**BAB 2**

**LANDASAN TEORI**

* 1. **Framework Codeigniter**

Framework atau dalam bahasa indonesia dapat diartikan sebagai “kerangka kerja” merupakan kumpulan dari fungsi-fungsi/prosedur-prosedur dan class-class untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programer, tanpa harus membuat fungsi atau class dari awal.

Budi (2015:2), “Framework adalah suatu struktur konseptual dasamr yang digunakan untuk memecahkan atau menangani suatu masalah kompleks. Istilah ini sering digunakan antara lain dalam bidang perangkat lunak untuk menggambarkan suatu desain sistem perangkat lunak yang dapat digunakan kembali”.

Framework atau sering disebut dengan kerangka kerja. Developer menggunakan framework agar memudahkan mereka untuk membuat sertamengembangkan suatu aplikasi atau software. Pada framework itu sendiri berisi kumpulan dari fungsi-fungsi dasar maupun perintah yang dapat digunakan dalam mengembangkan suatu software. dengan tujuan agar supaya software yang dibangun menjadi lebih cepat serta terstruktur. (Fauzan & Roza, 2019:126)

CodeIgniter adalah sebuah web application network yang bersifat open source yang digunakan untuk membangun aplikasi php dinamis. CodeIgniter menjadi sebuah framework PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Selain ringan dan cepat, CodeIgniter juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih CodeIgniter sebagai framework pilihannya. Karena kelebihan-kelebihan yang dimiliki oleh CodeIgniter, pembuat PHP Rasmus Lerdorf memuji CodeIgniter di frOSCon (Agustus 2008) dengan mengatakan bahwa dia menyukai CodeIgniter karena “it is faster, lighter and the least like a framework.”

Salah satu alasan mengapa orang menggunakan famework terutama dalam membangun sebuah website yaitu kemudahan – kemudahan yang ditawarkan. Di dalam sebuah framework biasanya sudah tersedia struktur website yang baik, standar coding, *best practice, design pattern, dan common function*. Dengan arti lain, kita bisa membuat program tanpa harus berfikir banyak masalah struktur, standar coding dan lain – lain.

Menurut (Zulkifli Ahmad., 2018:168). “Framework CodeIgniter adalah sekumpulan library yang diorganisasikan pada sebuah rancangan arsitektur untuk memberikan kecepatan, ketepatan, kemudahan dan konsistensi di da-lam pengembangan aplikasi dari definisi tersebut”. Dari definisidi atas, penulis menyimpulkan bahwa framework merupakan kumpulan library atauclass yang masing-masing memiliki fungsi tersendiri dalam mem-bangun sebuah kerangka kerja”.

Sebuah Framework CodeIgniter terdiri dari:

1. Model

Model mencakup semua proses yang terkait dengan pemanggilan struktur data baik berupa pemanggilan fungsi, input processing atau mencetak outputke dalam browser.

1. View

View mencakup semua proses yang terkait layout output. Bisa dibilang untuk menaruh template interface website atau aplikasi.

1. Controller

Controller mencakup semua proses yang terkait dengan pemanggilan databasedan kapsulisasi proses proses utama. Jadi semisal dibagian ini ada filebernama member.php, maka semua proses yang terkait dengan member akan di kapsulisasi/ dikelompokan dalam file ini.

Menurut Hidayatullah & Kawistara (2017:283-285), “Model View Controller atau yang kita sebut dengan MVC adalah suatu metode yang memisahkan data logic (Model) dari presentation logic (View) dan process logic (Controller) atau secara sederhana adalah memisahkan antara desain interface, data, dan proses”.

Menurut Raharjo (2015:2), “Framework adalah suatu kumpulan kode berupa Pustaka (library) dan alat (tool) yang dipadukan sedemikian rupa menjadi satu kerangka kerja (framework) guna memudahkan dan mempercepat proses pengembangan aplikasi web. Jadi, framework adalah kumpalan – kumpalan potngan program yang dipadukan menjadi satu kerja yang digunakan untuk membantu dalam pembuatan sebuah aplikasi”.

Beberapa keuntungan yang didapat dalam penggunaan framework adalah (Hidayatullah & Kawistara, 2017:283):

* + 1. Menghemat waktu pengembangan

Dengan library yang telah disediakan oleh framework maka tidak perlu lagi memikirkan hal – hal dasar atau hal – hal umum yang sudah dibuat pada library, jadi anda hanya focus ke proses bisnis yang akan dikerjakan.

* + 1. Penggunaan ulang program/kode

Dengan menggunakan framework maka pekerjaan anda akan memiliki struktur yang baku, sehingga kita dapat menggunakan program/kode itu kembali untuk pekerjaan lainnya.

* + 1. Bantuan komunitas

Pada umumnya setiap framework yang besar mempuanyai komunitas dimana komunitas inilah yang siap membantu jika ada permasalahan, selain itu juga bisa berbagi ilmu sehingga dapat meningkatkan kemampuan pemrograman anda.

* + 1. Kumpulan program terbaik

Sebuah framework merupakan kumpulan program terbaik yang sudah teruji, sehingga dapat meningkatkan kualitas program/kode anda.

Menurut Yenda (2016:18), “Codeigniter adalah sebuah framework yang dibuat menggunakan bahasa pemrograman PHP yang bertujuan untuk memudahkan para programer web untuk membuat atau mengembangkan aplikasi berbasis web”.

Sementara menurut Basuki (2017:4), “CodeIgniter adalah sebuah framework PHP yang kecil, sederhana, mudah digunakan dan cukup handal untuk membangun aplikasi berbasis web”.

CodeIgniter merupakan sebuah toolkit yang ditujukan untuk orang yang ingin membangun aplikasi web dalam bahasa pemrograman PHP. Beberapa keunggulan yang ditawarkan oleh CodeIgniter adalah sebagai berikut (Raharjo, 2015:3) :

1. CodeIgniter adalah framework yang bersifat free dan open source.
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 6MB.
3. Aplikasi yang dibuat menggunakan CodeIgniter bisa berjalan cepat.
4. CodeIgniter menggunakan pola desain Model-View-Controller (MVC) sehingga satu file tidak terlalu berisi banyak kode. Hal ini menjadikan kode lebih mudah dibaca, dipahami, dan dipelihara dikemudian hari
5. CodeIgniter dapat diperluas sesuai dengan kebutuhan.
6. CodeIgniter terdokumentasi dengan baik. Informasi tentang pustaka kelas dan fungsi yang disediakan oleh CodeIgniter dapat diperoleh melalui dokumentasi yang disertakan di dalam paket distribusinya.

Adapun sistem kerja Codeigniter memiliki beberapa tahapan yang harus dilewati. Alur kerjanya dapat dilihat sebagai berikut (Rais dan Sirad, 2019:28):

1. Index

Index.php merupakan controller awalyang menginisialisasikan kebutuhan untukmenjalankan CodeIgniter.

1. Router/Routing

Router/Routing merupakan bagian yangmenentukan kegiatan yang harusdilakukan ketika ada request/permintaan dari client/browser.

1. Caching

Caching merupakan bagian yangmengecek apakah data sudah pernahdiminta atau belum. Jika cache dalamkeadaan aktif, maka akan langsungdikirimkan kepada client atau browser dengan mengabaikan alur kerja normal.

1. Security

Security, sebelum aplikasi dikirimkan,maka akan lebih dahulu data difiltersebagai keamanan.

1. Controller

Controller merupakan pengendali dari jalannya aplikasi, dan akan segeramemproses sesuai request/permintaanyang diminta, yaitu models, libraries,helpers, plugins dan script.

1. View

View merupakan bagian untuk menyajikan suatu informasi ke client/browser sesuai dengan permintaan yang diminta.

