



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1. Perancangan

Al-Bahra Bin Ladjamudin dalam bukunya yang berjudul Analisis dan Desain Sistem Informasi, adalah Tahapan perancangan (*design*) memiliki tujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik. (Al-Bahra Bin Ladjamudin, 2005)

Mohamad Subhan dalam bukunya yang berjudul Analisa Perancangan Sistem mengungkapkan perancangan adalah proses pengembangan spesifikasi baru berdasarkan rekomendasi hasil analisis sistem. (Subhan, 2012)

Dengan demikian dapat disimpulkan perancangan merupakan proses mendesain sistem berdasarkan hasil analisis sistem untuk menyelesaikan suatu masalah.

2.2. *Business Canvas Analysis*

Bisnis model menejelaskan mengenai dasar pemikiran bagaimana sebuah bisnis diciptakan, diberikan, dan ditangkap nilainya (Osterwalder & Pigneur, 2010). Oster & Pigneur membuat sebuah pendekatan bagi para pebisnis untuk membangun dan mengembangkan bisnis mereka. *Nine Building Blocks* terdiri dari: *Value Propositions*, *Customer Segments*, *Customer Relationship*, *Channels*, *Key Resources*, *Key Activity*, *Key Partnership*, *Cost Structure*, dan *Revenue Streams*. Analisis ini memiliki fungsi yaitu:

1. Salah satu alat untuk membantu kita melihat lebih akurat bagaimana rupa usaha yang sedang atau kita akan jalani.
2. Melihat bisnis dari gambaran besar namun tetap lengkap dan mendetail untuk melihat elemen-elemen kunci yang terkait dengan bisnis.
3. Mengevaluasi masing-masing elemen kunci untuk memudahkan menganalisis apa yang kurang tepat, dan menentukan langkah untuk mencapai tujuan bisnis.

Dengan demikian dapat disimpulkan *business canvas analysis* merupakan suatu pendekatan untuk melihat bisnis secara mendetail berdasarkan *value propositions*, *customer segments*, *customer relationship*, *channels*, *key resources*, *key activity*, *key partnership*, *cost structure* dan *revenue streams* untuk menentukan langkah-langkah untuk mencapai tujuan bisnis.

2.3.Return On Investment Analysis

Hasil pengembalian investasi atau yang lebih dikenal dengan nama *Return On Investment (ROI)* atau *return on total assets* menurut Kasmir (2012) adalah rasio yang menunjukkan hasil (*return*) atas jumlah aktiva yang digunakan dalam perusahaan. *ROI* juga merupakan suatu ukuran tentang efektivitas manajemen dalam mengelola investasinya.

Sedangkan menurut Fahmi (2012) *ROI* merupakan suatu pendekatan untuk melihat sejauh mana investasi yang telah ditanamkan mampu memberikan pengembalian keuntungan sesuai dengan yang diharapkan, dan investasi tersebut sebenarnya sama dengan asset perusahaan yang ditanamkan.

Dengan demikian dapat disimpulkan *ROI* adalah suatu pendekatan untuk melihat efektivitas investasi dengan menunjukkan hasil investasi yang ditanamkan memberikan pengembalian keuntungan sesuai dengan yang diharapkan.

2.4. Database

basis data (*database*) adalah kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di simpanan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasinya. (Jogiyanto, 2005)

Pada bukunya Sutanta menyebutkan *database* adalah suatu kumpulan dari item data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu. (Sutanta, 2004)

Dengan demikian dapat disimpulkan database adalah suatu kumpulan dari *data* yang saling berhubungan yang dan diorganisasikan berdasarkan struktur tertentu dan tersimpan di luar komputer dan digunakan oleh perangkat lunak tertentu untuk memanipulasinya.

2.5. MySQL

MySQL menurut Arief dalam bukunya adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengolahan datanya. (Arief, 2011)

Sulhan menyebutkan bahwa *MySQL* merupakan perangkat lunak yang digunakan untuk membangun database yang sering digunakan di lingkungan *linux*. *MySQL* merupakan *software open source* yang berarti *free* untuk digunakan. (Sulhan, 2007)

Dengan demikian dapat disimpulkan *MySQL* merupakan salah satu jenis database *open source server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dari pengolahan datanya.

