

BAB II

TINJAUAN PUSTAKA

2.1. Pengembangan Perangkat Lunak

Pengembangan perangkat lunak merupakan pembangunan dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin. Atau dengan kata lain pengembang proses yang sistematis untuk membangun perangkat lunak yang berkualitas. Pondasi dalam pengembangan perangkat lunak yaitu process layer atau lapisan proses. Proses rekayasa perangkat lunak berfungsi untuk menghubungkan lapisan-lapisan teknologi dan memungkinkan pengembangan perangkat lunak yang rasional dan tepat waktu. Proses mendefinisikan suatu perangkat kerja yang harus dibangun sedemikian rupa sehingga dapat terjadi penghantaran teknologi rekayasa perangkat lunak yang efektif. Proses-proses perangkat lunak membentuk dasar bagi kendali manajemen proyek perangkat lunak dan membangun konteks dimana metode- metode teknis dapat diterapkan, dimana produkproduk kerja (model, dokumen, data, laporan, formulir, dsb) dibuat, dimana tonggak waktu dibuat, dimana kualitas dipastikan, dan dimana perubahan dapat diatur dengan tepat.

2.2. Konsep Dasar Aplikasi

Pada bagian ini akan dijelaskan definisi dasar dari istilah yang terkait dengan pembuatan aplikasi inventaris barang berbasis website *framework codeigniter*.

2.2.1 Pengertian Rancangan

Rancangan merupakan salah satu hal yang penting dalam membuat program. Adapun tujuan dari rancangan ialah untuk memberi gambaran yang

jelas lengkap kepada pemrogram dan ahli teknik yang terlibat. Rancangan harus berguna dan mudah dipahami sehingga mudah digunakan. Rancangan

adalah sebuah proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta di dalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaannya.

Menurut Pressman yang dikutip oleh Buchari dkk dalam jurnal E-Journal Teknik Informatika Vol. 6 No. 1 (2015), rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan.

2.2.2. Pengertian Bangun

Menurut Pressman yang dikutip oleh Taufan dalam jurnal E-Journal Teknik Informatika Vol. 11 No. 1 (2017), bangun atau pembangunan adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada secara keseluruhan. Jadi dapat disimpulkan bahwa Rancang Bangun adalah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari 7 8 beberapa elemen yang terpisah kedalam suatu kesatuan yang utuh dan berfungsi. Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut atau memperbaiki sistem yang sudah ada.

2.2.3. Pengertian Aplikasi

Aplikasi menurut Dhanta dikutip dari Sanjaya (2015) adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word, Microsoft Excel*. Aplikasi berasal dari kata *application* yang artinya penerapan lamaran penggunaan. Menurut Jogiyanto dikutip oleh Ramzi (2016) aplikasi merupakan penerapan, menyimpan sesuatu hal, data, permasalahan, pekerjaan kedalam suatu sarana atau media yang dapat digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan yang ada sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari hal data, permasalahan, dan pekerjaan itu sendiri. Jadi aplikasi merupakan sebuah transformasi dari sebuah permasalahan atau pekerjaan berupa hal yang sulit difahami menjadi lebih sederhana, mudah dan dapat dimengerti oleh pengguna. Sehingga dengan adanya aplikasi, sebuah permasalahan akan terbantu lebih cepat dan tepat.

2.2.4. Pengertian Database

Menurut Stephens dan Plew didalam buku Simamarta dan Paryudi (2016:1), *Database* (basis data) adalah kumpulan data yang di simpan secara sistematis didalam komputer yang dapat diolah atau di manipulasi menggunakan perangkat lunak (*software*) program atau aplikasi untuk menghasilkan informasi.

2.2.5. Pengertian Website

“*Website* (bahasa Indonesia: situs *web*) merupakan kumpulan halaman yang menampilkan informasi data, teks, data gambar, data animasi, suara, video, atau gabungan semuanya yang dapat diakses melalui *internet*” (Gusdeka, 2019:15). Terdapat dua jenis aplikasi *web* yaitu yang bersifat statis dan dinamis dan fungsi salah satunya untuk mengelola data pada suatu perusahaan.