Menurut Blanco & Upton (20019:7), “Codeigniter adalah powerful open source PHP framework yang mudah dikuasai, dibangun untuk PHP programmers yang membutuhkan toolkit sederhana dengan baik untuk membuat full-featured web applications. Codeigniter adalah MVC framework yang di desain untuk mempermudah penggunanya.

Menurut Daqiqil Id (2016:132), “Codeigniter adalah sebuah framework PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Dilengkapi banyak library dan helper yang berguna di dalamnya dan tentunya mempermudah proses development. Codeigniter akan mempermudah dan mempercepat pembuatan pengembangan website dinamis tanpa harus kesulitan membuat desain”.

Menurut Budi (2015:2), “Framework adalah suatu struktur konseptual dasamr yang digunakan untuk memecahkan atau menangani suatu masalah kompleks. Istilah ini sering digunakan antara lain dalam bidang perangkat lunak untuk menggambarkan suatu desain sistem perangkat lunak yang dapat digunakan kembali”.

Menurut Raharjo (2015:3), “Codeigniter adalah framework web untuk bahasa php (Hypertext Prepoccesor), yang dibuat oleh Rick Ellis yang digunakan untuk mempermudah pada developer dalam mengembangkan suatu aplikasi web”.

Menurut Hakim (2010:8), “CodeIgniter adalah sebuah framework PHP yang dapat membantu mempercepat developer dalam pengembangan aplikasi web

berbasis PHP dibanding jika menulis semua kode program dari awal”.

Menurut Budi Raharjo (2015:3), “CodeIgniter adalah framework web

untuk bahasa pemrograman PHP yang dibuat oleh Rick Ellis pada tahun

2006, penemu dan pendiri EllisLab. EllisLab adalah suatu tim kerja yang

berdiri pada tahun 2002 dan bergerak di bidang pembuatan software dan

tool untuk para pengembang web”

Menurut Arrhioui et al (2017:56), “Codeigniter adalah kerangka kerja

pengembangan aplikasi PHP berdasarkan arsitektur yang terstruktur. Codeigniter memiliki tujuan untuk memberikan alat bantu yang dibutuhkan seperti helpers and libraries untuk mengimplementasi tugas yang biasa dilakukan. Dengan demikian, pengembangan proyek menjadi lebih mudah dan cepat. Dan pengembang tidak perlu menulis lagi dari awal”.

Adapun beberapa keuntungan menggunakan CodeIgniter, diantaranya:

1. Gratis

CodeIgniter berlisensi dibawah Apache/BSD opensorce.

1. Ditulis Menggunakan PHP 4

Meskipun CodeIgniter dapat berjalan di PHP 5, namun sampai saat ini kode program CodeIgniter masih dibuat dengan menggunakan PHP 4.

1. Berukuran Kecil

Ukuran CodeIgniter yang kecil merupakan keunggulan tersendiri. Dibanding dengan framework lain yang berukuran besar.

1. Menggunakan Konsep MVC

CodeIgniter menggunakan konsep MVC yang memungkinkan pemisahan layer application-logic dan presentation.

1. URL yang Sederhana

Secara default, URL yang dihasilkan CodeIgniter sangat bersih dan Serach Engine Friendly (SEF).

1. Memiliki Paket Library yang Lengkap

CodeIgniter mempunyai library yang lengkap untuk mengerjakan operasioperasi yang umum dibutuhkan oleh sebuah aplikasi berbasis web, misalnya mengakses database, mengirim email, memvalidasi form, menangani session dan sebagainya

1. Extensible

Sistem dapat dikembangkan dengan mudah menggunakan plugin dan helper, atau dengan menggunakan hooks.

1. Tidak Memerlukan Template Engine

Meskipun CodeIgniter dilengkapi dengan template parser sederhana yang dapat digunakan, tetapi hal ini tidak mengharuskan kita untuk menggunakannya.

1. Dokumentasi Lengkap dan Jelas

Dari sekian banyak framework, CodeIgniter adalah satu-satunya framework dengan dokumentasi yang lengkap dan jelas.

1. Komunitas

Komunitas CodeIgniter saat ini berkembang pesat. Salah satu komunitasnya bisa dilihat di (<http://codeigniter.com/forum/>)

Menurut Blanco & Upton (2009:7), “CodeIgniter adalah powerful open source PHP framework yang mudah dikuasai, dibangun untuk PHP programmers yang membutuhkan toolkit sederhana dan baik untuk membuat full-featured web applications. CodeIgniter adalah MVC framework yang di design untuk mempermudah penggunanya”.

Menurut Menurut Ridha (2007:2), “CodeIgniter adalah sebuah frameworkPHP yang bersifat Open Source dan dapat digunakan untuk mempercepat pengembang dalam membuat aplikasi web”.

Menurut Andriyani (2016:164), “CodeIgniter (CI) merupakan aplikasi open source yang berupa framework dengan model MVC (Model, View, Controller) untuk membangun website dinamis menggunakan PHP”

Menurut Sulistiono (2018:7), “CodeIgniter aplikasi open source yang berupa framework untuk membangun bahasa pemrograman PHP. Tujuannya memungkinkan penegmbangan proyek yang lebih cepat dari pada penulisan kode dasar atau kode terstruktur.”

Menurut Kasman (2015:116), “*Framework* adalah komponen pemrograman yang siap re-use (bisa digunakan ulang) kapan saja, sehingga programmer tidak harus membuat skrip yang sama untuk tugas yang sama. Misalkan programmer ingin halaman-halaman web menampilkan data dengan paginasi (*paging*) halaman, *framework* telah menyediakan fungsi *paging* tersebut sedangkan programmer cukup menggunakan fungsi tersebut pada saat *coding*, tetapi tentu dengan kaidah-kaidah yang ditetapkan oleh masing-masing *framework”***.**

Menurut Rosa A.S (2011:182), “Framework adalah kerangka kerja yang memudahkan programmer untuk membuat sebuah aplikasi sehingga programmer akan lebih mudah melakukan perubahan (customize) terhadap aplikasinya dan dapat memakainya kembali untuk aplikasi lain yang sejenis”.

Menurut Betha Sidik dalam Destiningrum (2017,32), “Framework adalah kumpulan intruksi-intruksi yang dikumpulkan dalam classdan function-function dengan fungsi masingmasing untuk memudahkan developer dalam memanggilnya tanpa harus menuliskan syntax program yang sama berulang-ulang serta dapat menghemat waktu”.

Menurut Betha Sidik dalam Destiningrum (2017,32), “CodeIgniter adalah sebuah framework php yang bersifat open source dan menggunakan metode MVC (Model, View, Controller) untuk memudahkan developer atau programmer dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal”. Dalam situs resmi CodeIgniter, menyebutkan bahwa CodeIgniter merupakan framework PHP yang kuat dan sedikit bug”.

menurut Gungun (2011:7), “Codeigniter adalah aplikasi open source yang berupa framework dengan model MVC (Model, View, Controller) untuk membangun website dinamis”.

Menurut Purbadian (2016:10), “Framework adalah kerangka kerja atau sekumpulan file-file yang sudah ter-include, yang mana di dalam file tersebut terdapat perintah kode program dan fungsi dasar untuk melakukan tugas tertentu”.

Pengertian tentang framework tersebut dapat disimpulkan bahwa Framework adalah kumpulan fungsi yang dapat membantu programmer dalam menangani masalah dalam pemrograman seperti salah satu contoh nya koneksi ke database.

Menurut Purbadian (2016:10), “CodeIgniter adalah sebuah framework yang dibuat menggunakan bahasa pemrograman PHP yang bertujuan untuk memudahkan para programmer web untuk membuat atau mengembangkan aplikasi berbasis web”.