2.6. *HTML*

Menurut Suyanto *HTML* merupakan salah satu *form* yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman *web*. (Suyanto, 2007)

Sedangkan menurut Siberio *HTML (Hyper Text Markup Language)* adalah bahasa yang digunakan pada dokumen *web* sebagai bahasa untuk pertukaran dokumen *web*. (Siberio, 2012)

Dengan demikian dapat disimpulkan *HTML* merupakan Bahasa yang digunakan pada dokumen *web* yang berupa *form* sebagai Bahasa untuk pertukaran dokumen *web*.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Example</title>
5     <link rel="stylesheet" href="sty
6   </head>
7   <body>
8     <h1>
9       <a href="/">Header</a>
10    </h1>
11    <nav>
12      <a href="one/">One</a>
13      <a href="two/">Two</a>
14      <a href="three/">Three</a>
15    </nav>
```

Gambar 2. 1 Contoh Bahasa Pemrograman *HTML*

2.7. PHP

Luke Welling menyebutkan bahwa *PHP* adalah Bahasa *server-side –scripting* yang menyatu dengan *HTML* untuk membuat halaman web yang dinamis.

Sedangkan menurut Agus Saputra *PHP* atau yang memiliki kepanjangan *PHP Hypertext Preprocessor* merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu website dinamis. (Agus Saputra, 2011)

Dengan demikian dapat disimpulkan *PHP* adalah Bahasa *server-side-scripting* yang yang menyatu dengan *HTML* untuk membuat halaman *web* yang dinamis.

