



BAB II
TINJAUAN PUSTAKA DAN
LANDASAN TEORI

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Berikut beberapa penelitian mengenai bursa kerja khusus diantaranya adalah penelitian dengan judul “ *Pengembangan Sistem Informasi Bursa Kerja Khusus (BKK) Berbasis Web dengan PHP dan MySQL di SMK Negeri 2 Wonosari* “ pada tahun 2017. Dari penelitian milik David hasil yang didapatkan adalah mengatasi penyaluran informasi lowongan kerja yang belum terpusat dengan sistem informasi BKK yang berbasis web. Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode *Research & Development (R&D)* dengan *framework* Laravel. Perbedaan penelitian ini dengan penelitian yang akan dibuat yaitu penelitian ini tidak menginformasikan hasil tes kepada pelamar[3].

Penelitian serupa dilakukan dengan judul “*Sistem Informasi Pendaftaran Seleksi Kerja Berbasis Web Pada BKK (Bursa Kerja Khusus) Tunas Insan Karya SMK Negeri 2 Banyumas*”. Sistem informasi yang dikembangkan dalam penelitian ini adalah dengan berbasis web. Hasil dari penelitian ini adalah membuat sistem pendaftaran yang tadinya dilakukan secara manual, yang menghabiskan banyak waktu dan jarak tempuh menuju kantor BKK menjadi pendaftaran dengan sistem. Metode pengembangan yang digunakan dalam penelitian ini adalah metode *SDLC (Software Development Life Cycle)* dengan model *Waterfall* yaitu analisa, rancangan, pengkodean, dan implementasi. Perbedaan penelitian ini dengan penelitian yang akan dibuat yaitu pada sistem ini tidak terdapat fitur kontrol untuk mengelola pembagian kuota pendaftar[4].

Penelitian lainnya dengan judul “*Pengembangan Sistem Informasi Berbasis Web untuk Peningkatan Kinerja Unit Bursa Kerja Khusus SMK Negeri 1 Tanjung Raya*. Penelitian yang menghasilkan peningkatan kinerja pada unit BKK berjalan lebih baik karena adanya sistem informasi untuk peningkatan kinerja unit BKK. Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode *Sistem Development Life Cycle (SDLC)*. Perbedaan penelitian ini

dengan penelitian yang akan dibuat yaitu pada sistem ini tidak menginformasikan hasil tes kepada pelamar[5].

Penelitian ini bermaksud membangun sebuah sistem informasi Bursa Kerja Kusus berbasis web di SMK YPE Kroya. Sistem informasi Bursa Kerja Kusus berbasis web di SMK YPE Kroya menggunakan bahasa pemrograman PHP, Native dan database MySQL. Metode penelitian yang digunakan adalah metode pengembangan perangkat lunak dengan model *waterfall*. Kelebihan sistem yang akan dibuat dengan penelitian sebelumnya yaitu tersedianya fitur untuk validasi terhadap pencari kerja, sehingga dimungkinkan data pelamar yang dimasukkan merupakan data yang valid. Selain itu, penelitian ini menginformasikan hasil tes kepada pelamar. Kelebihan selanjutnya, penelitian ini terdapat fitur kontrol untuk mengelola pembagian kuota pendaftar.

2.2 Landasan Teori

Landasan teori berisi hal-hal atau teori-teori yang berkaitan dengan permasalahan dan ruang lingkup permasalahan sebagai landasan dalam pembuatan laporan ini.

2.2.1 Sistem Informasi

Sistem informasi merupakan suatu sistem yang dibuat oleh manusia, yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Sistem informasi di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Sistem informasi menerima masukan data dan instruksi, mengolah data tersebut sesuai instruksi, dan mengeluarkan hasilnya. Model dasar sistem menghendaki agar masukan, pengolahan, dan keluaran tiba pada saat bersamaan, yang sebaiknya sesuai untuk sistem pengolahan informasi yang paling sederhana, di mana semua masukan tersebut tiba pada saat bersamaan, meskipun hal tersebut jarang terjadi. Fungsi pengolahan informasi sering membutuhkan data yang telah dikumpulkan dan diolah dalam periode waktu sebelumnya. Oleh karena itu ditambahkan sebuah penyimpanan data file (data file *storage*) ke dalam model sistem informasi. Dengan

demikian, kegiatan pengolahan 9 tersedia baik bagi data baru maupun data yang telah dikumpulkan dan disimpan sebelumnya[6].