Kesimpulan dari pengertian tersebut bahwa CodeIgniter adalah Framework PHP yang di dalamnya terdapat fitur lengkap aplikasi web yang sudah dikemas menjadi satu.

* 1. **Perancangan Perangkat Lunak**

Rekayasa perangkat lunak adalah aplikasi dari sebuah pendekatan kuantifiabel, disiplin dan sistematis kepada pengembangan, operasi dan pemeliharaan perangkat lunak; yaitu aplikasi dari Rekayasa perangkat lunak.

Menurut Sommerville (2016:113), “Rekayasa Perangkat lunak adalah teknik yang baik memilih alat bantu dalam perencanaan maupun dalam penerapan perangkat lunak, menilai kulaitas perangkat lunak, mengkordinasi, mengontrol dan mengatur pelaksanaan pembuatan perangkat lunak”.

Menurut Pressman (2015:4), “Perangkat lunak adalah instruksi (program komputer) yang ketika dijalankan menyediakan fitur, fungsi, dan kinerja yang diinginkan struktur data yang memungkinkan program untuk memanipulasi informasi secara memadai, dan informasi deskriptif baik dalam bentuk *hard* *copy* maupun virtual yang menjelaskan operasi dan penggunaan program”.

Kerangka kerja proses yang umum bagi rekayasa perangkat lunak terdiri atas lima aktivitas yaitu sebagai berikut (Pressman & Maxim, 2019:10-11) :

1. Komunikasi. Sebelum pekerjaan teknis apa pun dapat dimulai, sangatlah penting untuk berkomunikasi dan berkolaborasi dengan pelanggan (dan para pemangku kepentingan stakeholder yang lain).
2. Perencanaan. alur yang rumit dapat disederhanakan jika terdapat suatu peta. Suatu proyek perangkat lunak pada dasarnya merupakan suatu alur yang rumit, dan kegiatan perencanaan perangkat lunak tersebut menciptakan suatu “peta” yang membantu membimbing tim perangkat lunak ketika mereka melakukan suatu perjalanan.
3. Pemodelan. seorang rekayasawan perangkat lunak membuat model-model untuk memahami kebutuhan perangkat lunak maupun rancangan-rancangan yang akan memenuhi kebutuhan tersebut.
4. Konstruksi. Kegiatan ini berupa desain harus dibangun. Aktivitas ini menggabungkan pembuatan kode (baik manual atau otomatis) dan pengujian yang diperlukan untuk mengungkap kesalahan dalam Kode.
5. Penyebaran. Perangkat lunak (sebagai entitas yang lengkap atau sebagai tambahan yang diselesaikan sebagian) dikirimkan ke pelanggan yang mengevaluasi produk yang dikirimkan dan menyediakan umpan balik berdasarkan evaluasi.

Karakteristik dari perangkat lunak adalah sebagai berikut(Sukamto dan Shalahudin, 2013:87) :

1. Perangkat lunak dibangun dengan rekayasa (*software engineering*) bukan diproduksi secara manufaktur atau pabrikan.
2. Perangkat lunak tidak pernah using (*“wear out”*) karena kecacatan dalam perangkat lunak dapat diperbaiki.
3. Barang produksi pabrikan biasanya komponen barunya akan terus diproduksi, sedangkan perangkat lunak biasanya terus diperbaiki seiring bertambahnya kebutuhan.

Rekayasa perangkat lunak adalah sebuah teknologi yang terdiri dari lapisan, yaitu(Pressman, 2019:20-21) :

1. Fokus pada kualitas

Pendekatan teknik apapun harus bersandar pada komitmen organisasi terhadap suatu mutu. Total kualitas manajemen dan filosofi yang sama mendorong budaya perbaikan proses yang berkesinambungan dan budaya inilah yang pada akhirnya mengarah pada pengembangan pendekatan yang semakin lebih dewasa untuk rekayasa perangkat lunak. Fondasi yang mendukung rekayasa perangkat lunak adalah fokus pada kualitas.

1. Proses

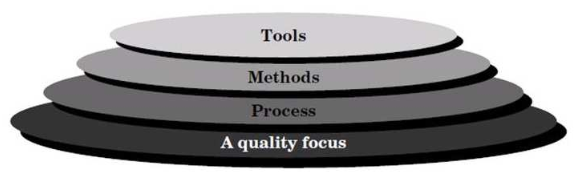
Dasar untuk rekayasa perangkat lunak adalah lapisan proses. Proses pada rekayasa perangkat lunak adalah perekat yang memegang teknologi lapisan bersama-sama dan memungkinkan pengembangan perangkat lunak yang rasional dan tepat waktu. Proses mendefinisikan sebuah *framework*untuk sekumpulan *key process area* yang harus dibangun demi keefektifan penyapaian teknologi pengembangan RPL.

1. Metode

Metode perangkat lunak menyediakan teknik “bagaimana” untuk membangun perangkat lunak. Metode mencakup tugas yang mencakup analisis kebutuhan, memodelkan desain, program konstruksi, pengujian dan dukungan.

1. Alat Bantu

Alat bantu otomatis tau semi-otomatis menyediakan dukungan untuk proses dan metode. Ketika alat-alat diintegrasikan sehingga informasi yang dibuat oleh salah satu alat dapat digunakan oleh alat lainnya.



**Gambar 2.2** Lapisan Rekayasa Perangkat Lunak (Pressman, 2019:20-21)

Beberapa kegiatan umum yang penting dalam proses perangkat lunak adalah sebagai berikut(Pressman, 2010:31) :

1. Spesifikasi

Penjelasan-penjelasan tentang fungsionalitas dan batas-batas dalam pengembangan suatu perangkat lunak.

1. Desain dan implementasi

Perangkat lunak yang mendukung spesifikasi harus dibuat dan direncanakan.

1. Validasi

Perangkat lunak harus divalidasi agar sesuai dengan kebutuhan pengguna.

Dua alasan penting dari perancangan aplikasi pengolahan data adalah sebagai berikut(Sommerville, 2015:48) :

1. Lebih dan lebih, individu dan masyarakat bergantung pada sistem aplikasi. Kita harus dapat menghasilkan sistem yang dapat diandalkan, dipercaya, ekonomis dan cepat.
2. Biasanya lebih murah, dalam pemakaian jangka panjang, menggunakan metode teknik dan perancangan untuk sistem aplikasi lebih baik sekedar sistem, mayoritas biaya adalah untuk merubah sistem aplikasi setelah diterapkan.

Proses perangkat lunak (*software process*) adalah sekumpulan aktifitas yang memiliki tujuan untuk mengembangkan atau mengubah perangkat lunak. Secara umum proses perangkat lunak terdiri dari (Rosa dan Shalahuddin, 2018:11):

* 1. Pengumpulan Spesifikasi (*Spesification*). Mengetahui apa saja yang harus dapat dikerjakan system perangkat lunak dan batasan pengembangan perangkat lunak.
  2. Pengembangan (*Development*) Pengembangan perangkat lunak untuk menghasilkan system perangkat lunak.
  3. Validasi (*Validation*). Memeriksa apakah perangkat lunak sudah memenuhi keinginan pelanggan (*Customer*).
  4. Evolusi (*Evolution*). Mengubah perangkat lunak untuk memenuhi perubahan kebutuhan pelanggan (*Customer*).

Dari beberapa buka teks, telah dijelaskan dengan sangat gamblang bahwa RPL meliputi beberapa pokok bahasan pentung antara lain :

1. Domain Engineering

Mampu memahami permasalahan yang muncul dan akan dijalaskan sebagai proyek perangkat lunak.

