

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Pada penelitian ini peneliti membuat judul penelitian dengan nama “*UX Orbit Design : Platform Kursus Online Bidang User Interface (UI) dan User Experience (UX)*”. Penelitian terdahulu yang peneliti cantumkan di bawah ini adalah sebagai acuan sehingga dapat memperkaya teori peneliti dalam mengkaji penelitian yang dilakukan. Berikut ini adalah penelitian yang peneliti kaji sebelumnya.

Penelitian yang dilakukan oleh Fathur Rahman dan Silvia Ratna dengan judul ‘Perancangan *E-learning* Berbasis *Web* Menggunakan *Framework Codeigniter*’ bertujuan merancang *e-learning* yang berbasis *web* untuk sekolah formal atau akademik. Sistem *e-learning* kadang-kadang berjalan tidak baik bahkan masih dilakukan secara manual sehingga sulit untuk mendapatkan hasil yang cepat dan tepat dalam waktu yang singkat. Untuk mengatasi permasalahan tersebut dapat dengan mudah dengan adanya sistem, guru yang terfasilitasi dan dapat mengontrol kegiatan pembelajaran peserta dengan mudah, kemudian interaksi antara pengajar dan peserta *e-learning* lebih menarik dan menyenangkan [6].

Selanjutnya, penelitian yang dilakukan oleh Abdullah Muammar dengan judul ‘Pengembangan *E-learning* Berbasis *Web* di Jurusan Pendidikan Biologi UIN Raden Intan Lampung’ bertujuan untuk membangun sistem yang berkonsep *e-learning* berbasis *web* pada jurusan pendidikan biologi. Permasalahannya belum

efektifnya penyaluran informasi, proses pembelajaran yang kurang menarik dan belum adanya pengembangan *e-learning* berbasis *web* untuk meningkatkan mutu pendidikan jurusan pendidikan biologi UIN Raden Intan Lampung. Pengembangan *website e-learning* ini bisa dikatakan layak dari berbagai penilaian yaitu terdiri dari ahli desain, ahli IT, ahli teknologi pendidikan, praktisi pembelajaran dan mahasiswa [7].

Kemudian, penelitian yang dilakukan oleh Angga Kurnia Putra, Rudy Dwi Nyoto dan Helen Sasty Pratiwi dengan judul ‘Rancang Bangun Aplikasi *Marketplace* Penyedia Jasa Les *Private* di Kota Pontianak Berbasis *Web*’ bertujuan untuk merancang dan membangun *marketplace* penyedia jasa yang berbasis *web*. Di Kota Pontianak pencarian calon guru les *private* yang memakan waktu cukup lama, sulit menemukan kesesuaian lokasi dan waktu antara calon guru les dan murid, tidak dapat mengetahui kompetensi, prestasi, dan *track-record* dari calon guru les *private* terbaik. Aplikasi *marketplace* jasa les *private* di Kota Pontianak dapat menjadi media dalam mempromosikan jasa les *private* bagi guru les *private* [8].

Dari berbagai jurnal penelitian yang sudah peneliti kaji, beberapa penelitian tentang media pembelajaran dan kursus *online* memberikan solusi dan menawarkan sistem secara *online* yang memanfaatkan media internet sebagai solusi utama. Selain dengan kecepatan media informasi era saat ini tentu para anggota atau siswa lebih nyaman dan lebih menyukai menggunakan sistem *online* karena bisa lebih interaktif dan menarik.

Dari beberapa penelitian tersebut solusi yang ditawarkan itu cukup bagus, tetapi belum adanya solusi yang berfokus untuk memecahkan masalah terkait pembelajaran non-akademik yaitu untuk masyarakat yang ingin meningkatkan pengetahuan dan kemampuan diri pada bidang UI/UX. Maka dari itu dalam penelitian ini peneliti memberikan solusi yang terbaru yaitu solusinya sebagai berikut.

