

BAB II

DASAR TEORI

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian mengenai Sistem Informasi Pemesanan Jasa Online berbasis *website* ini telah dilakukan oleh Wahyu Aji Tri Riswandhana dan Primaadi Airlangga pada tahun 2019. Penelitian yang dilakukan berbasis *website* di wilayah Kabupaten Jombang merupakan penelitian yang dirancang dapat membantu masyarakat bagi yang memiliki bakat atau keahlian sehingga dengan aplikasi ini mereka bisa mendaftarkan diri mereka melalui aplikasi ini supaya bakat dan keahlian mereka bisa bermanfaat untuk mendapat pekerjaan serta membantu orang lain yang membutuhkannya [3].

Penelitian kedua telah dilakukan oleh Indriyanita Br Sembiring dkk tahun 2021 dengan membangun sebuah Perancangan Aplikasi *Marketplace* Pemesanan Jasa Tukang Kunci berbasis *Android*. Pada Jasa Tukang Kunci ini mencakup proses pendaftaran para Jasa Tukang Kunci, pemesanan Jasa Tukang Kunci oleh pelanggan, dan konfirmasi pembayaran. Pembuatan Aplikasi *Android* menggunakan *eclipse*. Pemodelan sistem menggunakan *Unified Modeling Language (UML)*. Proses transaksi pembayaran untuk Jasa Tukang Kunci dengan pemesan ada dua yaitu dengan membayartunai atau mentransfer uang jasa ke rekening pemilik jasa dan pemesan mengisi formpembayaran untuk mengisi data bukti pembayaran yang sudah dilakukan dan kemudian penjual akan mengkonfirmasi pembayaran tersebut [4].

Penelitian ketiga yang dilakukan oleh Lalu Mutawal dkk tahun 2020 pada wilayah Lombok Tengah merupakan penelitian yang berupa pembuatan Sistem Informasi layanan jasa desain grafis, sistem informasi yang telah dibuat dapat berfungsi untuk Sistem Informasi layanan jasa desain dapat digunakan untuk melakukan pemesanan jasa desain seperti, logo, kemasan produk, *cover*, baner dan jasa desain lainnya. Penelitian ini menggunakan Kerangka Kerja *Scrum* karena praktisi dalam mengembangkan sistem. *Scrum* dapat mengetahui resiko dan kualitas projek perangkat lunak lebih cepat diprediksi [5].

Penelitian yang dilakukan oleh penulis memiliki beberapa perbedaan diantaranya, sistem ini dibuat untuk pemesanan tenaga kerja khususnya pemesanan tenaga kerja harian serta pengelolaan laporan bulanan kepada pemilik CV. Sistem yang dibuat menggunakan metode *prototype* karena untuk mengembangkan model atau rancangan produknya sampai menjadi produk *final* yang dapat memenuhi permintaan pengguna. Proses transaksi pembayaran untuk Jasa Tenaga Kerja Harian ini ada tiga jenis yaitu dengan membayar tunai kepada tenaga kerja kemudian tenaga kerja menyampaikan ke admin, *transfer*, dan mendatangi admin langsung ke kantor.

2.2 Landasan Teori

Dalam menunjang penelitian ini, maka diperlukannya teori-teori mendasar. Teori-teori yang digunakan dalam penelitian ini adalah:

2.2.1 Sistem

Sistem adalah kumpulan hal yang saling bekerja sama atau yang dihubungkan dengan cara tertentu sehingga membentuk satu kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan, sehingga dapat disimpulkan bahwa sistem adalah suatu kumpulan elemen yang saling berhubungan dan berfungsi untuk menyelesaikan suatu sasaran atau untuk mencapai suatu tujuan[6].

Ada 3 elemen yang membentuk sebuah sistem yaitu [7] :

a. *Input*

Segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk di proses.

b. Proses

Bagian yang melakukan perubahan dari *input* menjadi *output* yang berguna, misalnya berupa informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna, misalnya sisa pembuangan atau limbah.

c. *Output*

Hasil dari pemrosesan, misalnya berupa suatu informasi, saran, cetakan laporan, dll. Berdasarkan pengertian diatas dapat disimpulkan sistem adalah cara yang kita lakukan untuk mencapai tujuan yang telah kita buat mulai dari menginput sesuatu memprosesnya kemudian menghasilkan *output*.