1. Requirement Engineering

Mampu memahami kebutuhan pengguna sekaligus melikukan pemecahan masalah

1. Software Design

Mampu memahamu serta mengimplementasikan perancagan perngat lunak termasuk didalamnya aspek HCI (Human Computer Interaction)

1. Development

Dalam proses pengembangan sebuah perangkat lunak nantinya akan melibatkan pembelajaran mengenai algoritma, bahasa pemrogaman yang diguakan serta teknik yang berkaitan didalamnya seperti basis data dan sistem informasi.

1. Operatians

Operasioal perangkat lunak dapat dipisahkan menjadi dua bagian penting yakni pada saat proses testing yang seharusnyadilakukan bersama-sama antara pengembang dan pengguna perangkat lunak, dan proses implemantasi yang didalamnya bisa terdapat langkat-langkah awal seperti pelatihan dan perbaikan pasca produksi.

1. Maintenance

Dalam ruang lingkup yang terahkir ini selain melakukan pemeliharaan terhadap aspek perangkat lunak seperti basis data, intalasi juga didalamnya terdapat proses dokumentasi dari pengembang perangkat lunak. Proses dokumentasi nantinya tidak hanya ditunjukan untuk pengguna, tetapi juga unuk pengembangan perangkat lunak itu sendiri jika nantinya terjadi proses revisi atau penggantian lebih lanjut.

Sedangakan jika dilihat dari proses sebuah RPL sendiri melibatkan bebebrapa unsur antara lain :

1. Software Engineers atau pengembangan perangkat luank.
2. Software atau perangkat lunak.
3. User atau pengguna

Lila Setiyani (2018:205), “Perancangan perangkat lunak ini menyediakan secara rinci terkait dengan arsitektur perangkat lunak, struktur data yang digunakan dalam perangkat lunak, antarmua, komponen dan subsistem yang diperlukan untuk implementasi perangkat lunak. Sasaran dari perancangan perangkat lunak yang memperlihatkan kekuatan, komoditi serta kenyamanan”.

Perancangan dalam penmbangunan perangkat lunak merupakan upaya untuk mengonstruksi sebuah sistem yang akan memberikan kepuasan akan spesifikasi kebutuhan fungsional, memenuhi target, memenuhi kebutuhan secara implisit atau eksplisit dari segi performa maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu, dan perangkat. Kualitas ini biasanya dinilai dari segi kepuasan pengguna terhadap perangkat yang digunakan (Rosa & Shalahuddin, 2018:23).

Proses perangkat lunak (*software process*) adalah sekumpulan aktifitas yang memiliki tujuan untuk mengembangkan atau mengubah suatu perangkat lunak. Secara umum proses perangkat lunak terdiri dari (Rosa & Shalahuddin, 2018:11) :

1. *Spesification*

Mengetahui apa saja yang harus dapat dikerjakan atau diperlukan oleh sistem perangkat lunak dan batasan pengembangan perangkat lunak.

1. *Development*

Merupakan pengembangan perangkat lunak untuk menghasilkan sistem perangkat lunak.

1. *Validation*

Memeriksa apakah perangkat lunak sudah dapat memenuhi keinginan pelanggan (*custome*).

1. *Evolution*

Mengubah perangkat lunak untuk memenuhi perubahan kebutuhan pelanggan (*customer*).

Kesimpulannya rekayasa perangkat lunak adalah suatu ilmu dalam menerapkan suatu prinsip rekayasa tentang program yang dapat membantu dalam mempermudah aktivitas dengan spesifikasi yang dapat dikembangkan.

* 1. **Perancangan Antarmuka**

Perancangan antarmuka pengguna menciptakan media komunikasi yang efektif diantara manusia dan komputer. Perancangan antarmuka pengguna bekerja dengan cara mengidentifikasi objek danOup aksi, kemudian membuat tata letak (*layout*) yang membentuk basis untuk pembuatan suatu prototipe antarmuka. Sistem antarmuka juga merupakan bagian dari suatu sistem informasi yang melibatkan *input* dan *output* yang memerlukan intervensi manusia. Antarmuka pengguna adalah suatu bagian dari suatu sistem informasi yang memerlukan interaksi pengguna agar terciptanya masukan dan keluaran.

Salah satu pendekatan dalam merancang antarmuka pengguna adalah *User Centered Design* (UCD) yang melibatkan pengguna pada proses desain. Pengguna terlibat sejak tahap analisis awal kebutuhan pengguna untuk pengujian dan evaluasi. Konsep *User Centered Design* (UCD) yaitu “*In user – centered design, the users are the center focus”* maksudnya segala sifat, konteks, pengguna, dan lingkungan produk semua didasarkan pengalaman pengguna. IBM *User Centered Design* (UCD) *development community*, sebuah komunitas penelitian terkemuka yang berfokus pada *User Centered Design* (UCD), menjelaskan bahwa ada enam prinsip, yaitu : *set the business goals*, *understand the user*, *Assessing competitiveness*, *desigining the total user experience*, *evaluating the design*, *maintenance and support*.

Menurut Ibnu Akil (2018:14), “*Interface* atau antarmuka adalah sekumpulan dari deklarasi funsi-fungsi tanpa metode, dimana metodenya akan direalisasikan oleh kelas yang mengimplementasikan *interface* tersebut”.

Menurut Pressman (2015:321), “Antarmuka harus menyajikan dan memperoleh informasi secara konsisten. Ini artinya bahwa (1) semua informasi *visual* diatur menurut aturan desain yang dipertahankan di seluruh tampilan layar, (2) mekanisme input adalah dibatasi pada set terbatas yang digunakan secara konsisten di seluruh aplikasi, dan (3) mekanisme untuk *menavigasi* dari tugas ke tugas secara konsisten didefinisikan secara konsisten dan dilaksanakan”.

Desain antarmuka terdapat tiga *Golden Rules* atau aturan emas (Pressman, 2015:318) :

1. Pengguna pemula, mereka pada dasarnya hampir-hampir tidak memiliki pengetahuan sintaksis tentang sistem dan hanya memiliki sedikit pengetahuan semantik tentang aplikasi atau penggunaan komputer secara umum.
2. Pengguna berpengetahuan yang tidak terlalu sering menggunakan sistem, mereka pada dasarnya memiliki pengetahuan semantik yang cukup tentang aplikasi, tetapi hanya memiliki pengetahuan sintaksis yang relatif sedikit tentang penggunaan antarmuka.
3. Pengguna berpengetahuan yang sering menggunakan sistem, mereka pada dasarnya memiliki pengetahuan yang sangat baik tentang sintaksis dan semantik aplikasi dan sering dinamakan sebagai pengguna yang ahli, yang sering menggunakan tombol-tombol pemercepat (*shortcut*) dan modus-modus pemercepat interaksi yang lainnya.
   1. **Perancangan Sistem**

Perancangan sistem adalah sebagian gambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi. Untuk mencapai tujuan pada analisa sistem harus dapat mencapai sasaran-sasaran sebagai berikut :

1. Perencanaan sistem harus berguna, mudah dipahami dan nantinya mudah digunakan.
2. Rancangan sistem harus dapat mendukung tujuan utama perusahaan sesuai dengan yang telah didefinisikan pada tahap perancangan sistem yang dilanjutkan pada analisa sistem.
3. Perancangan sistem harus efisien dan efektif untuk dapat mendukung pengolahan transaksi.

Menurut Prehanto (2020:3), “Sistem merupakan bagian-bagian komponen dikumpulkan yang memiliki hubungan satu sama lain baik fisik maupun nonfisik yang bersama-sama dalam bekerja demi tujuan yang dituju secara harmonis”.

Menurut Muslihudin & Oktafianto (2016:2), “Sistem adalah sekumpulan komponen atau jaringan kerja dari prosedur-prosedur yang saling berkaitan dan saling bekerjasama membentuk suatu jaringan kerja untuk mencapai sasaran atau tujuan tertentu”.