```
1 <?php
2 $str_isset = "The isset example";
3 $bol_isset = isset($str_isset);
4
5 If ($bol_isset){
6     echo "The variable is set </BR>";
7 }
8 else {
9     echo "The variable is not set </BR>";
10 }
11 //Using unset function
12 unset ($str_isset);
13 $bol_isset = isset($str_isset);
14 If ($bol_isset){
15     echo "The variable is set";
16 }
17 else {
18     echo "The variable is not set";
19 }
20 ?>
21
```

Gambar 2. 2 Contoh Bahasa Pemrograman *PHP*

2.8. PhpMyAdmin

Menurut Delisle (2012) phpMyAdmin merupakan *web interface* yang bersifat *open source* yang menangani administrasi dari MySQL. phpMyAdmin dapat melakukan beragam tugas seperti membuat, memodifikasi, atau menghapus *database*, tabel, kolom, maupun baris. phpMyAdmin juga dapat meng-eksekusi *SQL statement* serta mengelola *user* beserta *permission* nya.

2.9. *SDLC*

Menurut Turban *System Development Life Cycle (SDLC)* atau siklus hidup Pengembangan Sistem adalah metode pengembangan sistem tradisional yang digunakan sebagian besar organisasi saat ini. *SDLC* adalah kerangka kerja (*framework*) yang terstruktur yang berisi proses-proses sekuensial di mana sistem informasi dikembangkan. (Turban, 2003)

System Development Life Cycle (SDLC) adalah suatu pendekatan yang memiliki tahap atau bertahap untuk melakukan analisa dan membangun suatu rancangan sistem dengan menggunakan siklus yang lebih spesifik terhadap kegiatan pengguna (Kendall & Kendall, 2006).

2.10. *Waterfall Model*

Seperti yang telah diungkapkan oleh Rosa dan Shalahuddin (2011) metode pengembangan sistem *waterfall model*. Model *SDLC* air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linier*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau urut dimulai dari analisis, desain, pengkodean, pengujian dan tahap *support*

Sedangkan menurut Pressman (2005) *waterfall model* adalah pola model tertua pada model pengembangan *software* yang bertujuan untuk memberikan saran yang sistematis kepada pengembang *software* yang diawali dari kebutuhan pelanggan yang kemudian diteruskan hingga berpuncak pada dukungan untuk penyelesaian *software*. Berikut merupakan tahapan-tahapan dari *waterfall model*:

1. *System Requirement*

Seluruh kebutuhan *software* didapatkan pada fase ini. Pada tahap ini informasi dapat didapatkan melalui wawancara, survei, atau diskusi yang kemudian informasi tersebut dianalisis untuk mendapatkan dokumentasi kebutuhan pengguna untuk digunakan pada tahapan selanjutnya.

2. *System Design*

Tahap ini adalah tahapan sebelum memulai pemrograman. Tahap ini bertujuan untuk memberikan gambaran tentang apa yang harus dikerjakan dan bagaimana tampilannya.

3. *Implementation & Unit Testing*

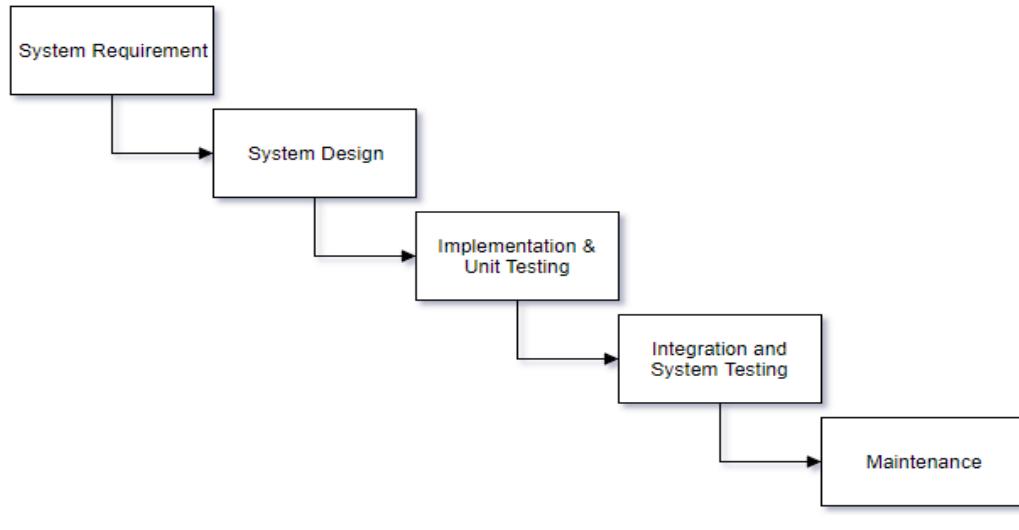
Tahapan ini adalah tahapan pemrograman. Selain itu juga pada tahapan ini juga dilakukan *testing* terhadap modul-modul yang dibuat apakah sudah berfungsi sesuai yang diharapkan atau belum.

4. *Integration and System Testing*

Tahapan ini dilakukan penggabungan model yang telah dibuat dan dilakukan pengujian pada tahap sebelumnya dan dilakukan *testing* untuk mengetahui sistem yang dibuat sudah berjalan dengan baik.

5. *Operation And Maintenance*

Tahap ini merupakan tahap terakhir pada *waterfall model*. *Software* yang sudah dibuat sudah beroperasi dan dilakukan *maintenance* agar *software* dapat berjalan dengan baik.



Gambar 2. 3 Tahapan Waterfall

2.10.1. Kelebihan Metode Waterfall

1. Merupakan model pengembangan yang sederhana dan mudah diaplikasikan
2. Pengerjaan proyek sistem akan terjadwal dengan baik dan mudah dikontrol karena tiap tahapan memiliki 1 tujuan dan pengembangan terjadi hanya dalam satu fase pada setiap tahapan.

2.10.2. Kekurangan Metode Waterfall

1. Waterfall model bersifat kaku sehingga penanganan perubahan pada saat proses sedang berlangsung menjadi lebih sulit.