2.2.2 Informasi

Informasi adalah hasil pemrosesan data yang diperoleh dari setiap elemen sistem menjadi bentuk yang mudah dipahami, dan merupakan suatu pengetahuan yang relevan yang dibutuhkan oleh orang untuk menambah pemahamannya terhadap fakta-fakta yang ada[8]. Informasi harus mempunyai nilai bila informasi tersebut dapat mengakibatkan suatu perubahan dalam Tindakan yang diambil. Perubahan ini penting untuk disadari karena sesungguhnya data tidak mempunyai nilai apa-apa untuk pengambilan keputusan, hanya informasi mempunyai nilai, dalam arti bahwa informasi akan memudahkan manajer untuk pengambilan keputusan[9].

Menurut Atin Hafidiah dan Dusan Sumartayaalam (2003:51-52), nilai informasi didasarkan pada sepuluh sifat sebagai berikut:

1. Mudahnya dapat diperoleh (*accessibility*). Sifat ini menunjukkan pada mudahnya dan cepatnya output informasi diperoleh, misalnya satu menit dibandingkan dua puluh empat jam.
2. Sifat dan luasnya (*comprehensive*). Sifat ini menunjukkan lengkapnya isi informasi ini tidak berarti hanya mengenai volumenya, akan tetapi juga mengenai outputnya.
3. Ketelitian (*accuracy*). Sifat ini berhubungan dengan tingkat kebebasan dari kesalahan.
4. Kecocokan (*appropriateness*). Sifat menunjuk betapa baiknya output informasi dalam hubungannya dengan permintaan pemakai. Isi informasi harus ada hubungannya dengan masalah yang sedang dihadapi.
5. Ketepatan waktu (*timeless*). Sifat ini berhubungan dengan waktu proses/siklus menghasilkan informasi yang lebih pendek.
6. Kejelasan (*clarify*). Menunjuk pada tingkat kebebasan dari istilah-istilah yang tidak jelas.
7. Keluwesan (*fleksibiliy*). Dapat dipakai tidak hanya pada satu keputusan saja, tetapi lebih dari satu.
8. Dapat dibuktikan/dicocokkan (*variability*). Dapat diuji oleh pemakai sehingga sampai pada kesimpulan yang sama.
9. Tidak mengandung prasangka, tidak ada keinginan untuk menghasilkan atau mengubah informasi guna mendapatkan kesimpulan yang telah dipertimbangkan atau ditentukan sebelumnya.
10. Dapat diukur, informasi tersebut dihasilkan dari sistem informasi formal dan legal.

2.2.3 Sistem Informasi

Sistem Informasi merupakan elemen-elemen yang saling berinteraksi secara sistematis dan teratur untuk menciptakan dan membentuk aliran informasi yang akan mendukung pembuatan keputusan dan melakukan *control* terhadap jalannya perusahaan[10].

Fungsi Sistem Informasi antara lain :

1. Mengembangkan proses perencanaan yang efektif.
2. Meningkatkan aksesibilitas data yang ada secara efektif dan efisien kepada pengguna, tanpa dengan perantara sistem informasi.
3. Mengantisipasi dan memahami akan konsekuensi ekonomi.

Ciri – Ciri Sistem Informasi yaitu :

1. Baru, yang dimaksud dengan baru adalah informasi yang didapat sama sekali baru dan segar bagi penerima.
2. Tambahan, adalah memberikan tambahan terhadap informasi yang sebelumnya telah ada.
3. Kolektif, adalah informasi yang dapat menjadi suatu koreksi dari informasi yang salah sebelumnya.
4. Penegas, adalah informasi yang dapat mempertegas informasi yang telah ada.