Menurut Triyono dalam Jurnal SENSI Vol.4 No.1 (2018:23), *website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau bergerak, animasi, suara, dan atau gabungan dari semuanya baik yang bersifat dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman.

Berdasarkan pengertian diatas, peneliti menyimpulkan bahwa *website* merupakan situs berupa halaman – halaman yang berisi informasi text, gambar, video, audio, animasi yang bersifat dinamis dan saling terkait.

2.2.6 Pengertian Framework Codeigniter

Framework Codeigniter merupakan salah satu framework *PHP* yang populer dan paling banyak digunakan, di Indonesia pun pengembang aplikasi berbasis *web* banyak menggunakan *framework PHP* ini. *Framework Codeigniter* disebut juga sebagai *Application Development Framework* yaitu sebuah *toolkit* untuk para pengembang aplikasi berbasis *web* yang menggunakan bahasa

pemrograman *PHP*. Menurut Rudianto (2015:43) *PHP* adalah Bahasa *server-side-scripting* yang menyatu dengan *HTML* untuk membuat halaman *website* yang dinamis.” Karena *PHP* merupakan *server-side-scripting* maka *sintaks* dan perintah-perintah *PHP* akan dieksekusi di server kemudian hasilnya akan dikirimkan ke *browser* dengan format *HTML*. Tujuannya yaitu untuk mempercepat dalam penulisan program karena *Codeigniter* menyediakan banyak *libraries* yang umum dan sering digunakan dengan cara penggunaan yang sangat mudah dan simpel, sehingga *programmer* dapat fokus pada pembuatan program dengan meminimalkan penulisan koding yang dibutuhkan.

2.2.6. Pengertian Inventaris Barang

Menurut Wibowo (2015), inventaris barang adalah kegiatan yang terdiri dari data masuknya barang, data retur serta data persediaan barang yang melaporkan seluruh transaksi keluar masuk barang dari perhari sampai perbulan.

Inventaris barang adalah daftar lengkap barang-barang yang ada di sebuah perusahaan atau instansi baik yang keluar maupun yang masuk, baik yang berwujud ataupun tidak. Daftar barang ini ditulis dalam satu buku khusus dan harus lengkap tanpa ada yang terlewat. Tujuannya untuk menganalisis alat penunjang aktivitas perusahaan, kekayaan dan modal.

2.3. Tools Pemodelan yang Digunakan

Pada proyek akhir ini *tools* yang digunakan adalah *UML*, *Use Case Diagram*, dan *Entity-Relationship Diagram*.

2.3.1. UML

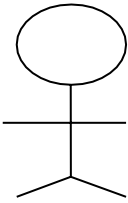
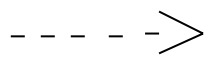
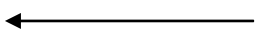
Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. *UML* merupakan metodologi dalam mengembangkan system berorientasi objek dan juga merupakan alat untuk mendukung pengembangan system. (Fatimah:2015). *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. *UML* hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada metodologi berorientasi objek.

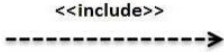
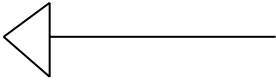


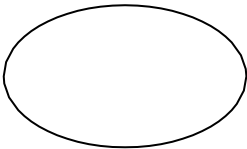

2.3.2. Use Case Diagram

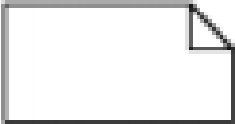
Menurut Mamed:2015 "*Use Case Diagram* adalah sesuatu atau proses merepresentasikan hal-hal yang dapat dilakukan oleh aktor dalam menyelesaikan sebuah pekerjaan." Menurut Shalahuddin dalam jurnal (Umar Al Faruq, 2015) mengungkapkan, bahwa "*Diagram use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut." (Yunahar: 2018).