Adapun karakteristik sistem sebagai berikut (Anggraeni & Irviani, 2017:24):

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerjasama membentuk suatu komponen sistem.

1. Batasan Sistem

Merupakan daerah yang membatasi suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya.

1. Subsistem

Bagian-bagian dari sistem yang beraktivitas dan berinteraksi satu sama lain untuk mencapai tujuan dengan sasaran masing-masing.

1. Lingkungan Luar Sistem

Suatu sistem yang ada diluar batas sistem yang dipengaruhi oleh operasi sistem.

1. Penghubung Sistem

Media penghubung antara suatu sub sistem dengan subsistem lainnya. Penghubung ini memungkinkan berbagai sumber daya mengalir dari suatu subsistem ke subsistem lainnya.

1. Masukan Sistem

Energi yang masuk ke dalam perawatan sistem berupa perawatan dan sinyal. Masukkan perawatan adalah energi yang dimasukkan supaya sistem tersebut dapat berinteraksi.

1. Keluaran Sistem

Hasil energi yang diolah dan diklasifikasi menjadi keluaran yang berguna dan sisa pembuangan.

1. Pengolahan Sistem

Suatu sistem dapat mempunyai bagian pengolahan yang akan mengubah masukkan menjadi keluaran.

1. Sasaran Sistem

Tujuan yang ingin dicapai oleh sistem, akan dikatakan berhasil apabila mengenai sasaran dan tujuan.

Menurut Satzinger, Jackson, dan Burd (2012:5), Perancangan Sistem adalah sekumpulan aktivitas yang menggambarkan secara rinci bagaimana sistem akan berjalan. Hal itu bertujuan untuk menghasilkan produk perangkat lunak yang sesuai dengan kebutuhan user.

Perancangan Sistem yaitu rancangan atau susunan sistem yang akan dibangun. Tedapat 3 Jenis Perancangan sistem yaitu :

1. Perancangan databese. Sejumlah kumpulan data yang tersimpan di dalam media penyimpanan sekunder yang dipakai untuk menyimpan data-data panjang yang digunakan untuk inputan sistem kemudian data tersebut diolah menjadi data output/ keluaran sistem.
2. Perancangan proses. Menjelaskan sebuah proses bekerjanya sebuah sistem untuk melakukan pengolahan data input menjadi data outpu menggunakan fungsi-fungsi yang sudah direncanakan.
3. Perancangan Interface. Bagian software yang bisa dipakai oleh end user yang bisa dilihat pada layar monitor apabila program dijalankan.
   * 1. **Perancangan Masukkan *(Input)***

Menurut Jogiyanto (2015:52), “Input adalah awal dimulainya proses informasi. Bahan mentah dari informasi adalah data yang terjadi dari transaksi-transaksi yang dilakukan oleh organisasi. Data dari hasil transaksi merupakan masukan untuk sistem informasi”.

Terdapat beberapa tahap dalam proses input yang utama, antara lain (Jogiyanto, 2016:18) :

1. Penangkapan data *(data capture),* merupakan proses mencatat kejadian nyata yang terjadi akibat transaksi yang dilakukan oleh organisasi kedalam dokumen dasar merupakan bukti transaksi.
2. Penyiapan data *(data preparation),* yaitu merubahd ata yang telah dirangkap kedalam bentuk yang dapat dibaca oleh mesin.
3. Pemasukkan data *(data entry),* merupakan proses membaca atau memasukkan data kedalam komputer.

Sedangkan pedoman-pedoman yang berfokus pada input data (Pressman, 2019:27):

1. Meminimalkan jumlah aksi input yang dibutuhkan oleh pengguna.
2. Jagalah konsistrensi di antara tampilan informasi dan input data.
3. Ijinkan pemakaian mengkostumasi input.
4. Interaksi harus fleksibel tetapi juga diatur kedalam konteks aksi yang sedang berlangsung.
5. Non-aktifkan perintah yang tidak sesuai dalam konteks aksi yang sedang berlangsung.
6. Biarkan pengguna mengontrol aliran interaktif.
7. Sediakan bantuan untuk membantu semua aksi input.

Dalam melakukan input kedalam komputer dilakukan suatu pengkodean, karena akan mempermudah dalam proses pengolahan data, dengan adanya pengkodean maka data akan lebih mudah untuk didefenisikan dan pengkodean ini merupakan suatu bagian yang sangat diperlukan untuk menggabungkan suatu penjelasan.

Pengkodean yang sering digunakan dapat digolongkan menjadi beberapa macam adalah sebagai berikut (Jogiyanto, 2016:31) :

1. Kode numerik *(numeric code)* menggunakan 10 (sepuluh) macam kombinasi angka didalam kode.
2. Kode alphabet *(alphabhetic code)* merupakan 26 (dua puluh enam) huruf untuk kodenya.
3. Kode alphanumerik *(alphanumeric code)* merupakan kode yang menggunakan bangunan angka, huruf dan karakter-karakter khusus.
   * 1. **Perancangan Keluaran *(Output)***

Menurut Jogiyanto (2015:23), “Keluaran *(Output)* adalah produk dari sistem informasi yang dapat dilihat, dapat diubah tampilan layar dimonitor, hasil cetak pada kertas dan hasil proses yang akan digunakan proses lain dalam bentuk format file dalam media penyimpanan.Output adalah produk dari sistem informasi yang dapat dilihat yaitu berupa tampilan pada media atau layar monitor”.

Output dapat diwujudkan dalam beberapa bentuk, misalnya hardcopy dari laporan cetakan atau softcopy seperti layar komputer dan keluaran audio.

Beberapa tujuan dalam perancangan keluaran *(output)* adalah(Kendall & Kendall, 2017:46) :

1. Merancang output untuk disesuaikan dengan pengguna.
2. Mengirim jumlah output yang tepat.
3. Merancang output untuk tujuan-tujuan tertentu.
4. Menampilkan distribusi output yang tepat.
5. Memilih metode yang paling efektif.
6. Menampilkan output tepat waktu.

Desain *output*secara umum dapat dilakukan dengan langkah-langkah sebagai berikut (Hartono, 1999:78) :

1. Menentukan kebutuhan *output* dari system baru.
2. Menentukan parameter dari *output.*
3. Parameter ini meliputi tipe dari *output,* formatnya, media yang digunakan, alat *output* yang digunakan, jumlah tembusnya, distribusinya dan periode *output.*

Perancangan keluaran dapat digolongkan menjadi empat macam, (Jogiyanto, 2001:43) :

1. Tulisan, terdiri dari huruf, kata, angka, karakter khusus, dan simbol-simbol lain.
2. *Image,* berbentuk grafik atau gambar.
3. Suara, dalam bentuk musik.
4. Bentuk yang dapat dibaca oleh mesin *(Machine Readable Form)* dalam bentuk simbol yang hanya dapat dibaca dan dimengerti oleh komputer.

*Output* dari proses perancangan perangkat lunak adalah informasi yang dapat dibagikan kedalam tiga kategori besar, (Pressman, 2019:25) :

1. Program komputer (baik tingkat sumber maupun bentuk yang dapat dieksekusi).
2. Dokumen yang menggambarkan program komputer (ditargetkan untuk praktisi teknis dan pemakai).
3. Data yang diisikan dalam program atau dikeluarkan dari program.
   1. **Perancangan Perangkat Lunak *Website***

Pengertian *Website* adalah kumpulan dari halaman web yang sudah dipublikasikan di jaringan internet dan memiliki domain *URL* (*Uniform Resource Locator*) yang dapat diakses semua pengguna internet dengan cara mengetikan alamatnya.