2.2.4 Tenaga Kerja

Tenaga kerja merupakan penduduk yang berumur didalam batas usia kerja. Angkatan kerja adalah penduduk dalam usia kerja yang terlibat atau berusaha untuk terlibat dalam kegiatan produktif yaitu memproduksi barang dan jasa[11]. Tenaga kerja atau pekerja adalah setiap orang bekerja dengan menerima upah dan imbalan dalam bentuk lain[12].

Secara umum tenaga kerja di kalsifikasi menjadi 4 golongan yaitu :

1. Tenaga kerja terdidik
2. Tenaga kerja terlatih
3. Tenaga kerja terdidik dan terlatih
4. Tenaga kerja tidak terdidik dan tidak terlatih.

2.2.5 Rekayasa perangkat lunak (RPL)

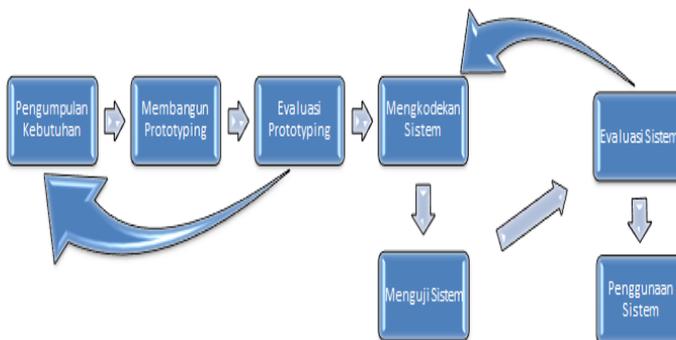
Rekayasa perangkat lunak (*software engineering*) merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi maupun manfaat yang dipercaya dan bekerja secara efisien menggunakan mesin komputer. Rekayasa perangkat lunak lebih fokus pada praktik pengembangan perangkat lunak dan mengirimkan perangkat lunak yang bermanfaat kepada pelanggan (*customer*) dan dapat digunakan oleh pengguna (*user*) dengan baik.

Rekayasa perangkat lunak lebih fokus pada bagaimana membuat perangkat lunak yang memenuhi kriteria [13]:

1. Dapat terus dipelihara setelah perangkat lunak selesai dibuat seiring berkembangnya teknologi dan lingkungan (*maintainability*).
2. Dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi (*dependability* dan *robust*).
3. Efisien dari segi sumber daya dan penggunaan
4. Kemampuan untuk dipakai sesuai dengan kebutuhan (*usability*).

1. *Prototype*

Prototype adalah sebuah skema rancangan sistem yang membentuk model dan standar ukuran atau skalabilitas yang akan dikerjakan nantinya. Setiap pengembang maupun pengguna dapat berinteraksi langsung dengan model tersebut tanpa harus membuat produk nyatanya[14].



Gambar 2. 1 Metode *Prototype* (Pressman, 2012)

Tahapan *prototyping* adalah sebagai berikut :

1. Pengumpulan Kebutuhan
Pada tahapan ini pelanggan dan pengembang bersama-sama mengidentifikasi semua kebutuhan dan garis besar sistem yang akan dibuat.
2. Membangun *Prototyping*
Tahap membangun *prototyping* dengan cara membuat perancangan secara sementara.
3. Evaluasi *Prototyping*
Evaluasi *prototyping* ini dilakukan oleh pelanggan apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai, maka dilanjutkan ke langkah 4. Jika tidak, *prototyping* direvisi dengan mengulang pada langkah ke 1, 2, dan 3.
4. Mengkodekan Sistem
Dalam tahapan ini, *prototyping* yang sudah disepakati customer akan diterjemahkan kedalam bahasa pemrograman yang sesuai. Jika ada pengembangan sistem, maka masuk pada tahap ini.
5. Menguji Sistem
Setelah sistem sudah menjadi suatu perangkat lunak yang siap dipakai, harus di tes terlebih dahulu sebelum digunakan.
6. Evaluasi Sistem
Pelanggan yang mengevaluasi apakah sistem yang sudah dibuat dan sudah jadi ini sudah sesuai dengan yang diharapkan atau belum. Jika iya, langkah selanjutnya ke 7 dapat dilakukan. Jika belum, ulangi langkah ke 4 dan 5.
7. Penggunaan Sistem
Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