Berikut simbol-simbol yang ada pada diagram *use case* :

Tabel 2.1 Simbol *Use Case* Diagram

| No. | Nama | Simbol | Keterangan |
|-----|-----------------------|--|--|
| 1. | <i>Actor</i> |  | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> . |
| 2. | <i>Depedency</i> |  | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>). |
| 3. | <i>Generalization</i> |  | Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>) |

| | | | |
|----|----------------------|--|---|
| 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> |  | Apa yang menghubungkan antara objek satu |
| 7. | Sistem |  | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8. | <i>Use Case</i> |  | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 9. | <i>Collaboration</i> |  | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk |

| | | | |
|-----|-------------|---|---|
| | | | menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10. | <i>Note</i> |  | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

2.3.3 Class Diagram

Menurut (Hendini, 2016) *Class Diagram* merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class Diagram* secara khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalization* dan *Aggregation*, attribute (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*.

Tabel 2.2 Multiplicity Class Diagram

| <i>Multiplicity</i> | Penjelasan |
|---------------------|------------|
|---------------------|------------|

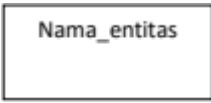

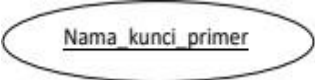
| | |
|---|---------------------|
| 1 | Satu dan hanya satu |
|---|---------------------|


| | |
|------|---|
| 0..* | Boleh tidak ada atau 1 atau lebih |
| 1..* | 1 atau lebih |
| 0..1 | Boleh tidak ada, maksimal 1 |
| n..n | Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4 |



2.3.4. ERD

Menurut (Minda Septiani, Nurul Afni, 2019) Pemodelan awal basis data yang paling banyak digunakan adalah ERD yang dikembangkan berdasarkan teori himpunan dalam bidang matematika.

Tabel 2.3 Simbol ERD

| Nama Simbol | Simbol | Deskripsi |
|----------------------|---|---|
| Entitas |  | Entitas merupakan data inti yang akan disimpan; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel. |
| Atribut |  | Kolom data yang butuh disimpan dalam suatu Entitas |
| Atribut Kunci Primer |  | Kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan. |

| | | |
|--------------------------------------|---|---|
| Atribut multinilai/multivalu e |  | Kolom data yang butuh disimpan dalam suatu entitas yang dapat |
|--------------------------------------|---|---|

| | | |
|----------------------|---|--|
| | | memiliki nilai lebih dari satu. |
| Relasi |  | Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja. |
| Asosiasi/Association |  | Penghubung antara relasi dan entitas di mana dikedua ujungnya memiliki <i>multiplicity</i> . |

2.4. Tools Pembangunan Aplikasi

Berikut merupakan definisi dan penjelasan tools yang dibutuhkan dalam pembuatan dan pengembangan aplikasi.

2.4.1. PHP

Menurut (Sitohang, 2019) PHP adalah PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman web berbasis *server (server side)* yang mampu memarsing kode PHP dari kode dengan ekstensi PHP sehingga menghasilkan tampilan *website* yang dinamis disisi client.

2.4.1.1 Codeigniter

Menurut (Subagia, 2018) *CodeIgniter* merupakan sebuah web *framework* yang dikembangkan oleh Rick Ellis dari Ellis Lab. *CodeIgniter* dirancang untuk menjadi sebuah web framework yang ringan dan mudah digunakan. Bahkan dari Rasmus Lerdorf, sang pencipta bahasa pemrograman PHP mengatakan bahwa *CodeIgniter* merupakan web *framework* yang mudah, cepat, dan handal. Dirilis pada dokumentasi *CodeIgnite, CodeIgniter*

merupakan *toolkit* bagi orang yang ingin membangun aplikasi web menggunakan PHP.

Berikut adalah konsep *Model-View-Controller* yang diterapkan pada *CodeIgniter*:

1. *Model*, menggambarkan struktur data. Biasanya kelas model akan berisi fungsi yang digunakan untuk mengambil, menambah dan memperbaharui informasi yang ada di *database*.
2. *View*, informasi yang diperlihatkan kepada user. View adalah halaman web yang terdiri dari HTML, CSS dan *Javascript*, akan tetapi pada *CodeIgniter*, *view* dapat juga sebagai potongan halaman seperti *header* atau *footer*. Bahkan dapat juga halaman RSS atau tipe halaman lainnya.
3. *Controller*, perantara *Model*, *View*, dan *resource* lainnya yang dibutuhkan untuk menangani HTTP *request* dan menghasilkan halaman web.