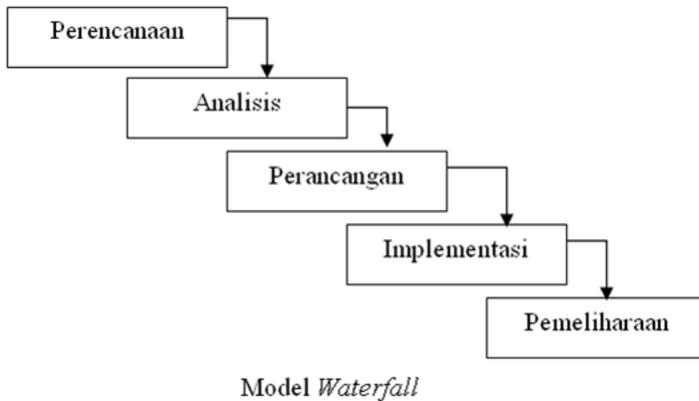
2.2.2 Rekayasa Web

Rekayasa Web merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin[13].

Berikut adalah metode dan *tools* yang digunakan :

1. Metode

Metode dapat diartikan sebagai menyediakan cara bagaimana secara teknis membangun perangkat lunak yang harus berada pada sebuah komitmen dasar menuju kualitas. Metode yang digunakan pada penelitian ini adalah metode *waterfall*. Metode *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut[2]. Metode *waterfall* dapat dilihat pada gambar 2.1.



Gambar 2.1 Pemodelan *waterfall*

Adapun tahap-tahapnya yaitu :

- a. Analisa Kebutuhan Perangkat Lunak
Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.
- b. Desain atau Perancangan Sistem
Proses perancangan sistem membagi persyaratan dalam perancangan sistem perangkat keras dan perangkat lunak. Pada tahap ini, hasil analisa kebutuhan perangkat lunak pada sistem ini akan dideskripsikan ke dalam beberapa diagram, antara lain *use case diagram*, *sequence diagram*, dan *flowchart*.
- c. Penulisan Kode Program
Desain atau perancangan sistem harus ditranslasikan ke dalam program perangkat lunak. Pada tahap ini, hasil desain diimplementasikan ke dalam kode atau bahasa yang dimengerti oleh komputer agar program komputer sesuai dengan hasil desain yang telah dibuat.
- d. Pengujian
Pengujian fokus pada perangkat lunak secara dari segi *logic* dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.
- e. Pemeliharaan
Pemeliharaan mencakup koreksi dari tahap sebelumnya, perbaikan, atau implementasi dari persyaratan-persyaratan baru yang ditambahkan, tetapi tidak untuk membuat program perangkat lunak yang baru.

Terkait dengan pengujian program, pengujian yang akan digunakan adalah pengujian *black-box*. Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan rekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black-box*

bukan merupakan alternatif dari teknik *white-box*, tetapi merupakan pendekatan komplementer yang memungkinkan besar mampu mengungkap kelas kesalahan daripada metode *white-box*. Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut [14] :

1. Fungsi-fungsi yang tidak benar atau hilang,
2. Kesalahan *interface*,
3. Kesalahan dalam struktur data atau akses *database* eksternal,
4. Kesalahan kinerja,
5. Inisialisasi dan kesalahan terminal.

Tidak seperti pengujian *white-box*, yang dilakukan pada saat awal proses pengujian, pengujian *black-box* cenderung diaplikasikan selama tahap akhir pengujian. Karena pengujian *black-box* memperhatikan struktur kontrol, maka perhatian berfokus pada domain informasi.

Pengujian di desain untuk menjawab pertanyaan-pertanyaan berikut[15].