2. Black box Testing

Sistem Informasi Tenaga Kerja Harian ini menggunakan metode pengujian *Blackbox Testing*. *Blackbox Testing* adalah sebuah metode yang dipakai untuk menguji sebuah *software* tanpa harus memperhatikan *detail software*[15]. *Blackbox Testing* adalah Pengujian *black-box* menguji dari

sisi fungsionalitas, aplikasi diuji agar dapat memenuhi fungsi sebagaimana mestinya[16].

Keuntungan penggunaan metode *Blackbox Tetsting* adalah [16]:

1. Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu.
2. Pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan.
3. Programmer dan tester keduanya saling bergantung satu sama lain.

Dengan menggunakan pengujian metode *Blackbox Testing* ini, diharapkan nantinya berfokus pada sisi fungsionalitas, khususnya pada *input* dan *output* aplikasi (apakah sudah sesuai dengan apa yang diharapkan atau belum).

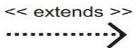
3. UML (*Unified Modeling Language*)

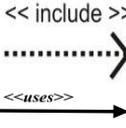
(UML) *Unified Modeling Language* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung[17]. UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, sertamenggambarakan arsitektur dalam pemrograman berorientasi objek[18].

a. *Use Case Diagram*

Use Case Diagram Merupakan diagram yang bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan suatu sistem tersendiri melalui sebuah cerita bagaimana sebuah sistem dipakai [14].

Tabel 2. 1 Simbol-Simbol Use Case Diagram

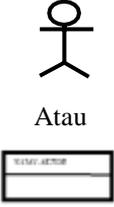
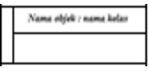
No.	Simbol	Nama Simbol	Keterangan
1.		<i>Use case</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<i>Actor</i>	Simbol yang menunjukkan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem itu sendiri.
3.		<i>Association</i>	Simbol yang digunakan untuk menunjukkan adanya komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.		<i>Extend</i>	Simbol yang digunakan untuk menunjukkan adanya relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan tersebut.
5.		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

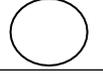
6.	 <p>The diagram shows two types of relationships between use cases. The first is an include relationship, represented by a dashed arrow pointing to a right-facing curly bracket, with the text '<< include >>' above it. The second is a uses relationship, represented by a solid arrow pointing to the right, with the text '<< uses >>' above it.</p>	Menggunakan /include / uses	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> ini untuk menjalankan fungsi atau sebagai syarat dijalankannya <i>use case</i> ini.
----	--	-----------------------------	--

b. *Sequence Diagram*

Sequence diagram atau diagram urutan adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci [14].

Tabel 2. 2 Simbol-Simbol *Sequence Diagram*

No	Simbol	Nama Simbol	Kegunaan
1.	 <p>The diagram shows two symbols for an actor: a stick figure and a rectangular box with a horizontal line near the top. The text 'Atau' is placed between the two symbols.</p>	<i>Actor</i>	Digunakan untuk menggambarkan <i>user</i> atau pengguna. <i>User</i> atau pengguna bisa berupa orang, proses, atau sistem lain yang saling berinteraksi dengan informasi yang akan dibuat.
2.	 <p>The diagram shows a vertical dashed line representing a life line.</p>	<i>LifeLine</i>	Menyatakan kehidupan suatu objek.
3.	 <p>The diagram shows a rectangular box with a horizontal line near the top. The text 'Nama objek / nama kelas' is written inside the box.</p>	Objek	Digunakan untuk menggambarkan objek yang berinteraksi pesan.