2.4.1.2 HTML

Menurut (Enterprise, 2018) HTML, digunakan untuk membuat struktur halaman *website*. Bisa dibilang secara umum bahwa HTML digunakan untuk mendesain *website*, meskipun dalam praktiknya HTML tidak berdiri sendiri sebab akan digabungkan dengan CSS atau *Script* lain, seperti *Javascript*. Pada kesempatan ini, kita akan menganggap HTML sebagai program untuk membuat desain sebuah *website*, dimana kita bisa menulis teks, memasukkan gambar, membuat *form*, dan sebagainya.

2.4.2. MySQL

Menurut (Enterprise, 2018) MySQL merupakan *server* yang melayani *database*. Untuk membuat dan mengelola *database*, kita dapat mempelajari pemrograman khusus yang disebut *query* (perintah) SQL. *Database* sendiri dibutuhkan jika kita ingin menginput data dari *user* menggunakan *form* HTML untuk kemudian diolah PHP agar bisa disimpan kedalam *database* MySQL.

2.4.3. XAMPP

Menurut (Enterprise, 2018) XAMPP merupakan *server* yang paling banyak digunakan untuk keperluan belajar PHP secara mandiri, terutama bagi *programmer* pemula. Selain gratis, fiturnya tergolong lengkap dan gampang digunakan oleh *programmer* PHP tingkat awal, yang perlu dilakukan hanyalah menjalankan *module Apache* yang ada di dalam XAMPP tersebut.

2.4.4. Visual Studio Code

Menurut (Lardinois, 2015) Visual Studio Code adalah editor source code yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS. Ini termasuk dukungan untuk debugging, GIT Control yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode refactoring. Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, shortcut keyboard, dan preferensi. Visual Studio Code gratis dan open-source, meskipun unduhan resmi berada di bawah lisensi proprietary.

BAB II

TINJAUAN PUSTAKA

2.5. Pengembangan Perangkat Lunak

Pengembangan perangkat lunak merupakan pembangunan dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin. Atau dengan kata lain pengembang proses yang sistematis untuk membangun perangkat lunak yang berkualitas. Pondasi dalam pengembangan perangkat lunak yaitu process layer atau lapisan proses. Proses rekayasa perangkat lunak berfungsi untuk menghubungkan lapisan-lapisan teknologi dan memungkinkan pengembangan perangkat lunak yang rasional dan tepat waktu. Proses mendefinisikan suatu perangkat kerja yang harus dibangun sedemikian rupa sehingga dapat terjadi penghantaran teknologi rekayasa perangkat lunak yang efektif. Proses-proses perangkat lunak membentuk dasar bagi kendali manajemen proyek perangkat lunak dan membangun konteks dimana metode- metode teknis dapat diterapkan, dimana produkproduk kerja (model, dokumen, data, laporan, formulir, dsb) dibuat, dimana tonggak waktu dibuat, dimana kualitas dipastikan, dan dimana perubahan dapat diatur dengan tepat.

2.6. Konsep Dasar Aplikasi

Pada bagian ini akan dijelaskan definisi dasar dari istilah yang terkait dengan pembuatan aplikasi inventaris barang berbasis website *framework codeigniter*.

2.6.1 Pengertian Rancangan

Rancangan merupakan salah satu hal yang penting dalam membuat program. Adapun tujuan dari rancangan ialah untuk memberi gambaran yang jelas lengkap kepada pemrogram dan ahli teknik yang terlibat. Rancangan harus berguna dan mudah dipahami sehingga mudah digunakan. Rancangan

adalah sebuah proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta di dalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaannya.

Menurut Pressman yang dikutip oleh Buchari dkk dalam jurnal E-Journal Teknik Informatika Vol. 6 No. 1 (2015), rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan.