2. Karena dilakukan pertahap, maka metode waterfall memakan waktu lebih lama.
3. Semua kebutuhan harus sudah terdefinisi sejak awal, sehingga membuat metode *waterfall* cenderung kaku/kurang fleksibel.

2.11. Rapid Application Development

Martin mengungkapkan bahwa *RAD* adalah sebuah model proses pengembangan perangkat lunak sekuensial *linear* yang menekankan siklus pengembangan yang sangat pendek.

Model *RAD* ini merupakan pengembangan dari metode sekuensial *linear* di mana perkembangan cepat dicapai dengan menggunakan model pendekatan berbasis komponen. Jika kebutuhan *user* dapat dipahami dengan baik, proses *RAD* memungkinkan pengembangan sistem dalam periode waktu yang lebih singkat.
(Pressman, 2002)

Pada bukunya Noertjahyana (2002) metode *RAD* memiliki 3 tahapan utama dalam pengembangannya, yaitu:

1. Requirement Planning

Pada tahapan ini, dilakukan pertemuan dengan *user* untuk melakukan identifikasi tujuan dari aplikasi atau sistem dan melakukan identifikasi kebutuhan informasi untuk mencapai tujuan. Pada tahap ini hal terpenting adalah adanya keterlibatan dari kedua belah pihak, bukan hanya sekedar persetujuan akan proposal yang dibuat.

2. Design Workshop

Pada tahapan ini adalah melakukan proses desain dan melakukan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian desain antara *user* dan analis.

3. Implementation

Setelah desain dari sistem yang akan dibuat sudah disetujui baik itu oleh *user* maka pada tahap ini *programmer* mengembangkan *prototype* menjadi suatu program.

2.12. *Structured Analysis and Design (SSAD)*

Jogiyanto (2005) menjelaskan bahwa *SSAD* merupakan pendekatan struktur dimulai dari awal tahun 1970 disebut dengan pendekatan terstruktur (*structured approach*). Pendekatan terstruktur dilengkapi dengan alat-alat (*tools*) dan teknik-teknik (*techniques*) yang dibutuhkan dalam pengembangan sistem, sehingga hasil akhir dari sistem yang dikembangkan akan didapatkan sistem yang strukturnya didefinisikan dengan baik dan jelas.

Pendekatan secara terstruktur adalah mengenalkan penggunaan alat-alat dan teknik-teknik untuk mengembangkan sistem yang terstruktur. Teknik terstruktur, merupakan pendekatan formal untuk memecahkan masalah-masalah dalam aktivitas bisnis menjadi bagian-bagian kecil yang dapat diatur dan berhubungan untuk kemudian dapat disatukan kembali menjadi satu kesatuan yang dapat dipergunakan untuk memecahkan masalah. Tujuan pendekatan terstruktur adalah agar pada akhir pengembangan perangkat lunak dapat memenuhi kebutuhan user, dilakukan tepat waktu, tidak melampaui anggaran biaya, mudah.

Ciri-ciri utama yang mendukung pendekatan terstruktur menurut Jogiyanto (2005):

1. Dilakukan secara iterasi : dengan iterasi akan didapat hasil yang lebih baik, terlalu banyak iterasi juga akan menurunkan hasilnya dan

menunjukkan bahwa tahap sebelumnya tidak dilakukan dengan baik.

2. Merancang berdasar modul : modularisasi adalah proses yang membagi suatu sistem menjadi beberapa modul yang dapat beroperasi secara *independent*.
3. Bekerja dengan pendekatan *top-down* : dimulai dari level atas (secara global) kemudian diuraikan sampai ke tingkat modul (rinci).
4. Kegiatan dilakukan secara paralel : pengembangan subsistem-subsistem dapat dilakukan secara paralel, sehingga akan memperpendek waktu pengembangan sistem.

Beberapa *tools* yang digunakan pada pendekatan pengembangan sistem secara terstruktur seperti:

1. DFD (*Data Flow Diagram*)
2. Kamus Data
3. ERD (*Entity Relationship Diagram*)
4. STD (*State Transition Diagram*)

2.13. *Entity Relationship Diagram (ERD)*

Brady & Loonam menyebutkan *entity relationship diagram* merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh sistem analis dalam tahap analisis persyaratan proyek pengembangan sistem. *ERD* bersama-sama dengan *detail* pendukung merupakan model data yang digunakan sebagai spesifikasi untuk *database* (Brady & Looman, 2010).