4.		Waktu Aktif	Digunakan untuk menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan simbol ini adalah sebuah tahapan yang dilakukan di dalamnya.
5.		<i>Entity Class</i>	Digunakan penghubung kegiatan yang akan dilakukan.
6.		<i>Control Class</i>	Digunakan sebagai penghubung <i>boundary</i> dengan tabel.
7.		<i>Boundary</i>	Digunakan untuk menggambarkan sebuah <i>form</i> .
8.		<i>Control Class</i>	Digunakan untuk menghubungkan <i>Boundary</i> dengan tabel
9.		<i>Entity Class</i>	Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.

4. *Flowchart*

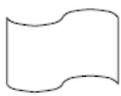
Flowchart adalah cara untuk menjelaskan tahap-tahap pemecahan masalah dengan merepresentasikan simbol-simbol tertentu yang mudah dipahami, mudah digunakan dan standar [19].

Tujuan digunakannya *flowchart* antara lain :

1. Untuk mengembangkan pemahaman tentang bagaimana proses dilakukan.
2. Untuk mempelajari perbaikan proses.
3. Untuk berkomunikasi dengan orang lain bagaimana proses dilakukan.
4. Untuk keperluan komunikasi yang lebih baik di antara orang-orang yang terlibat dalam proses yang sama.
5. Untuk mendokumentasikan proses.
6. Untuk merencanakan sebuah kegiatan.

Tabel 2. 3 Simbol Flowchart Sistem

No	Simbol	Nama Simbol	Kegunaan
1.		<i>Input/Output</i>	Mempresentasikan <i>input</i> data atau <i>output</i> data yang diproses atau informasi.
2.		Proses	Mempresentasikan operasi.
3.		Penghubung	Keluar ke atau masuk dari bagian lain <i>flowchart</i> khususnya halaman yang sama.
4.		Anak Panah	Merepresentasikan alur kerja.
5.		Penjelasan	Digunakan untuk komentar tambahan.
6.		Keputusan	Keputusan dalam program.
7.		<i>Predefined Process</i>	Rincian operasi berada di tempat lain.
8.		<i>Preparation</i>	Pemberian harga awal.

9.		<i>Terminal Points</i>	Awal/akhir <i>flowchart</i> .
10.		<i>Punched card</i>	Input/output yang menggunakan kartu berlubang.
11.		Dokumen	I/O dalam format yang dicetak.
12.		<i>Magnetic Tape</i>	I/O yang menggunakan pita magnetik.
13.		<i>Magnetic Disk</i>	I/O yang menggunakan <i>disk</i> magnetik.
14.		<i>Magnetic Drum</i>	I/O yang menggunakan drum magnetik.
15.		<i>On-line Storage</i>	I/O yang menggunakan penyimpanan akses langsung.
16.		<i>Punched Tape</i>	I/O yang menggunakan pita kertas berlubang.
17.		Manual Input	Input yang dimasukkan secara manual dari <i>keyboard</i> .
18.		<i>Display</i>	Output yang ditampilkan pada terminal.
19.		<i>Manual Operation</i>	Operasi Manual.
20.		<i>Off-line Storage</i>	Penyimpanan yang tidak dapat diakses oleh komputer secara langsung.

2.2.6 Website

Website merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut dengan *web page* dan *link* dalam *website* memungkinkan pengguna bisa berpindah dari satu *page* ke *page* lain, *Website* adalah sistem *hypermedia* yang ber area luas yang ditujukan untuk akses secara *universal*[20].

Website merupakan kumpulan informasi yang terdiri dari halaman *web* yang saling terhubung satu sama lain yang disediakan secara perorangan, kelompok, atau pun organisasi. Situs yang baik menampilkan visual yang menarik dan berfungsi sesuai dengan kebutuhan pengguna[21]. Pada penelitian ini menggunakan jenis *website* dinamis, karena sebuah *website* menyediakan *content* atau isi yang selalu berubah-ubah setiap saat. Bahasa pemrograman yang digunakan antara lain PHP dan memanfaatkan *database* MySQL.

1. CSS (*Cascading Style Sheet*)

CSS merupakan sebuah temuan untuk membantu pengembang web yang tadinya hanya bergantung pada HTML sebagai bahasa *markup*[22]. CSS adalah bahasa pemrograman yang di gunakan untuk *men-design* sebuah halaman *website*. dalam merancang halaman *website*, CSS menggunakan penanda yang kita kenal dengan *id dan class*[23].