2.2.7. Pengertian Bangun

Menurut Pressman yang dikutip oleh Taufan dalam jurnal E-Journal Teknik Informatika Vol. 11 No. 1 (2017), bangun atau pembangunan adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada secara keseluruhan. Jadi dapat disimpulkan bahwa Rancang Bangun adalah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari 7 8 beberapa elemen yang terpisah kedalam suatu kesatuan yang utuh dan berfungsi. Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut atau memperbaiki sistem yang sudah ada.

2.2.8. Pengertian Aplikasi

Aplikasi menurut Dhanta dikutip dari Sanjaya (2015) adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word, Microsoft Excel*. Aplikasi berasal dari kata *application* yang artinya penerapan lamaran penggunaan. Menurut Jogiyanto dikutip oleh Ramzi (2016) aplikasi merupakan penerapan, menyimpan sesuatu hal, data, permasalahan, pekerjaan kedalam suatu sarana atau media yang dapat digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan yang ada sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari hal data, permasalahan, dan pekerjaan itu sendiri. Jadi aplikasi merupakan sebuah transformasi dari sebuah permasalahan atau pekerjaan berupa hal yang sulit difahami menjadi lebih sederhana, mudah dan dapat dimengerti oleh pengguna. Sehingga dengan adanya aplikasi, sebuah permasalahan akan terbantu lebih cepat dan tepat.

2.2.9. Pengertian Database

Menurut Stephens dan Plew didalam buku Simamarta dan Paryudi (2016:1), *Database* (basis data) adalah kumpulan data yang di simpan secara sistematis didalam komputer yang dapat diolah atau di manipulasi menggunakan perangkat lunak (*software*) program atau aplikasi untuk menghasilkan informasi.

2.2.10. Pengertian Website

“*Website* (bahasa Indonesia: situs *web*) merupakan kumpulan halaman yang menampilkan informasi data, teks, data gambar, data animasi, suara, video, atau gabungan semuanya yang dapat diakses melalui *internet*” (Gusdeka, 2019:15). Terdapat dua jenis aplikasi *web* yaitu yang bersifat statis dan dinamis dan fungsi salah satunya untuk mengelola data pada suatu perusahaan.

Menurut Triyono dalam Jurnal SENSI Vol.4 No.1 (2018:23), *website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau bergerak, animasi, suara, dan atau gabungan dari semuanya baik yang bersifat dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman.

Berdasarkan pengertian diatas, peneliti menyimpulkan bahwa *website* merupakan situs berupa halaman – halaman yang berisi informasi text, gambar, video, audio, animasi yang bersifat dinamis dan saling terkait.

2.2.7 Pengertian Framework Codeigniter

Framework Codeigniter merupakan salah satu framework *PHP* yang populer dan paling banyak digunakan, di Indonesia pun pengembang aplikasi berbasis *web* banyak menggunakan *framework PHP* ini. *Framework Codeigniter* disebut juga sebagai *Application Development Framework* yaitu sebuah *toolkit* untuk para pengembang aplikasi berbasis *web* yang menggunakan bahasa

pemrograman *PHP*. Menurut Rudianto (2015:43) *PHP* adalah Bahasa *server-side-scripting* yang menyatu dengan *HTML* untuk membuat halaman *website* yang dinamis.” Karena *PHP* merupakan *server-side-scripting* maka *sintaks* dan perintah-perintah *PHP* akan dieksekusi di server kemudian hasilnya akan dikirimkan ke *browser* dengan format *HTML*. Tujuannya yaitu untuk mempercepat dalam penulisan program karena *Codeigniter* menyediakan banyak *libraries* yang umum dan sering digunakan dengan cara penggunaan yang sangat mudah dan simpel, sehingga *programmer* dapat fokus pada pembuatan program dengan meminimalkan penulisan koding yang dibutuhkan.

2.2.11. Pengertian Inventaris Barang

Menurut Wibowo (2015), inventaris barang adalah kegiatan yang terdiri dari data masuknya barang, data retur serta data persediaan barang yang melaporkan seluruh transaksi keluar masuk barang dari perhari sampai perbulan.