1. Bagaimana validitas fungsional diuji?
2. Kelas *input* apa yang akan *test case* menjadi baik?
3. Apakah sistem sangat sensitif terhadap harga *input* tertentu?
4. Bagaimana batasan dari suatu data diisolasi?
5. Kecepatan data apa dan volume data apa yang dapat ditolelir oleh sistem?
6. Apa pengaruh kombinasi tertentu dari data terhadap operasi sistem?

Pengujian *black-box* ini terdapat beberapa proses. Proses-proses yang ada dalam pengujian ini diantaranya :

- a. Menganalisa kebutuhan dan spesifikasi dari perangkat lunak.
- b. Pemilihan jenis *input* yang memungkinkan menghasilkan *output* dengan benar serta jenis *input* yang memungkinkan *output* salah pada perangkat lunak yang sedang diuji.
- c. Menentukan *output* untuk satu jenis *input*.
- d. Pengujian dilakukan dengan *input-input* yang telah benar-benar diseleksi.
- e. Melakukan pengujian

Perbandingan *output* yang dihasilkan dengan *output* yang diharapkan. Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan[7].

2. *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah hasil kerja dari konsorium berbagai organisasi yang berhasil dijadikan sebagai standar baku dalam *Object Oriented Analysis* dan *Design (OOAD)*. Macam-macam dari *Unified Modeling Language (UML)* antara lain : *use case diagram*, *sequence diagram* dan *class diagram*[16].

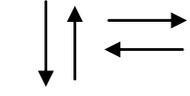
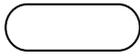
a. *Flowchart*

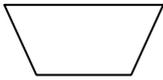
Flowchart adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir (*flowchart*) digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi[10]. Ada beberapa jenis-jenis *flowchart* diantaranya :

- A. Bagan alir sistem (*system flowchart*)
- B. Bagan alir dokumen (*document flowchart*)
- C. Bagan alir skematik (*schematic flowchart*)
- D. Bagan alir program (*program flowchart*)
- E. Bagan alir proses (*process flowchart*)

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

Table 2. 1 Simbol *Flowchart*

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

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

b. Use Case

Use Case 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 *scenario* 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 abstraksi dari interaksi antara sistem dan *actor*. *Use case* dibuat berdasarkan keperluan *actor*.

Berdasarkan definisi diatas maka dapat disimpulkan bahwa *Use Case* adalah kontruks untuk mendeskripsikan bagaimana sistem akan terlihat dimata pengguna potensial yang terdiri dari sekumpulan *scenario* dan *actor*. Sedangkan *use case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta analis dan klien[17]. Simbol *Use Case* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol *Use Case Diagram*

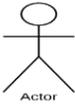
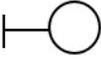
No.	Simbol	Nama Simbol dan Keterangan
1.		<i>Actor</i> , Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<i>Dependency</i> , Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri(independent).
3.		<i>Generalization</i> , Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
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>Association</i> , yang menghubungkan antara objek satu dengan objek lainnya
8.		<i>Use Case</i> , Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.

c. *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah obyek dan *message* yang diletakan antara obyek-obyek didalam *use case*.

Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*. *Sequence Diagram* menambahkan dimensi waktu pada interaksi diantara obyek[18]. Simbol-simbol yang dipakai dalam pembuatan *sequence diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol *Sequence Diagram*

No	Gambar	Nama Gambar	Keterangan
1.		<i>Actor</i>	Menggambar 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>message</i>
6.		<i>A Message</i>	Menggambarkan Pengiriman Pesan

d. Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas. Sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas[19]. Diagram kelas dibuat agar *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

Class diagram membantu dalam visualisasi struktur kelas – kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap – tiap kelas di dalam model desain (dalam *logical view*) dari suatu sistem. Selama proses analisis, *class diagram* memperlihatkan aturan – aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama proses analisis, *class diagram* memperlihatkan aturan – aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap *decían*, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat[20]. Berikut beberapa simbol dari *class diagram* dapat dilihat pada tabel 2.4 :

Tabel 2.4 Simbol *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.