Pada buku yang ditulis oleh Mata-Toledo dan Cushman *Entity Relationship Diagram (ERD)* merupakan representasi grafis dari logika *database* dengan

menyertakan deskripsi detail mengenai seluruh entitas (*entity*), hubungan (*relationship*), dan batasan (*constraint*). (Mata-Toledo dan Cushman, 2007)

Dengan demikian dapat disimpulkan *ERD* merupakan model grafis untuk analisis logika *database* untuk mendeskripsikan hubungan antar entitas pada suatu *database*.

2.14. Data Flow Diagram (DFD)

Pengertian *Data Flow Diagram (DFD)* menurut Jogiyanto Hartono adalah *Diagram* yang menggunakan notasi simbol untuk menggambarkan arus *data system*. (Jogiyanto 2005).

Menurut Jogiyanto pada tahun 2005 dalam bukunya Basis Data ada beberapa symbol yang digunakan pada *DFD* yaitu:

1. *External Entity*

External Entity merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lain yang berada pada lingkungan luarnya yang memberikan *input* dan menerima *output* dari sistem.

2. *Data Flow*

Arus Data (*data flow*) di *DFD* diberi simbol suatu panah. Arus data ini mengalir di antara proses, simpan data dan kesatuan luar. Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

3. Process

Process menunjukkan pada bagian yang mengubah input menjadi output, yaitu menunjukkan bagaimana satu atau lebih input diubah menjadi beberapa output. Setiap proses mempunyai nama, nama dari proses ini menunjukkan apa yang dikerjakan proses.

4. Data Store

Data Store merupakan simpanan dari *data* yang dapat berupa suatu *file* atau *database* pada sistem komputer.

Keterangan	DeMarco and Yourdan Simbol	Gane and Sarson Simbol
Entitas Luar		
Proses		
Aliran data (data flow)		
Simpan data		

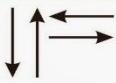
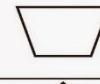
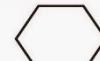
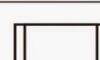
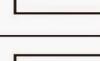
Gambar 2. 4 Simbol-Simbol DFD (Sumber: <http://csa04-stmik-samarinda.blogspot.co.id/2014/04/simbol-dfd.html>)

2.15. Flow Chart Diagram

Menurut Jogiyanto (2005) *flow chart diagram* adalah bagan (*chart*) yang menunjukkan alur atau arus (*flow*) di dalam *program* atau prosedur sistem secara logika.

Menurut Pahlevy (2010) *flow chart diagram* merupakan sebuah gambaran dalam bentuk *diagram* alir dari algoritma-algoritma dalam suatu program, yang menyatakan arah alur program tersebut secara logika.

Menurut Pahlevy *flow chart diagram* merupakan sebuah gambaran dalam bentuk *diagram* alir dari algoritma-algoritma dalam suatu program, yang menyatakan arah alur program tersebut.

	Flow Direction symbol Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.
	Terminator Symbol Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.
	Processing Symbol Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer
	Simbol Manual Operation Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer
	Simbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada.
	Simbol Input-Output Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya
	Simbol Manual Input Simbol untuk pemasukan data secara manual on-line keyboard
	Simbol Preparation Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	Simbol Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure
	Simbol Display Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	Simbol disk and On-line Storage Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	Simbol magnetik tape Unit Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	Simbol Punch Card Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	Simbol Dokumen Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

Gambar 2. 5 Contoh Simbol *Flow Chart Diagram* (Sumber:
<http://dewacoding21.blogspot.co.id/2016/04/pengertian-flowchart-berikut-simbol.html>)

2.16. Penelitian Terdahulu

Berikut adalah beberapa penelitian terdahulu yang dijadikan landasan dan tolak ukur pada penelitian ini, yaitu:

Table 2. 1 Penelitian Terdahulu

No	Judul	Masalah	Solusi	Kesimpulan
1.	Sistem informasi lowongan dan lamaran pekerjaan berbasis web menggunakan ASP.NET	Kesulitan pelamar kerja dalam pencarian informasi lowongan kerja melalui media cetak dan mendatangi perusahaan	Membangun sistem informasi lowongan dan lamaran pekerjaan berbasis web	Aplikasi ini dapat membantu pelamar mendapatkan informasi lowongan dengan menjadi perantara antara pelamar dan perusahaan
2.	Sistem informasi penyedia lowongan kerja berbasis web.	Kurang maksimalnya Unit Pengembangan karir ITS dalam mengelola <i>input</i> , proses, <i>output</i> lowongan kerja, sehingga penyampaian informasi lowongan kurang maksimal	Membangun sistem informasi lowongan kerja yang dapat memudahkan Unit Pengembangan Karir ITS	Sistem informasi ini membantu mahasiswa dan penyedia lowongan karena proses penyampaian informasi lowongan dapat diproses dengan cepat
3.	Rancangan <i>Portal Lapangan Pekerjaan</i> berbasis web untuk membantu pencari kerja dan instansi dalam rangka pemenuhan akan kebutuhan SDM	Masalah pengangguran yang sangat krusial 2006 yaitu 40,1 juta orang menganggur dari 106,9 juta angkatan kerja	Membuat portal web untuk menambah informasi sekaligus membuka jalan bagi pencari kerja mendapat kerja	portal lapangan pekerjaan ini akan mempermudah pencari kerja dalam memperoleh informasi lowongan pekerjaan dan

				pencari kerja dapat melakukan pelamaran kerja secara <i>online</i> .
--	--	--	--	--

2.16.1. Sistem Informasi Lowongan dan Lamaran Pekerjaan Berbasis Web Menggunakan ASP.NET

Sistem informasi ini dibuat oleh Swarry Chalatra dan Meliana Christianti. yang memiliki tujuan untuk merancang sebuah sistem informasi untuk lowongan dan lamaran pekerjaan menggunakan bahasa pemrograman *ASP.NET* yang dimana dalam pembangunan sistem informasi diharapkan dapat membantu para pelamar pekerjaan dalam mendapatkan informasi tentang lowongan pekerjaan yang akan ditampilkan di *website* serta dapat memudahkan perusahaan (Divisi *Human Resource*) dalam mencari sumber daya manusia serta memudahkan perusahaan untuk mengelola profil perusahaan karena disediakan fitur *editor* untuk mengolah informasi profil perusahaan dalam bentuk *pdf*. Pada pembangunan sistem ini digunakan pendekatan tersetruktur yang digambarkan melalui *DFD*, *Flow Chart*, dan *ERD* sebagai *design* untuk pembangunan sistem informasi ini.

Kesimpulan dari pembangunan sistem informasi lowongan dan lamaran pekerjaan ini adalah:

1. Aplikasi ini dapat membantu pelamar untuk mendapatkan informasi tentang lowongan pekerjaan yang ditampilkan dalam *website*.

2. Aplikasi ini menyediakan *editor* yang dapat mengelola *data* profil perusahaan dan persyaratan tambahan untuk perusahaan yang akan disimpan dalam bentuk *pdf*.
3. Aplikasi ini memberikan informasi mengenai lowongan pekerjaan yang tersedia, deskripsi pekerjaan dan lokasi penempatan kerja dari berbagai perusahaan yang sudah terdaftar dalam *website*.
4. Aplikasi ini menyediakan *form* bagi pelamar untuk melakukan registrasi yang harus diisi jika ingin menggunakan *website* ini.
5. Aplikasi ini mengizinkan pihak pelamar yang sudah melakukan registrasi untuk mencari pekerjaan yang diminati dan mendaftarkan diri pada lowongan pekerjaan yang dipilih.
6. Aplikasi ini menjadi perantara yang memberikan informasi kepada pihak perusahaan melalui pesan hasil pendaftaran dengan menyeleksi *data* pelamar yang masuk dalam persyaratan yang diberikan perusahaan.

2.16.2. Sistem Informasi Penyedia Lowongan Kerja Berbasis Web

