif pada tahap *construction*. Merupakan tugas dari programmer meneruskan dalam bentuk coding melalui tinjauan pemrograman berdasarkan rancangan sistem yang telah dibuat oleh desainer sistem. Tinjauan pemrograman yang dipakai tergantung dari permintaan *user*, contoh : dengan pemrograman *visual studio*, *java*, *php*, dan lain sebagainya.

4. Tahap *Cutover*

Tahap terakhir ini terdiri dari proses *maintenance* dan *deployment*. Merupakan tahap yang memastikan bahwa suatu sistem

berjalan dengan baik secara berkala serta melakukan pengembangan terhadap sistem.

2.2.7. Pemrograman Berorientasi Objek

Strategi pembangunan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis [14]. Berikut ini adalah beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek :

1. Kelas (*class*)

Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut. 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.

2. Objek (*object*)

Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, suatu 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. Objek mempunyai siklus hidup yaitu diciptakan dimanipulasi, dan dihancurkan.

3. Metode (*method*)

Operasi atau metode pada sebuah kelas hamper 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.

4. Atribut (*attribute*)

Atribut dari sebuah kelas adalah variabel 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

objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama dan sebagainya. Atribut sebaliknya bersifat privat untuk menjaga konsep enkapsulasi.

5. Abstraksi (*abstract*)

Prinsip untuk mempresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*encapsulation*)

Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

7. Pewarisan (*inheritance*)

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

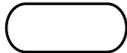
8. Antarmuka (*interface*)

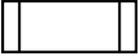
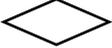
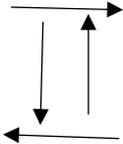
Antarmuka sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah interface dapat diimplementasikan oleh kelas lain.

2.2.8. Flowchart

Ladjamudin (2013:211), flowchart adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. Flowchart merupakan cara penyajian dari suatu algoritma [10].

Tabel 2. 1 Simbol-Simbol *Flowchart*

No.	Simbol	Nama	Keterangan
1.		Terminal	Memulai dan mengakhiri suatu program
2.		Proses	Proses perhitungan atau proses pengolahan data

3.		<i>Predefined Process</i> (sub program)	Permulaan sub program atau proses pengolahan data
4.		<i>Preparation</i>	Proses inialisasi atau pemberian harga awal
5.		<i>Input – Output</i>	Memasukan data maupun menunjukkan hasil dari suatu proses tanpa tergantung dengan jenis peralatannya
6.		<i>Decision</i>	Memilih proses berdasarkan kondisi yang ada
7.		<i>Flow</i>	Menghubungkan antara simbol satu dengan simbol yang lain atau menyatakan jalannya arus dalam suatu proses. Simbol arus ini sering disebut juga dengan <i>connecting line</i> .
8.		<i>Document</i>	Merupakan simbol untuk data yang terbentuk informasi
9.		<i>On Page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada satu halaman

10.		<i>Off Page Conector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada halaman berbeda
11.		<i>Manual Operation</i>	Menunjukkan pengolahan yang tidak dilakukan oleh komputer
12.		<i>Manual Input</i>	Memasukan data secara manual <i>on-line keyboard</i>

2.2.9. UML (Unified Modeling Language)

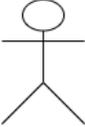
UML (*Unified Modeling Language*) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung [15].

a. *Use Case Diagram*

Use Case diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*). Sehingga pembuatannya lebih dititikberatkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan urutan kejadian (alur). Sebuah *use case diagram* mempresentasikan sebuah interaksi antara aktor dengan sistem.

Tabel 2. 2 Simbol-Simbol *Use Case Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Use case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .

2.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
3.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5.		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

b. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem.

Tabel 2. 3 Simbol-Simbol *Sequence Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Menggambarkan <i>user</i> atau pengguna.