Inventaris barang adalah daftar lengkap barang-barang yang ada di sebuah perusahaan atau instansi baik yang keluar maupun yang masuk, baik yang berwujud ataupun tidak. Daftar barang ini ditulis dalam satu buku khusus dan harus lengkap tanpa ada yang terlewat. Tujuannya untuk menganalisis alat penunjang aktivitas perusahaan, kekayaan dan modal.

2.7. *Tools* Pemodelan yang Digunakan

Pada proyek akhir ini *tools* yang digunakan adalah *UML*, *Use Case Diagram*, dan *Entity-Relationship Diagram*.

2.3.3. *UML*

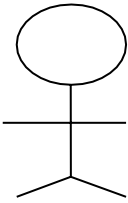
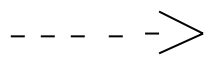
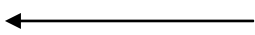
Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. *UML* merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. (Fatimah:2015). *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. *UML* hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada metodologi berorientasi objek.

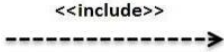
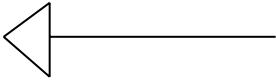


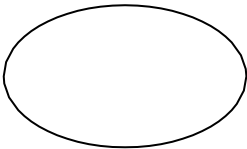

2.3.4. *Use Case Diagram*

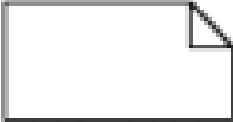
Menurut Mamed:2015 "*Use Case Diagram* adalah sesuatu atau proses merepresentasikan hal-hal yang dapat dilakukan oleh aktor dalam menyelesaikan sebuah pekerjaan." Menurut Shalahuddin dalam jurnal (Umar Al Faruq, 2015) mengungkapkan, bahwa "*Diagram use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut." (Yunahar: 2018).

Berikut simbol-simbol yang ada pada diagram *use case* :

Tabel 2.1 Simbol *Use Case* Diagram

| No. | Nama | Simbol | Keterangan |
|-----|-----------------------|--|--|
| 1. | <i>Actor</i> |  | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> . |
| 2. | <i>Depedency</i> |  | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>). |
| 3. | <i>Generalization</i> |  | Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>) |

| | | | |
|----|----------------------|--|---|
| 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> |  | Apa yang menghubungkan antara objek satu |
| 7. | Sistem |  | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8. | <i>Use Case</i> |  | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 9. | <i>Collaboration</i> |  | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk |

| | | | |
|-----|-------------|---|---|
| | | | menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10. | <i>Note</i> |  | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

2.3.4 Class Diagram

Menurut (Hendini, 2016) *Class Diagram* merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class Diagram* secara khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalization* dan *Aggregation*, attribute (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*.

Tabel 2.2 Multiplicity Class Diagram

| <i>Multiplicity</i> | Penjelasan |
|---------------------|------------|
|---------------------|------------|

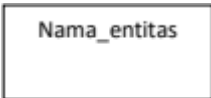

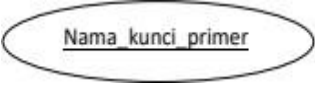
| | |
|---|---------------------|
| 1 | Satu dan hanya satu |
|---|---------------------|


| | |
|------|---|
| 0..* | Boleh tidak ada atau 1 atau lebih |
| 1..* | 1 atau lebih |
| 0..1 | Boleh tidak ada, maksimal 1 |
| n..n | Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4 |



2.3.5. ERD

Menurut (Minda Septiani, Nurul Afni, 2019) Pemodelan awal basis data yang paling banyak digunakan adalah ERD yang dikembangkan berdasarkan teori himpunan dalam bidang matematika.