Sistem informasi ini dibuat oleh Bagus Arianto, I. Made Kamisutara. Dan Tubagus Purworusmiadi. Sistem ini dibuat dengan tujuan untuk membantu Unit Pengembangan Karir ITS dalam memberikan fasilitas informasi lowongan kerja bagi mahasiswa. Sebelumnya Unit Pengembangan Karir ITS masih kurang dalam memberikan informasi lowongan kerja. Informasi lowongan kerja tersebut dapat dilihat melalui *web* dan tertempel di papan pengumuman yang ada di kantor SAC

(Unit Pengembangan Karir) serta dengan adanya sistem informasi ini Unit Pengembangan Karir ITS dapat bekerja sama dengan lebih banyak perusahaan-perusahaan sehingga peluang yang diberikan kepada mahasiswa lebih besar karena perusahaan-perusahaan yang lebih bervariasi. Selain itu, Sistem Informasi ini memiliki tujuan yaitu memberikan kemudahan bagi pelamar kerja khususnya mahasiswa ITS yang akan mencari kerja. Pada pembangunan sistem ini menggunakan metode pengembangan dengan model yang sederhana yaitu *waterfall model* dan menggunakan pendekatan terstruktur yang digambarkan dengan *DFD*, *Flowchart*, dan *ERD*.

Kesimpulan dari pembangunan sistem informasi penyedia lowongan kerja berbasis web ini adalah sistem informasi ini mudah digunakan dan bermanfaat bagi mahasiswa maupun bagi penyedia lowongan kerja. Dengan adanya sistem ini pelamar tidak perlu susah-susah lagi dalam mencari lowongan kerja. Dengan hanya *input data* saja, maka pelamar kerja dapat mendapatkan informasi lowongan pekerjaan yang lebih lengkap dibandingkan secara *manual* serta mendapatkan informasi lowongan kerja yang sesuai dengan kualifikasi dari pelamar tersebut dengan cepat.

2.16.3. Rancangan *Portal Lapangan Pekerjaan* berbasis *web* untuk membantu pencari kerja dan instansi dalam rangka pemenuhan akan kebutuhan SDM

Sistem informasi ini dibuat oleh Pujiono & M. Arief Soleman. Sistem ini dibuat dengan tujuan untuk membantu menekan atau mengurangi tingkat penganguran yang sangat tinggi pada tahun 2006. Dengan adanya sistem

informasi ini diharapkan informasi lowongan pekerjaan yang dapat lebih mudah ditemukan oleh pencari kerja serta sistem informasi ini dibuat untuk menjadi jembatan perusahaan dan pencari kerja, sehingga proses rekrut karyawan menjadi lebih mudah dan cepat.

Kesimpulan dari pembuatan sistem ini adalah *portal* lapangan pekerjaan ini akan mempermudah pencari kerja dalam memperoleh informasi lowongan pekerjaan dan pencari kerja dapat melakukan pelamaran kerja secara *online*.

2.16.4. Kesimpulan Penelitian Terdahulu

Kesimpulan yang dapat diambil dari penelitian terdahulu yang digunakan menjadi patokan untuk pembangunan *platform* pencarian karyawan ini adalah dengan membuat *platform* pencarian karyawan berbasis *web* tentu dapat membantu pelamar kerja untuk mempermudah mencari pekerjaan dikarena pelamar kerja dapat dipermudah dalam mencari informasi lowongan pekerjaan karena dengan pembuatan *platform* ini tentu ini dapat pelamar kerja akan mendapatkan peluang yang lebih besar dikarenakan perusahaan yang lebih banyak atau bervariasi. Selain membantu pelamar kerja pembangunan *platform* pencarian karyawan berbasis *web* ini juga membantu perusahaan dalam mencari SDM (Sumber Daya Manusia) karena dengan pembuatan *platform* ini tentu memperluas ruang lingkup SDM yang dapat dipekerjaan sesuai dengan kualifikasi yang diinginkan oleh perusahaan-perusahaan yang terdaftar didalam *platform* ini.

Dari penelitian sebelum nya pembangunan *platform* ini mengadopsi dari penelitian ini menggunakan kerja sama yang sebelumnya hanya dengan unit

pengembangan karir dan dikembangkan dengan membangun kerja sama dengan beberapa kampus untuk mendapatkan *data* mahasiswa semester akhir dan lulusan universitas tersebut sehingga jangkauan dari *platform* yang dibangun ini lebih luas sehingga dapat membantu para mahasiswa dan lulusan universitas yang bekerja sama untuk mendapatkan informasi lowongan kerja melalui *platform* ini.