Menurut Rohi Abdulloh (2018:1) “Website merupakan kumpulan halaman yang berisikan informasi data digital baik berupa teks, gambar, animasi, suara dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang di seluruh dunia”.

Menurut Zabar dan Novianto (2015:69), “Website atau situs merupakan sebuah kumpulan halaman-halaman web beserta file-file pendukngnya yang menampilkan informasi seperti file gambar, video dan file digital lainnya yang disimpan pada sebuah web server yang umumnya dapat diakses melalui internet”.

Jadi dapat diambil intisarinya bahwa website adalah kumpulan halaman yang terdiri dari beberapa laman yang menampilkan informasi seperti file gambar, video, teks, dan audio yang dapat diakses melalui koneksi internet.

Perancangan perangkat lunak ini menyediakan secara rinci terkait dengan arsitektur perangkat lunak, struktur data yang digunakan dalam perangkat lunak, antarmua, komponen dan subsistem yang diperlukan untuk implementasi perangkat lunak. Sasaran dari perancangan perangkat lunak yang memperlihatkan kekuatan, komoditi serta kenyamanan (Lila Setiyani, 2018:205).

Perancangan dalam penmbangunan perangkat lunak merupakan upaya untuk mengonstruksi sebuah sistem yang akan memberikan kepuasan akan spesifikasi kebutuhan fungsional, memenuhi target, memenuhi kebutuhan secara implisit atau eksplisit dari segi performa maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu, dan perangkat. Kualitas ini biasanya dinilai dari segi kepuasan pengguna terhadap perangkat yang digunakan (Rosa & Shalahuddin, 2018:23).

Proses perangkat lunak (*software process*) adalah sekumpulan aktifitas yang memiliki tujuan untuk mengembangkan atau mengubah suatu perangkat lunak. Secara umum proses perangkat lunak terdiri dari (Rosa & Shalahuddin, 2018:11) :

1. *Spesification*

Mengetahui apa saja yang harus dapat dikerjakan atau diperlukan oleh sistem perangkat lunak dan batasan pengembangan perangkat lunak.

1. *Development*

Merupakan pengembangan perangkat lunak untuk menghasilkan sistem perangkat lunak.

1. *Validation*

Memeriksa apakah perangkat lunak sudah dapat memenuhi keinginan pelanggan (*custome*).

1. *Evolution*

Mengubah perangkat lunak untuk memenuhi perubahan kebutuhan pelanggan (*customer*).

* 1. **Perancangan Basis Data (*Database*)**

Menurut Prehanto (2020:8-9), “Data merupakan kumpulan fakta atau sesuatu digunakan sebgaia input yang diolah dalam proses dan akan menghasilkan suatu informasi (*output*). Sebuah data harus sesuai kebenarannya (*reliable*), akurat, tepat waktu dan mencangkup ruang lingkup yang luas”.

Menurut Indrajani (2018:2), “Basis data merupakan komponen mendasar suatu system informasi, dimana pengembangan atau peggunaannya harus dilihat dari perspektif yang lebih luas berdasarkan kebutuhan organisasi. Dengan system informasi, memungkinkan terjadinya proses pengumpulan data, pengaturan, pengawasan, dan penyebaran informasi ke seluruh organsasi”.

Menurut Rosa & Shalahuddin (2018:43), “*System* basis data adalah *system* terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat”.

Menurut Dengen dan Hatta (2019:49), “Basis data dapat dibayangkan sebagai sebuah lemari arsip. Tetapi basis data terdiri atas dua kata yaitu basis dan data. basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek. Jadi basis data adalah gudang atau kumpulan dari data”.

Sedangkan menurut Jogiyanto (2015:17), “Basis Data adalah kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di komputer dan digunakan perangkat lunak untuk memanipulasikannya”.

Langkah-langkah untuk merancang database secara umum, adalah sebagai berikut(Jogiyanto, 2015:17) :

1. Menentukan kebutuhan file *database* untuk sistem baru.

File yang dibutuhkan dapat ditentukan dari diagram alir data (DAD) sistem baru yang telah dibuat.

1. Menentukan parameter dari file database

Setelah file-file yang dibutuhkan telah dapat ditentukan, maka parameter dari file selanjutnya juga dapat ditentukan, parameter ini meliputi :

1. Tipe dari file : file include, file transaksi, file semntara dan lain sebagainya.
2. Media file : harddisk, disket atau pita magnetik.
3. Organisasi dari file : apakah file tradisional (file unit, ISAM atau file akses langsung) atau organisasi database(struktur berjenjang, jaringan atau hubungan).
4. Field kunci dari file.

Ada beberapa hal yang menjadi tujuan digunakannya basis data, antara lain (Dengen dan Hatta, 2019:52) :

1. Salah satu komponen penting dalam sistem informasi, karena merupakan dasar dalammenyediakan informasi.
2. Menentukan kualitas informasi : data lebihrelevan. Informasi dapat dikatakan bernilaibila manfaatnya lebih efektif dibandingkandengan biaya mendapatkannya.
3. Mengurangi duplikasi data (*data redundancy*).
4. Hubungan data dapat ditingkatkan (*datarelatability*).
5. Mengurangi pemborosan tempat simpananluar (*space*).
6. Keamanan data dapat ditingkatkan (*security*).
7. Mempersingkat waktu pengolahan (tambah, ubah, hapus, dan cari) data (efisien).
8. Data lebih akurat sesuai dengan informasiyang dibutuhkan (*accurate*).
9. Pembatasan hak akses.

Pada basis data terdapat normalisasi, menurut Jogiyanto (2015:19), “normalisasi adalah proses untuk mengorganisasikan file untuk menghilangkan grup elemen yang berulang-ulang”.

Normalisasi adalah bagian perancangan basis data. Tanpa normalisasi, sistem basis data menjadi tidak akurat, lambat, tidak efisien, serta tidak memberikan data yang diharapkan. Pada waktu normalisasi basis data, ada empat tujuan yang harus dicapai yaitu(Simarmata dan Paryudi, 2010:31) :

1. Mengatur data dalam kelompok-kelompok sehingga masing-masing kelompok hanya menangani bagian kecil sistem.
2. Meminimalkan jumlah data berulang dalam basis data.
3. Membuat basis data yang datanya diakses dan dimanipulasi secara cepat dan efisien tanpa melupakan integrasi data.
4. Mengatur data sedemikian rupa sehingga ketika memodifikasi data, anda hanya mengubah pada satu tempat.

Berikut merupakan tahapan pada normalisasi tabel(Connoly dan Begg, 2010:98) :

1. *Unnormalizedform* merupakan suatu tabel yang berisikan informasi data mentah dari sumber informasi yang didapatkan. Tabel *unnormalized form* pada umumnya masih terdiri dari tampilan informasi secara berulang
2. *First Normal Form* (1NF) merupakan suatu relasi yang jika relasi tersebut tidak mengandung atribut yang bernilai ganda (*multivalued attribute*), atribut composite atau kombinasinya dalam domain data yang sama, harus bersifat atomik (tidak dapat terbagi), dan masing-masing *cells* pada tabel memiliki nilai. Akan tetapi, jika proses normalisasi hanya sampai pada tahapan 1NF, maka tabel tersebut masih belum bisa dikatakan normal dan efisien, sebab masih terdapat beberapa permasalahan, seperti : *Duplication, Inconsistency, Data Isolation, dan Un-Efficiency.*
3. *Second Normal Form*(2NF) merupakan tahapan dengan melakukan penghilangan perulangan (*redundant*) atribut dengan menentukan *primary key*. Bentuk Normal 2NF terpenuhi dalam sebuah tabel jika telah memenuhi bentuk 1NF, dan semua atribut selain *primary key* memiliki *functional dependency* pada *primary key*. Jika terdapat atribut yang tidak memiliki ketergantungan terhadap *primary key*, maka atribut tersebut harus dipindah atau dihilangkan.
4. *ThirdNormal Form* (3NF) adalah sebuah relasi dalam 1NF dan 2NF dan dimana tidak terdapat atribut *non-primary key* yang bersifat *transitively dependent* pada *primary key*. Bentuk normal 3NF terpenuhi jika telah memenuhi bentuk 2NF, dan jika tidak ada atribut *non primary key* yang memiliki ketergantungan terhadap atribut *non primary key* yang lainnya.
5. ***MySQL***

Menurut Nixon (2018:195), “Basis data adalah kumpulan terstruktur dari catatan atau data yang disimpan dalam sistem komputerdan diatur sedemikian rupa sehingga dapat dicari dengan cepat dan informasi dapat diperoleh cepat diambil”.