Setelah melihat kelebihan dan kekurangan dari penelitian sebelumnya, maka peneliti berusaha untuk menggali dan terus mencari guna memberikan solusi yang terbaik yang bersumber dari penelitian sebelumnya. Maka dari itu peneliti memberikan sebuah solusi yaitu membuat sebuah *platform* kursus *online* khusus untuk meningkatkan pengetahuan dan kemampuan dalam bidang *User Interface* (UI) dan *User Experience* (UX) agar masyarakat bisa lebih menjurus dalam satu bidang dan lebih fokus pada bidang tersebut sehingga masyarakat bisa tahu bidang mana yang ingin mereka geluti. Kursus *online* ini tentunya juga menyediakan fasilitas pembelajaran dengan menggunakan *video* agar lebih menarik.

2.2 Teori-teori Dasar

Berikut ini adalah istilah-istilah atau teori-teori dasar dalam penelitian ini yang tentunya harus peneliti ketahui dan mengerti guna untuk melakukan penelitian dengan baik.

2.2.1 Platform

Platform adalah suatu kombinasi antara perangkat lunak (*software*) dengan perangkat keras (*hardware*) yang bertujuan untuk menjalankan sebuah program tertentu. Singkatnya adalah *platform* ini bisa disebut rencana kerja atau program

yang akan menjadi dasar berjalannya sebuah sistem. Juga bisa disebut dengan tempat berjalannya sebuah perangkat lunak [9].

Dalam penelitian ini kenapa peneliti menggunakan *platform* sebagai medianya, karena peneliti bermaksud untuk membuat sebuah *platform* digital yang terhubung melalui teknologi *internet*. Membangun sebuah *platform* yang bernama “UX Orbit Design” ini peneliti juga mencoba memanfaatkan teknologi yang ada dan mencoba mengkombinasikan antara perangkat keras dan perangkat lunak guna mencapai kebutuhan manusia.

Perangkat keras yang digunakan oleh pengguna dari *platform* yang sedang peneliti bangun nantinya adalah bisa komputer, bisa *tablet* dan juga *handphone* tergantung dari penggunanya.

Sedangkan perangkat lunak yang digunakan oleh pengguna adalah *browser* yang bisa menjalankan aplikasi berbasis *web*. *Browser* ini kebanyakan sudah terpasang juga di berbagai macam komputer, *tablet* dan *handphone*.

Platform ini bisa memfasilitasi masyarakat dan menghubungkan masyarakat dengan mentor untuk memilih berbagai macam kelas kursus *online*, melakukan pembelian kelas, melakukan pembelajaran dengan fasilitas video pembelajaran, menyediakan fasilitas diskusi bila ada yang hal mau didiskusikan, mendapatkan sertifikat, dan melakukan *submission*.

2.2.2 User Interface (UI) dan User Experience (UX)

User Interface (UI) dan *User Experience* (UI) keduanya adalah dua hal yang berbeda. UI/UX ini berkaitan erat dengan konsep *Human Computer Interaction* (*HCI*) dimana *device*, komputer dan *smartphone* yang sekarang kita pegang adalah

bagian dari HCI. Ketika kita menekan *keyboard*, menggeser *mouse*, menggeser layar *smartphone*, *gesture* sehingga kita dapat mendapatkan kontrol yang penuh terhadap *device* tersebut itulah yang dinamakan interaksi.

UI/UX juga bekerja dengan interaksi yaitu interaksi antara pengguna dengan *device*. Sehingga bagaimana cara kita mendesain sebuah aplikasi, *web* dan atau *platform* lain yang memang bisa digunakan oleh manusia dari desain antarmuka yang disuguhkan.

User Interface (UI) dalam bahasa Indonesianya adalah desain antarmuka pengguna, desain antarmuka disini yaitu baik itu untuk mesin dan perangkat lunak seperti komputer, peralatan rumah tangga, perangkat *mobile* dan perangkat elektronik lainnya. Mudahnya adalah *user interface (UI)* itu suatu tampilan *website* atau aplikasi yang kita buat agar terlihat oleh pengguna. Apa yang bisa dilihat oleh mata (*visible*) oleh pengguna itu adalah UI. UI berfokus pada bagaimana membuat visualisasi, warna dan hal-hal lain yang berkaitan dengan kreativitas dari *interface* yang akan digunakan oleh pengguna. Visualisasi ini adalah kunci utama dari UI agar tampilan yang kita buat dan dipakai oleh pengguna bisa menjadi menarik [10].