2.2.3 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan proyek yang berisi data dan operasi yang diperlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis[21].

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus menjadi kelompok data dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut, sifat, komponen lainnya, dan dapat berinteraksi satu sama lain. Berikut komponennya :

a. Kelas (*class*)

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 (*object*)

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 (*method*)

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 (*attribute*)

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 (*inheritance*)

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

f. 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. 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. Polimorfisme

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

2.2.4 Basis Data

Basis data adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file lain sehingga membentuk satu bangun data untuk penginformasian suatu perusahaan instansi, dalam bahasan tertentu [22].

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 memiliki dua bentuk lisensi, yaitu *Free Software* dan *Shareware*. MySQL yang biasa kita gunakan adalah *MySQL Free Software I* yang berada dibawah Lisensi *General Public License (GPL)*. MySQL juga dapat didefinisikan sebagai sebuah database server, dapat juga berpersion 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*[23].

MySQL dikembangkan oleh sebuah perusahaan Swedia bernama MySQL AB, yang pada saat itu bernama TeX Data Konsult AB sekitar tahun 1994-1995. MySQL sudah ada sejak 1979. MySQL termasuk jenis *Relation Database Management System (RDBMS)* digunakan oleh banyak portal-portal internet sebagai basis data dari informasi yang ditampilkan pada situs *web*. Kepopuleran MySQL dimungkinkan karena kemudahannya untuk digunakan, cepat secara kinerja *query*, dan mencukupi untuk kebutuhan basis data perusahaan-

perusahaan skala menengah dan kecil. Istilah seperti tabel, baris, dan kolom tetap digunakan dalam MySQL. Sebuah basis data yang terdapat pada MySQL mengandung satu atau beberapa tabel yang terdiri dari sejumlah baris dan kolom[24].

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 memanajemen data yang terdapat dalam *database*. Secara garis besar, *SQL Server* mempunyai 3 (Tiga) jenis perintah *SQL* yaitu [25]:

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 [26]:

- a.) *CREATE*: perintah ini digunakan untuk membuat, termasuk di dalamnya membuat *database baru*, tabel baru *view* baru, dan kolom baru. Contoh: *CREATE DATABASE nama_database*.
- 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*.
- c.) *DROP*: perintah *DROP* berfungsi untuk menghapus *database* atau tabel. Contoh: *DROP DATABASE nama_database*.

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[27] :

- 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. Contoh: *INSERT INTO nama_tabel VALUES (data1, data2, dst...)*;

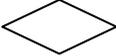
- b.) *SELECT*: perintah ini digunakan untuk mengambil dan menampilkan data dari tabel atau bahkan dari beberapa tabel dengan penggunaan relasi. Contoh: *SELECT* nama_kolom1, nama_kolom2 *FROM* nama_tabel;
- c.) *UPDATE*: perintah update digunakan untuk memperbaharui data pada sebuah tabel. Contoh: *UPDATE* nama_tabel *SET* kolom1=data1, kolom2=data2,... *WHERE* kolom=data;
- d.) *DELETE*: perintah *delete* digunakan untuk menghapus data dari sebuah tabel. Contoh: *DELETE FROM* nama_tabel *WHERE* kolom=data;

2.2.5 Entity Relationship Diagram (ERD)

ERD adalah suatu rancangan atau bentuk hubungan suatu kegiatan di dalam sistem yang berkaitan langsung dan mempunyai fungsi di dalam proses tersebut. ERD adalah suatu pemodelan dari basis data relasional yang didasarkan atas persepsi di dalam dunia nyata, dunia ini senantiasa terdiri dari sekumpulan objek yang saling berhubungan antara satu dengan yang lainnya. Suatu objek disebut *entity* dan hubpeungan yang dimilikinya disebut *relationship*. Suatu *entity* bersifat unik dan memiliki atribut sebagai pembeda dengan entity lainnya [17].