Tujuan utama CSS diciptakan untuk membedakan konten dari dokumen dan dari tampilan dokumen, dengan itu, pembuatan ataupun pemrograman ulang *website* akan lebih mudah dilakukan. Dengan adanya CSS, konten dan *desain website* akan mudah dibedakan, jadi memungkinkan untuk melakukan pengulangan pada tampilan-tampilan tertentu dalam suatu *website*, sehingga akan memudahkan dalam membuat halaman *website* yang banyak, yang pada akhirnya dapat memangkas waktu pembuatan *website* [24].

2. HTML (*Hypertext Markup Language*)

HTML adalah suatu bahasa yang digunakan untuk membuat sebuah halaman *website*. HTML itu bahasa yang fleksibel karena tidak tergantung pada suatu *platform* (sistem operasi) tertentu[25]. HTML merupakan bahasa pemrograman *website* yang memberitahukan peramban *website*

(*web browser*) bagaimana menyusun dan menyajikan konten di halaman web[26].

Menurut Hidayatullah dan Kawistara (2015) “*Hypertext Merkup Languange (HTML)* adalah bahasa standar yang digunakan untuk menampilkan halaman *website*”. Yang bisa dilakukan dengan HTML yaitu [27] :

- a. Mengatur tampilan dari halaman *website* dan isinya.
- b. Membuat tabel dalam halaman *website*.
- c. Mempublikasikan halaman web secara online.
- d. Membuat *form* yang bisa digunakan untuk menangani *registrasi* dan transaksi *via website*.

3. **PHP (*Personal Home Page*)**

PHP merupakan singkatan dari *PHP Hypertext Preprocessor*. PHP merupakan bahasa pemrograman *script* yang diletakkan dalam *server* yang biasa digunakan untuk membuat aplikasi *web* yang bersifat dinamis[28]. PHP adalah proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. PHP merupakan suatu bahasa dengan hak cipta terbuka atau yang juga dikenal dengan istilah *Open Source*, yaitu pengguna dapat mengembangkan kode fungsi PHP dengan kebutuhannya[29].

Keuntungan dari PHP, yaitu :

1. *Open source*, ini akan membantu dalam menciptakan sebuah komunitas dukungan, perpustakaan ekstensi berlimpah.
2. Kecepatan, relatif cepat karena menggunakan banyak sumber daya sistem.
3. Dapat dijalankan di banyak *platform*.
4. Dukungan *library* yang kuat.

4. **Framework Codeigniter**

Codeigniter merupakan sebuah aplikasi terbuka yang berfondasi dari *framework* PHP yang digunakan untuk membangun suatu *website* dinamik dengan menggunakan kode PHP. CodeIgniter membantu memudahkan *developer* untuk bisa membuat suatu aplikasi berbasis *website* dengan sangat dengan cepat dan memberikan kemudahan[30].

Framework CodeIgniter adalah sebuah *framework* PHP yang dapat membantu mempercepat developer dalam pengembangan aplikasi *web* berbasis PHP[31].

Keuntungan dari menggunakan *framework CodeIgniter* adalah [32] :

1. *CodeIgniter* adalah *framework* yang bersifat *free* dan *opensource*.
2. *CodeIgniter* memiliki ukuran yang kecil dibandingkan dengan *framework* lain. Setelah proses *instalasi*, *framework CodeIgniter* hanya berukuran kurang lebih 2 mb. Dokumentasi *CodeIgniter* memiliki ukuran sekitar 6 mb.
3. Aplikasi yang dibuat menggunakan *CodeIgniter* bisa berjalan lebih cepat.
4. *CodeIgniter* dapat diperluas sesuai dengan kebutuhan.

2.2.7 Basis Data

Basis Data adalah sistem berkas terpadu yang dirancang terutama untuk meminimalkan pengulangan data[33].