User Experience (UX) adalah proses bagaimana membuat suatu produk terkhusus produk digital menjadi mudah untuk digunakan dan tentunya tidak membingungkan pengguna ketika menggunakan produk tersebut. UX ini memiliki ranah yang begitu luas, dimulai dari riset produk untuk mengetahui siapa penggunanya bagaimana *behaviour* pengguna yang nantinya digunakan untuk membuat suatu tampilan desain antarmuka aplikasi atau *website* yang sesuai dengan

kebutuhan pengguna dan keinginan pengguna. UX ini berkaitan dengan bagaimana memecahkan permasalahan pengguna yang lebih spesifik [10].

User Interface (UI) dan *User Experience (UX)* ini adalah memang dua hal yang jauh berbeda tetapi memiliki keterhubungan yang sangat dekat. Tugas dari seorang *UI/UX Designer* adalah bukan hanya merancang dan mendesain antarmuka aplikasi yang cantik tetapi mendesain bagaimana membuat produk kita bisa digunakan oleh pengguna dengan mudah dan dapat menyelesaikan permasalahan yang pengguna hadapi.

2.2.3 Kursus *Online*

Kursus *online* adalah sebuah program belajar yang dilakukan secara *online* dengan menggunakan perangkat digital seperti komputer atau *smartphone* yang bertujuan untuk meningkatkan pengetahuan dan kemampuan seseorang pada bidang tertentu [11].

Kursus *online* pada media sekarang digunakan menggunakan *video* sehingga jadi lebih menarik dan diyakini dapat lebih mudah memahami pembelajaran. Kelebihan dari kursus *online* adalah bisa diakses kapan saja dan dimana saja. Selain bertujuan untuk meningkatkan pengetahuan dan kemampuan dengan lebih mudah, pembelajaran dengan menggunakan teknologi saat ini bisa meningkatkan juga kualitas pelayanan yang dilakukan para guru atau dosen, yang dimana mereka akan sangat terbantu oleh hadirnya sebuah teknologi pada bidang pendidikan terutama pengembangan kemampuan individu terhadap dunia kerja [12].

Perbedaan antara kursus *online* dengan *e-learning* adalah kursus *online* itu sebetulnya bagian dari *e-learning* sehingga bisa dikatakan *e-learning* mencakup segala hal terkait sistem pembelajaran secara elektronik atau *online*. Kursus *online* adalah pembelajaran *online* atau pembelajaran yang menggunakan internet sebagai medianya. Sedangkan *e-learning* adalah pembelajaran melalui media perangkat elektronik [13].

Adapun kursus *online* atau pembelajaran secara daring tidak lebih dari bertujuan untuk mengembangkan pengetahuan dan kemampuan baik *soft skill* maupun *technical skill* atau *hard skill*.

Soft skill adalah keterampilan yang berkaitan dengan mengelola diri dan orang lain. Keterampilan tersebut bisa dengan mudah kita pelajari melalui belajar kursus *online*. Selain keterampilan tersebut, tentunya harus dibarengi dengan keterampilan *hard skill* atau bisa disebut dengan kemampuan teknis dalam suatu bidang tertentu [14].

2.2.4 Metode Pengembangan *Prototype*

Metode pengembangan perangkat lunak yang peneliti gunakan adalah *prototype*. *Prototype* ini metode pengembangan perangkat lunak yang berfokus untuk menghubungkan pemahaman kebutuhan pengguna dengan pengembang sehingga persepsi antara pengguna dan pengembang itu menjadi sama. Untuk dapat berinteraksi dalam proses kegiatan pengembangan maka metode *prototype* ini akan menghasilkan sebuah *prototype* sistem yang akan dibuat sebagai tahap awal dari sistem [15].

Tujuan peneliti menggunakan *prototype* ini agar pengembang sistem bisa mengumpulkan informasi dari pengguna terlebih dahulu sehingga dapat dengan mudah menggambarkan model interaksi yang akan dikembangkan.

Untuk menjalankan metode pengembangan *prototype* ini ada beberapa langkah-langkah yang dilakukan sebagai berikut :

1. Identifikasi Kebutuhan

Dalam mengumpulkan kebutuhan akan melibatkan pertemuan antara pengembang dan pengguna untuk menentukan apa yang menjadi tujuan dan kebutuhan dibuatnya perangkat lunak atau sistem ini.