Berikut merupakan simbol-simbol dari ERD dapat dilihat pada tabel 2.5:

Tabel 2.5 Simbol ERD

No.	Nama	Simbol	Keterangan Fungsi
1.	Entitas		Persegi panjang menyatakan himpunan entitas adalah orang, kejadian, atau berada.
2.	Atribut		Atribut merupakan informasi yang diambil tentang sebuah entitas.
3.	Relasi		Belah ketupat menyatakan himpunan relasi merupakan hubungan antar entitas.
4.	Link		Garis penghubung antara himpunan, relasi, dan himpunan entitas dengan

			atributnya
--	--	--	------------

ERD memiliki derajat relasi atau biasa disebut kardinalitas. Kardinalitas menjelaskan batasan jumlah keterhubungan satu entity dengan entity lainnya. Berikut simbol kardinalitas pada tabel 2.6.

Tabel 2. 6 Macam-Macam *Kardinalitas*

No.	Simbol	Nama	Keterangan
1.		Relasi Satu ke Satu (<i>One to One</i>)	Relasi yang menunjukkan bahwa setiap himpunan entitas berhubungan dengan tepat satu himpunan entitas lainnya
2.		Relasi Satu ke Banyak (<i>One to Many</i>)	Relasi yang menunjukkan bahwa hubungan antara entitas pertama dengan entitas kedua adalah satu banding banyak, begitu pula sebaliknya
3.		Relasi Banyak ke Banyak (<i>Banyak to Many</i>)	Relasi yang menunjukkan bahwa setiap himpunan entitas boleh berhubungan dengan banyak himpunan entitas lainnya dan sebaliknya

2.2.6 Bursa Kerja

Bursa kerja adalah suatu lembaga yang mempunyai fungsi untuk merekrut tenaga kerja baru dan kemudian disalurkan atau ditempatkan ke dunia kerja atau dunia industri sesuai dengan perjanjian yang telah disepakati sebelumnya. Penyelenggaraan bursa kerja merupakan upaya untuk mendekatkan antara peluang kerja yang ada dengan sumber daya manusia yang tersedia. Hal ini sama halnya dengan yang dikemukakan oleh Kementerian Transmigrasi bahwa “Bursa Kerja adalah lembaga yang menjalankan fungsi penempatan untuk mempertemukan antara pencari kerja dengan pekerjaan baik dalam hubungan kerja maupun diluar hubungan. Satuan Pendidikan Tinggi dan di Lembaga kerja, pendaftaran pencari kerja, memberi penyuluhan dan bimbingan jabatan serta penyaluran dan penempatan pencari kerja[10].

2.2.7 PHP Mailer

PHP *Mailer* merupakan salah satu library yang disediakan dan bersifat free. Library ini dapat dijalankan di berbagai macam framework PHP, Library ini memungkinkan user dapat memanfaatkan *e-mail* sebagai media interaksi anatar user dengan sistem[11].

PHP *Mailer* adalah salah satu library PHP open source yang digunakan untuk mengirim *e-mail* dari localhost. PHP *Mailer* dapat menjalankan fungsinya sebagai *e-mail* jika kita mensupportnya dengan *Simple Mail Transfer Protocol* (SMTP). SMTP adalah suatu protocol yang diperlukan untuk mengirim dan menerima *e-mail*. Karena itulah kita harus menggunakan SMTP sebagai layanan mengirim *e-mail*. Layanan ini dapat digunakan untuk keperluan seperti memverifikasi *e-mail*, contohnya ketika mendaftarkan di Twitter atau Facebook. Setelah mendaftar maka kita diharuskan membuka *e-mail* dan memverifikasi[12].

PHP *Mailer* dapat digunakan sebagai notifikasi saat ada lowongan pekerjaan masuk untuk para pencari kerja, sehingga pencari kerja dapat mengetahui jika ada informasi lowongan pekerjaan lagi. PHP *Mailer* digunakan juga sebagai notifikasi jika pengguna sistem sudah aktivasi akun sehingga pengguna sistem dapat mengetahui jika ada balasan pesan yang masuk.