*MySQL* bekerja menggunakan *SQL* *Languange (Structure Query Languange),* yang dapat diartikan bahwa MySQL merupakan standar penggunaan database di dunia untuk pengolahan data. Kelebihan yang dimiliki *MySQL* yaitu bersifat *open source*, yang memiliki kemampuan untuk dikembangkan lagi. (Wahyudi, 2017).

Menurut Anissa (2016), “*MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *SQL* atau *DBMS* yang *multithread, multi-user”*.

Adapun kelebihan MySQL (Anissa, 2016) :

1. *Free* (bebas didownload)
2. Stabil dan tangguh
3. Fleksibel dengan berbagai pemrograman
4. *Security* yang baik
5. Dukungan dari banyak komunitas
6. Kemudahan management *database*.
7. Mendukung transaksi
8. Perkembangan *software* yang cukup cepat.
9. **Normalisasi**

Menurut Indrajani (2018:48), “Normalisasi adalah suatu Teknik formal yang dapat digunakan dalam perancangan basis data dengan peran penggunaan pendekatan *bottom-up* dan teknik validasi”.

Tujuan utama normalisasi adalah mengidentifikasi kesesuaian hubungan yang mendukung data untuk memenuhi kebutuhan perusahaan, dengan karakteristik sebagai berikut (Indrajani, 2018:47) :

1. Minimal jumlah atribut yang diperlukan untuk mendukung kebutuhan perusahaan.
2. Atribut dengan hubungan logika yang menjelaskan functional denpendencies.
3. Minimal duplikasi untuk tiap atribut.
   1. **Bahasa Pemrograman**
4. ***Hyper Text Markup Language (HTML)***

*HTML* bertanggung jawab mengatur tampilan halaman *website*. Sebutan lazim untuk *HTML* adalah bahasa markup (*Markup Language*) seperti yang adad di dalam singkatan *HTML* itu sendiri. Itu artinya, *HTML* adalah bahasa struktur untuk menandai bagian – bagian dari sebuah halaman (Jubile Enterprise, 2018:20).

Menurut Hidayatullah & Kawistara (2017:15), “*Hyper Text Markup Language (HTML)* adalah bahasa standard yang digunakan untuk menampilkan halaman web”.

*HTML* setidaknya memiliki struktur dasar yang terdiri dari (Enterprise, 2018:20-24) :

1. *Tag DTD* atau *DOCTYPE*

*DTD* sendiri adalah singkatan dari *Document* *Type* *Declaration* yang berfungsi untuk memberi tahu browser bahwa dokumen yang akan ditampilkan adalah dokumen berjenis *HTML*.

1. *Tag* *HTML*

*Tag* ini merupakan tag pembuka dari seluruh halaman web yang akan dibuat. Selanjutnya, semua kode *HTML* yang membentuk desain sebuah halaman website harus berada di dalam *tag <html>* ini*.*

1. *Tag HEAD*

*Tag* yang ditulis dengan bentuk *<head>* ini berfungsi secara teknis. Bagian atau elemen yang ditulis pada *tag <head>* umumnya berbentuk keterangan teknis, seperti definisi dan judul halaman, kode *CSS*, *Javascript*, deskripsi halaman, dan kode – kode lainnya yang tidak terlihat sebagai salah satu desain di dalam sebuah halaman.

1. *Tag BODY*

*Tag <body>* digunakan untuk meletakkan semua elemen yang akan tterlihat di dalam halaman website pada saat halaman itu diakses oleh *browser*. Sama seperti dokumen biasa yang mengandung beberapa elemen, seperti paragraf, tabel, link, gambar, dan sebagainya.

Menurut Junaedi (2005:19), “HTML (Hyper Text Markup Language) adalah suatu format data yang digunakan untuk membuat dokumen hypertext yang dapat dieksekusi dari satu platform komputer ke platform komputer lainya tanpa perlu melakukan suatu perubahan apapun dengan suatu alat tertentu”.

Pengertian lainnya menurut Asropudin (2013:44), “HTML atau *Hyper Text Markup Language* adalah bahasa komputer yang digunakan untuk membuat sebuah halaman webpage”. Menurut Badiyanto (2013:23), “HTML (Hyper Text Markup Language) adalah bahasa standar pemrograman untuk membuat halaman web yang terdiri dari kode-kode tag tertentu, kemudian kode-kode tersebut diterjemahkan oleh web browser untuk menampilkan halaman web yang terdiri dari beberapa macam format tampilan seperti teks, grafik, animasi link, maupun audio-video”.

Sedangkan menurut Kustiyahningsih dan Anamisa (2011:35), “HTML adalah file text murni yang dapat dibuat dengan editor text sembarang. Dokumen ini dikenal sebagai web page. File-file HTML ini berisi instruksi-instruksi yang kemudian diterjemahkan oleh browser yang ada dikomputer client (user) sehingga isi formasinya dapat ditampilkan secara visual dikomputer pengguna (user).”

1. ***PHP Hypertext Preprocessor (PHP)***

Menurut Jubile Enterprise (2018:2), “*PHP* merupakan bahasa pemrograman yang digunakan untuk membuat website dinamis dan interaktif. Dinamis artinya, website tersebut bisa berubah – ubah tampilan dan kontennya sesuai kondisi tertentu”.

Menurut Hidayatullah & Kawistara (2017:223), “*PHP Hypertext Preprocessor* atau disingkat dengan *PHP* ini adalah suatu bahasa *scripting* khususnya digunakan untuk web development. Karena sifatnya yang *serverside* *scripting*, maka untuk menjalankan *PHP* harus menggunakan web *server*. *PHP* juga dapat diintegrasikan dengan *HTML*, *i, JQuery, Ajax.* Namun, pada umumnya *PHP* lebih banyak digunakan bersamaan file bertipe *HTML*. Dengan menggunakan *PHP* Anda bisa membuat *website* *powerful* yang dinamis dengan disertai manajemen database-nya”.

Adapun kelebihan *PHP* (Hidayatullah & Kawistara, 2017:226-228) :

1. *PHP* merupakan sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya. Tidak seperti halnya bahasa pemrograman aplikasi yang lainnya.
2. *PHP* dapat berjalan pada web server yang dirilis oleh *Microsoft,* seperti *IIS* atau *PWS* juga pada *apache* yang bersifat *open source*.
3. Karena sifatnya yang open source, maka perubahan dan perkembangan interpreter pada *PHP* lebih cepat dan mudah, karena banyak milis-milis dan *developer* yang siap membantu pengembanganya.
4. Jika dilihat dari segi pemahaman, *PHP* memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami.
5. *PHP* dapat berjalan pada 3 (tiga) operating sistem, yaitu: *Linux, unux,* dan *windows*, dan juga dapat dijalankan secara *runtime* pada suatu *console*.
6. *PHP* bersifat bebas dipakai (*free*).
   1. ***Unfield Modelling Language (UML)***

Menurut Hendini (2016:8), “*Unified modeling language (UML)* merupakan metodologi yang digunakan untuk mengembangkan sistem berorientasi objek dan juga sebagai alat bantu dalam pengembangan sistem”.