2. Membuat Prototipe

Desain yang cepat ini mengarah ke pembangunan prototipe yang akan dihasilkan mencakup seperti apa *input*-nya, proses dan format *output*-nya.

3. Menguji prototipe

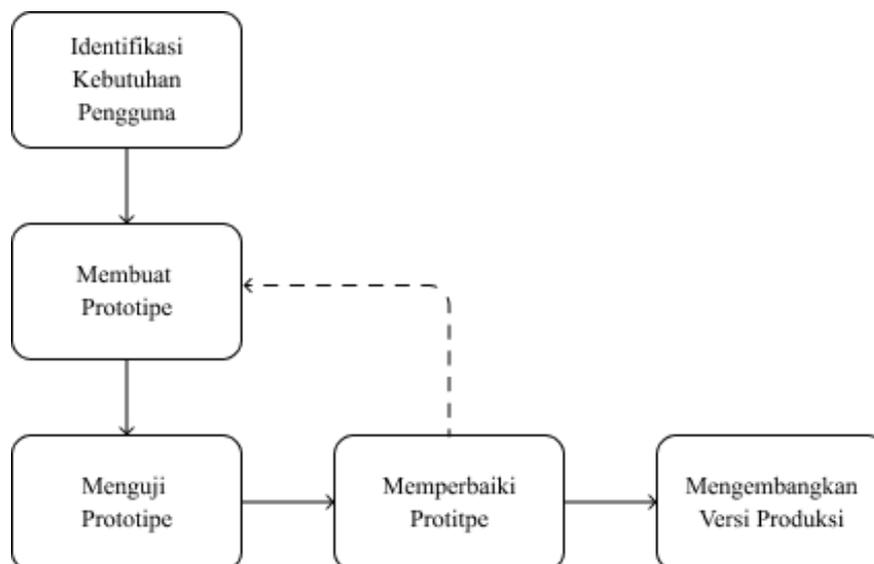
Dalam langkah ini prototipe yang sudah dibangun akan dievaluasi bersama dengan pengguna dan pengembang untuk menyesuaikan kebutuhan perangkat lunak yang akan dikembangkan. Pada saat ini pengguna dan pengembang akan lebih jelas dan detail memahami apa yang perlu dilakukannya.

4. Memperbaiki Prototipe

Setelah beberapa tahapan sudah dilakukan maka saatnya untuk membangun perangkat lunak dan kemudian melakukan pengujian dan perbaikan apabila masih ada perubahan dalam sistem yang sedang dibangun.

5. Mengembangkan Versi Prototipe

Tahapan terakhir adalah mengembangkan versi prototipe yang bertujuan untuk mengembangkan versi berikutnya sehingga menjadi produk nyata yang digunakan oleh pengguna.



Gambar 2. 1 Langkah-langkah *Prototype*

2.2.5 *Unified Modeling Language (UML)*

UML ini adalah sebuah bahasa pemodelan sistem ataupun pemodelan di bidang lain yang membutuhkan sebuah pemodelan. Pemodelan disini menggunakan model dalam bentuk grafis yang bertujuan untuk menspesifikasi, memvisualisasi dan membangun atau mendokumentasikan sebuah sistem agar mudah dipahami oleh manusia atau mesin. UML ini menggunakan konsep berorientasi *object*

bertujuan untuk memudahkan manusia atau pengembang memahami sistem yang akan dibuat dalam bentuk grafis [16].

UML ini memiliki beberapa *diagram* yang digunakan untuk melakukan pemodelan data ataupun sistem. Berikut ini adalah *diagramnya* :