Dalam Sistem Basis data memiliki beberapa komponen yaitu [34]:

1. Perangkat Keras (*Hardware*)
Perangkat keras yang biasanya terdapat dalam sistem basis data adalah memori sekunder hardisk.
2. Sistem Operasi (*Operating System*)
Sistem Operasi (*Operating System*) merupakan program yang mengaktifkan atau mengfungsikan sistem 2 komputer, mengendalikan seluruh sumber daya (*resource*) dan melakukan operasi-operasi dalam komputer. Sistem Operasi yang banyak digunakan seperti: MS-DOS, MS-Windows 95 MS Windows NT, dan Unix.
3. Basis data (Database) dapat memiliki beberapa basis data.
Setiap basis data dapat berisi atau memiliki sejumlah objek basis data seperti *file* atau tabel. Database Management System (DBMS) Pengolahan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak yang disebut DBMS yang menentukan bagaimana data disimpan, diubah dan diambil kembali.
4. Pemakai (*User*)
Bagi pemakai dapat berinteraksi dengan basis data dan memanipulasi data dalam program yang ditulis dalam bahasa pemrograman.

1. MySQL (*My Structured Query Language*)

MySQL merupakan *software database opensource* yang paling populer di dunia. MySQL menjadi pilihan utama bagi banyak pengembang *software* dan aplikasi hal ini dikarenakan kelebihan MySQL diantaranya sintaks nya yang mudah dipahami, didukung program-program umum seperti *C, C++, Java, PHP, Python*. Pengguna MySQL tidak hanya sebatas pengguna perseorangan maupun perusahaan kecil, namun perusahaan seperti *Yahoo!, Google, Nokia, Youtube, Wordpress* juga menggunakan DBMS MySQL[35].

MySQL merupakan *database* yang bersifat *client server*, dimana data diletakan di *server* yang bisa diakses melalui *client*. Pengaksesan dapat dilakukan apabila komputer telah terhubung dengan *server*[36].

Adapun keuntungan dari MySQL adalah [37]:

1. MySQL dapat digunakan bebas oleh siapapun tanpa membelinya atau harus membayar lisensi (*open source*).
2. Merupakan database server yang dapat dihubungkan dengan jaringan internet dan diakses dari jauh.
3. Memiliki kapasitas yang cukup besar bahkan jangkauan *Gygabite* sekalipun.
4. Memiliki sistem *software* yang ringan dan tidak membebani kinerja server dari komputer karena juga bekerja pada *background*.
5. MySQL bisa diakses aplikasi apa saja yang berupa gambaran contohnya *Visual Basic* maupun *Delphi*

2. Class Diagram

Class diagram adalah diagram yang dibuat pada tahap desain suatu perangkat lunak. Pengukuran kualitas dari desain class diagram dari software yang akan dibangun dapat mengurangi revisi-revisi yang mungkin terjadi di kemudian hari (Khanahmadliravi and Khataee, 2012).

Tabel 2. 4 Simbol-simbol *class diagram*

No	Simbol	Nama Simbol	Kegunaan
1.		Kelas	Kelas pada struktur system

2.		Antarmuka <i>/Interface</i>	Sama dengan konsep interface dalam pemrograman berorientasi objek
3.		Asosiasi/ <i>association</i>	Keluar ke atau masuk dari bagian lain <i>flowchart</i> khususnya halaman yang sama.
4.		Asosiasi berarah / <i>Directed association</i>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i> .
5.		Generalisasi	Relasi antarkelas dengan makna generalisasipesialisasi (umum khusus).
6.		Kebergantungan/ <i>depedency</i>	Relasi antar kelas dengan makna ketergantungan antarkelas.
7.		Agresasi/ <i>aggregation</i>	Relasi antarkelas dengan makna semua bagian.

2.2.8 Pemrograman Berorientasi Objek

Pemrograman berorientasi objek adalah suatu cara baru dalam berpikir serta berlogika untuk menghadapi masalah-masalah yang akan dicoba atasi dengan bantuan komputer[38]. Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya [38].

~Halaman Ini Sengaja Dikosongkan~