Tabel 2.3 Simbol ERD

| Nama Simbol | Simbol | Deskripsi |
|----------------------|---|---|
| Entitas |  | Entitas merupakan data inti yang akan disimpan; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel. |
| Atribut |  | Kolom data yang butuh disimpan dalam suatu Entitas |
| Atribut Kunci Primer |  | Kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan. |

| | | |
|--------------------------------------|---|---|
| Atribut multinilai/multivalu e |  | Kolom data yang butuh disimpan dalam suatu entitas yang dapat |
|--------------------------------------|---|---|

| | | |
|----------------------|---|--|
| | | memiliki nilai lebih dari satu. |
| Relasi |  | Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja. |
| Asosiasi/Association |  | Penghubung antara relasi dan entitas di mana dikedua ujungnya memiliki <i>multiplicity</i> . |

2.8. Tools Pembangunan Aplikasi

Berikut merupakan definisi dan penjelasan tools yang dibutuhkan dalam pembuatan dan pengembangan aplikasi.

2.4.5. PHP

Menurut (Sitohang, 2019) PHP adalah PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman web berbasis *server (server side)* yang mampu memarsing kode PHP dari kode dengan ekstensi PHP sehingga menghasilkan tampilan *website* yang dinamis disisi client.

2.4.5.1 Codeigniter

Menurut (Subagia, 2018) *CodeIgniter* merupakan sebuah web *framework* yang dikembangkan oleh Rick Ellis dari Ellis Lab. *CodeIgniter* dirancang untuk menjadi sebuah web framework yang ringan dan mudah digunakan. Bahkan dari Rasmus Lerdorf, sang pencipta bahasa pemrograman PHP mengatakan bahwa *CodeIgniter* merupakan web *framework* yang mudah, cepat, dan handal. Dirilis pada dokumentasi *CodeIgnite, CodeIgniter*

merupakan *toolkit* bagi orang yang ingin membangun aplikasi web menggunakan PHP.

Berikut adalah konsep *Model-View-Controller* yang diterapkan pada *CodeIgniter*:

1. *Model*, menggambarkan struktur data. Biasanya kelas model akan berisi fungsi yang digunakan untuk mengambil, menambah dan memperbaharui informasi yang ada di *database*.
2. *View*, informasi yang diperlihatkan kepada user. View adalah halaman web yang terdiri dari HTML, CSS dan *Javascript*, akan tetapi pada *CodeIgniter*, *view* dapat juga sebagai potongan halaman seperti *header* atau *footer*. Bahkan dapat juga halaman RSS atau tipe halaman lainnya.
3. *Controller*, perantara *Model*, *View*, dan *resource* lainnya yang dibutuhkan untuk menangani HTTP *request* dan menghasilkan halaman web.

2.4.5.2 HTML

Menurut (Enterprise, 2018) HTML, digunakan untuk membuat struktur halaman *website*. Bisa dibidang secara umum bahwa HTML digunakan untuk mendesain *website*, meskipun dalam praktiknya HTML tidak berdiri sendiri sebab akan digabungkan dengan CSS atau *Script* lain, seperti *Javascript*. Pada kesempatan ini, kita akan menganggap HTML sebagai program untuk membuat desain sebuah *website*, dimana kita bisa menulis teks, memasukkan gambar, membuat *form*, dan sebagainya.

2.4.6. MySQL

Menurut (Enterprise, 2018) MySQL merupakan *server* yang melayani *database*. Untuk membuat dan mengelola *database*, kita dapat mempelajari pemrograman khusus yang disebut *query* (perintah) SQL. *Database* sendiri dibutuhkan jika kita ingin menginput data dari *user* menggunakan *form* HTML untuk kemudian diolah PHP agar bisa disimpan kedalam *database* MySQL.

2.4.7. XAMPP

Menurut (Enterprise, 2018) XAMPP merupakan *server* yang paling banyak digunakan untuk keperluan belajar PHP secara mandiri, terutama bagi *programmer* pemula. Selain gratis, fiturnya tergolong lengkap dan gampang digunakan oleh *programmer* PHP tingkat awal, yang perlu dilakukan hanyalah menjalankan *module Apache* yang ada di dalam XAMPP tersebut.

2.4.8. Visual Studio Code

Menurut (Lardinois, 2015) Visual Studio Code adalah editor source code yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS. Ini termasuk dukungan untuk debugging, GIT Control yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode refactoring. Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, shortcut keyboard, dan preferensi. Visual Studio Code gratis dan open-source, meskipun unduhan resmi berada di bawah lisensi proprietary.