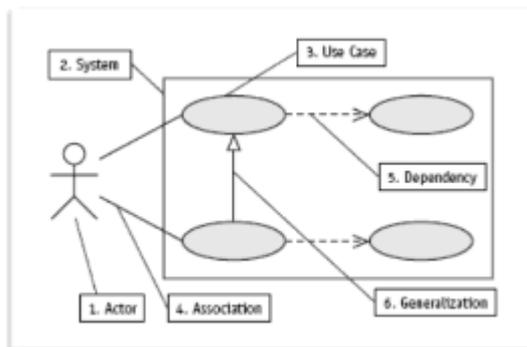
1. *Use Case Diagram*

Use Case Diagram adalah salah satu *diagram* UML yang bertujuan untuk menggambarkan hubungan antara manusia atau sistem lain dengan sistem yang mau dirancang. Manusia atau sistem lain disebut dengan aktor. Selain menggambarkan hubungan antara aktor dengan sistem, *use case* ini adalah menggambarkan sebuah fungsi dari sistem. Mudah-mudahan adalah apa yang dilakukan oleh sistem bukan bagaimana yang dilakukan oleh sistem. Contohnya adalah aktor membuat daftar belanja dan sistem menjalankan fungsi buat daftar belanja [17].

Di dalam *use case* ada beberapa elemen yang digunakan yaitu seperti yang sudah disinggung di atas yaitu aktor, *use case*, sistem, *association*, *dependency*, dan *generalization*.

- a. Aktor adalah manusia atau sistem lain yang berinteraksi dengan sistem.
- b. *Use case* adalah fungsionalitas yang ada dalam sebuah sistem.
- c. Sistem adalah proses yang sudah dibatasi sesuai dengan apa yang kita buat.
- d. *Association* adalah sebuah garis yang menggambarkan interaksi atau hubungan antara aktor dengan sistem.

- e. *Dependency* adalah sebuah garis yang menggambarkan interaksi atau hubungan antara dua *use case*. Interaksinya ada dua macam yaitu *include* dan *extends*. *Include* adalah sebuah garis yang dimana menghubungkan satu *use case* membutuhkan *use case* lain, sedangkan *extends* adalah sebuah garis yang dimana menghubungkan satu *use case* ketika dibutuhkan atau tergantung dengan keadaan *use case* tersebut.
- f. *Generalization* adalah konsep pewarisan yang dimana aktor atau *use case* bisa mewariskan kepada aktor dan *use case* itu sendiri atau aktor ke *use case* dan *use case* ke aktor.



Gambar 2. 2 Elemen dari Use Case Diagram
 (Sumber : Analisis dan Perancangan Sistem Informasi
 Manajemen Keuangan [12, p.43])

2. Use case Scenario

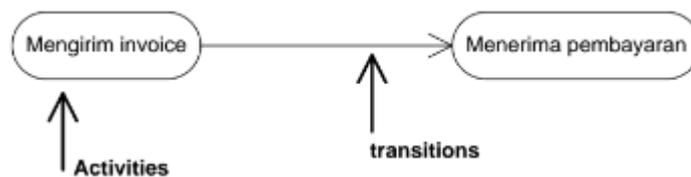
Use case scenario adalah suatu gambaran model yang berbentuk skenario sehingga nantinya ada beberapa kemungkinan-kemungkinan kejadian dalam menjalankan sebuah sistem tersebut. Kemungkinan-kemungkinan tersebut nantinya akan dipecah menjadi beberapa skenario. Di dalam proses sebuah sistem nantinya akan ada sekumpulan proses, proses inilah yang

akan menjadi beberapa kemungkinan-kemungkinan yang kemudian bisa disebut dengan skenario *use case*.

3. *Activity Diagram*

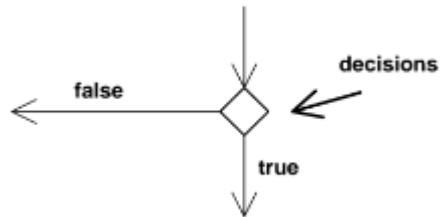
Activity diagram adalah *diagram* yang menggambarkan suatu aktivitas proses yang dilakukan dalam suatu sistem yang tentunya menghubungkan antara aktor dengan sistem. Di dalam *activity diagram* ada beberapa elemen yang digunakan yaitu berikut ini.

- a. *Activities* adalah elemen yang menggambarkan suatu aktivitas tertentu pada suatu sistem.
- b. *Transition* adalah elemen yang digunakan untuk melakukan transisi atau peralihan dari aktor ke sistem atau dari sistem ke aktor dan atau antar elemen.