Menurut Ibnu Akil (2018:4-5), “*Unified modeling language (UML)* adalah bahasa pemodelan visual yang digunakan untuk menspisifikasikan, memvisualisasikan, membangun, dan mendokumentasikan rancangan dari suatu *system* perangkat lunak. Pemodelan memberikan gambaran yang jelas mengenai system yang akan dibangun baik dari sisi *structural* ataupun fungsional”.

Menurut Rosa & Shalahuddin (2015:137), “*UML* merupakan bahasa *visual* untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”.

*UML* dapat diterapkan pada semua model pengembangan, tingkatan siklus system, dan berbagai macam domain aplikasi. Dalam *UML* terdapat konsep *semantic*, notasi, dan panduan masing-masing diagram. *UML* juga memiliki bagian statis, dinamis, ruang lingkup, dan organisasional (Ibnu Akil, 2018:5).

Menurut Pratama (2010:6), “UML (Unified Modeling Language) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek”.

Sedangkan Menurut Booch, dkk (2005:7), ”UML adalah Bahasa standar untuk membuat rancangan software. UML biasanya digunakan untuk menggambarkan dan membangun, dokumen artifak dari software –intensive system”

Berdasarkan pendapat beberapa ahli di atas, dapat disimpulkan bahwa *unified modeling language* (UML) merupakan bahasa visual untuk pendokumentasian dan pemodelan rancangan sistem dimana hasil visualisasi ini dapat dijadikan sebagai alat bantu dalam pengembangan sistem.

1. ***Activity Diagram***

Menurut Ibnu Akil (2018:14), “*Activity Diagram* menggambarkan arus dari suatu aktivitas ke aktivitas lainnya dari suatu fungsionalsistem”.

Menurut Rosa dan Shalahuddin (2018:161), “Diagram aktivitas atau *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak, serta menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem”.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut (Rosa dan Shalahuddin, 2018:161-162) :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan

merupakan proses bisnis sistem yang didefinisikan.

1. Urutan atau pengelompokan tampilan dari sistem (*user interface)* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
2. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
3. Rancangan menu yang ditampilkan pada perangkat lunak.

Secara umum tujuan dari *activity diagram* bisa digambarkan sebagai berikut (Munawar, 2018:128) :

* 1. Menggambarkan aliran aktifitas dari system
  2. Menggambarkan urutan satu aktifitas ke aktifitas lainnya
  3. Menggambarkan paralelisme dan aliran konkuren dari sistem

Menurut Hendini (2016:78) “*Activity diagram* merupakan diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.”

Menurut Meilinda (2016:34) “*Activity diagram* merupakan diagram yang menerangkan tentang aktifitas-aktifitas yang dapat dilakukan oleh seorang aktor atau pengguna terhadap aplikasi yang dirancang.”

Berdasarkan kutipan dari para ahli di atas, dapat disimpulkan bahwa *activity diagram* merupakan diagram yang menjelaskan tentang *work flow* (aliran kerja) atau aktifitas aktor terhadap aplikasi yang dirancang.

1. ***Use Case Diagram***

Menurut Tohari dalam Tabrani dan Aghniya (2019:46), “menyimpulkan bahwa, use case adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor”.

Menurut Rosa dan Shalahuddin (2018:155), “*Use Case* atau diagram *Use Case* merupakan pemodelan untuk kelakuan (*behavior*) system informasi yang akan dibuat. *Use case* mendiskripsikan sebuah interaksi antara satu atau lebih actor dengan system informasi yang akan dibuat”.

Ada dua hal utama pada *use case* yaitu pendifisian apa yang disebut aktor dan *use case* (Rosa dan Shalahuddin, 2018:155)

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antar unit atau aktor.
3. ***Class Diagram***

Menurut Munawar (2018:101), “*Class Diagram* adalah diagram statis. Ini mewakili pandangan statis dari suatu aplikasi. *Class diagram* tidak hanya digunakan untuk memvisualisasikan, menggambarkan dan mendokumentasikan berbagai aspek system tetapi juga untuk membangun kode eksekusi (*executable code*) dari aplikasi perangkat lunak”.

Menurut Munawar (2018:101), “*Class* diagram menggambarkan atribut, operation dan juga *constraint* yang terjadi pada system. *Class* diagram menunjukkan koleksi *Class*, antarmuka, asosiasi, kolaborasi, dan *constraint”*.

Tujuan dari class diagram adalah untuk memodelkan pandangan statis suau aplikasi. Secara lebih rinci, tujuan dari *class* diagram dapat diringkas sebaai berikut (Munawar, 2018:102) :

* 1. Analisis dan desain pandangan statis aplikasi.
  2. Mejelaskan tangggunga jawab suatu system.
  3. Basis untuk diagram komponen dan penyebaran (deployment).
  4. Forward and reverse engineering.

Menurut Rosa & Shalahuddin (2015:141) menyatakan bahwa “Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segini pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

1. ***Sequence Diagram***

Menurut Ibnu Akil (2018:14), “*Sequence Diagram* menggambarkan

urutan penyampaian pesan atau pemangilan metode antar objek dalam suatu event atau scenario”.

Menurut Rosa & Shalahuddin (2018:165), “*Diagram Sequence*menggambarkan kelakuan objek pada *use case* dengan mendsikripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Jadi, untuk menggambar diagram sekuen maka harus diketahui objek – objek yang terlibat dalam sebuah *use case* beserta metode – metode yang dimiliki kelas yang diinstasiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat scenario yang ada pada *use case”*.

Menurut Hendini (2016:65) “*Sequence diagram* berfungsi sebagai alat untuk menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.”

Menurut Meilinda (2016:21) “*Sequence diagram* merupakan UML yang menggambarkan interaksi antar objek di dalam dan disekitar sistem, termasuk pengguna, *display*, dan sebagainya berupa *message* yang digambarkan terhadap waktu.”

Berdasarkan pernyataan para ahli di atas, dapat disimpulkan bahwa *sequence diagram* merupakan diagram UML yang berfungsi sebagai alat untuk menggambarkan kelakukan objek pada *use case* dengan cara mendeskripsikan waktu hidup objek dan pesan yang terkirim dan diterima.

Berdasarkan definisi dari beberapa ahli, maka dapat disimpulkan bahwa *class diagram* merupakan diagram yang menjelaskan hubungan antar kelas, struktur definisi, operator dan modul yang digunakan untuk membangun aplikasi.

* 1. **Pengujian Perangkat Lunak**

Menurut Cholifah dkk. (2018:2), “Pengujian adalah satu set aktifitas yang

direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang

diinginkan. Aktifitas pengujian terdiri dari satu set atau sekumpulan langkah dimana dapat menempatkan desain kasus uji yang spesifik”.

Menurut Pressman (2016:459), “*Black-box* testing berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan engineer untuk memperoleh set kondisi input yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program”.

Keuntungan penggunaan metode *Blackbox Tetsting* adalah (Ammann,dkk, 2015:46) :

* 1. Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu.

1. Pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan.
2. Programmer dan tester keduanya saling bergantung satu sama lain.

Kekurangan dari metode *Black-box Testing* adalah (Ammann, 2015:46) :

* 1. Uji kasus sulit disain tanpa spesifikasi yang jelas.
  2. Kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh programmer.
  3. Beberapa bagian back end tidak diuji sama sekali.