2.		<i>Lifeline</i>	Menyatakan kehidupan suatu objek.
3.		<i>Boundary Class</i>	Menggambarkan sebuah <i>form</i> .
4.		<i>Control Class</i>	Menghubungkan <i>boundary</i> dengan tabel.
5.		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan.

2.2.10. Basis Data (*Database*)

Basis data atau *database* merupakan suatu kumpulan data yang terhubung secara *logical*, terdeskripsi dari data yang dirancang untuk memenuhi kebutuhan informasi dari sebuah organisasi. Komponen-komponen penting sistem basis data adalah : data, perangkat keras (*hardware*), perangkat lunak (*software*), dan pengguna (*user*) yang dikategorikan sebagai *Administrator*, *Programmer* dan *End-user*. *Database Management System* (DBMS) dirancang khusus sebagai aplikasi yang berinteraksi dengan pengguna, dengan aplikasi lain, dan dengan basis data (*database*) itu sendiri untuk menangkap dan menganalisa data [16].

A. DBMS (*Database Management System*)

DBMS atau sering disebut sebagai Sistem Manajemen Basis Data adalah perangkat lunak khusus yang digunakan untuk membuat, mengontrol dan mengelola basis data [16]. DBMS menyediakan beberapa fasilitas sebagai berikut [17] :

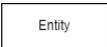
1. DDL (*Data Definition Language*) merupakan sebuah metode *Query SQL* yang berguna untuk mendefinisikan data pada sebuah *database*. Adapun *Query* yang dimiliki adalah :
 - a. CREATE : Digunakan untuk pembuatan *table* dan

- database.*
- b. DROP : Digunakan untuk penghapusan *table* dan *database.*
 - c. ALTER : Digunakan untuk perubahan struktur *table* yang dibuat.
2. DML (*Data Manipulation Language*) merupakan bentuk bahasa basis data yang berguna untuk melakukan manipulasi dan pengambilan data pada suatu basis data. Adapun *Query* yang termasuk didalamnya adalah :
 - a. INSERT : Digunakan untuk penginputan data pada *table database.*
 - b. UPDATE : Digunakan untuk perubahan data pada *table database.*
 - c. DELETE : Digunakan untuk penghapusan data pada *table database.*
- A. ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram atau ERD adalah sebuah diagram structural yang digunakan untuk merancang sebuah database. Sebuah ERD mendeskripsikan data yang akan disimpan dalam sebuah sistem maupun batasannya. Komponen utama yang terdapat di dalam sebuah ERD adalah *entity set*, *relationship set*, dan *constraints*. Simbol-simbol yang digunakan dalam ERD dijelaskan pada tabel 2.4.

Tabel 2. 4 Simbol-Simbol ERD

No.	Simbol	Nama	Keterangan
1.		<i>Relasi/Hubungan</i>	Hubungan yang terjadi antara 1 entitas atau lebih yang tidak mempunyai fisik tetapi hanya sebagai konseptual. Dan untuk mengetahui jenis hubungan yang ada antara 2 file.
2.		<i>Atribut</i>	Atribut ialah karakteristik dari entitas atau relasi yang menyediakan penjelasan detail tentang entitas atau

			relasi tersebut. Berfungsi untuk memperjelas atribut yang dimiliki oleh sebuah entitas. Atribut memiliki bentuk lingkaran lebih tepatnya <i>elips</i> .
3.		<i>Alur</i>	Alur memiliki fungsi untuk menghubungkan atribut dengan entitas dan entitas dengan relasi. Dan berbentuk garis.
4.		<i>One to One</i>	Digunakan untuk menghubungkan antar entitas dengan hubungan satu ke satu.
5.		<i>One to Many</i>	Digunakan untuk menghubungkan antar entitas dengan hubungan satu ke banyak atau sebaliknya.
6.		<i>Entitas (Entity)</i>	Entitas ialah suatu objek yang dapat dibedakan dengan objek lainnya. Entitas berfungsi untuk memberikan identitas pada entitas yang memiliki label dan nama. Entitas memiliki bentuk persegi panjang.