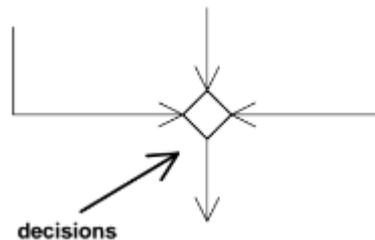
Gambar 2. 3 Gambaran *Transition*
 (Sumber : Analisis dan Perancangan Sistem Informasi
 Manajemen Keuangan [12, p.55])

- c. *Decision* adalah suatu elemen untuk melakukan percabangan logika di dalam suatu sistem. Sama seperti halnya di dalam *flowchart* ada juga yang dinamakan *decision*. Berikut ini gambarannya.



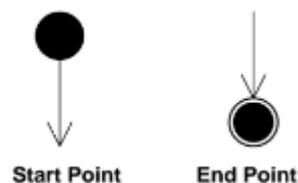
Gambar 2. 4 *Gambaran Decision*
 (Sumber : Analisis dan Perancangan Sistem Informasi
 Manajemen Keuangan [12, p.56])

- d. *Merge Point* adalah elemen yang digunakan untuk menggabungkan suatu proses di dalam sistem. Kalo diatas tadi *decision* itu untuk melakukan percabangan sedangkan *merge point* ini untuk melakukan penggabungan dari hasil percabangan tersebut.



Gambar 2. 5 *Gambaran Merge Point*
 (Sumber : Analisis dan Perancangan Sistem Informasi
 Manajemen Keuangan [12, p.56])

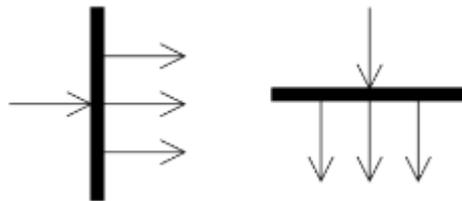
- e. *Start Point and End Point* adalah elemen untuk menandakan dimulainya suatu aktivitas dan berakhirnya aktivitas dalam suatu sistem.



Gambar 2. 6 *Gambaran Start Point dan End Point*

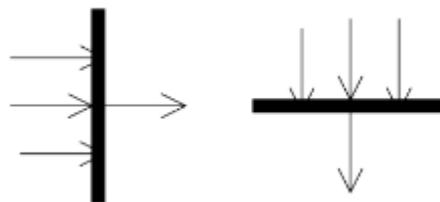
(Sumber : Analisis dan Perancangan Sistem Informasi Manajemen Keuangan [12, p.57])

- f. *Concurrency* adalah elemen untuk melakukan percabangan proses dalam suatu sistem. Berbeda halnya dengan *decision*, *decision* untuk melakukan percabangan logika tetapi kalau ini untuk melakukan percabangan proses.



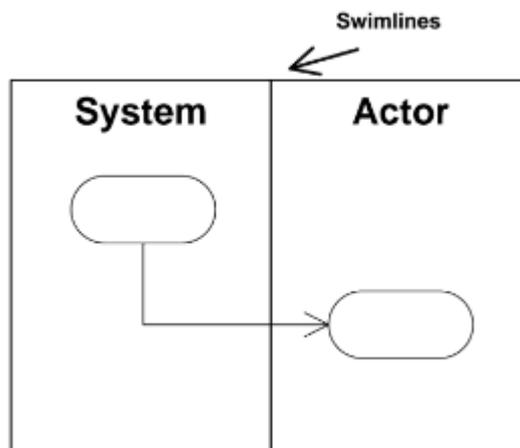
Gambar 2. 7 Gambaran *Concurrency*
(Sumber : Analisis dan Perancangan Sistem Informasi Manajemen Keuangan [12, p.57])

- g. *Synchronization* adalah elemen untuk melakukan menggabungkan hasil proses yang dipisahkan oleh *concurrency*.



Gambar 2. 8 Gambaran *Synchronization*
(Sumber : Analisis dan Perancangan Sistem Informasi Manajemen Keuangan [12, p.58])

- h. *Swimlanes* adalah elemen digunakan untuk memisahkan aktor dengan sistem, aktor dengan aktor yang lain atau sistem dengan sistem yang lain.



Gambar 2. 9 Gambaran *Swimlanes*
 (Sumber : Analisis dan Perancangan Sistem Informasi
 Manajemen Keuangan [12, p.58])

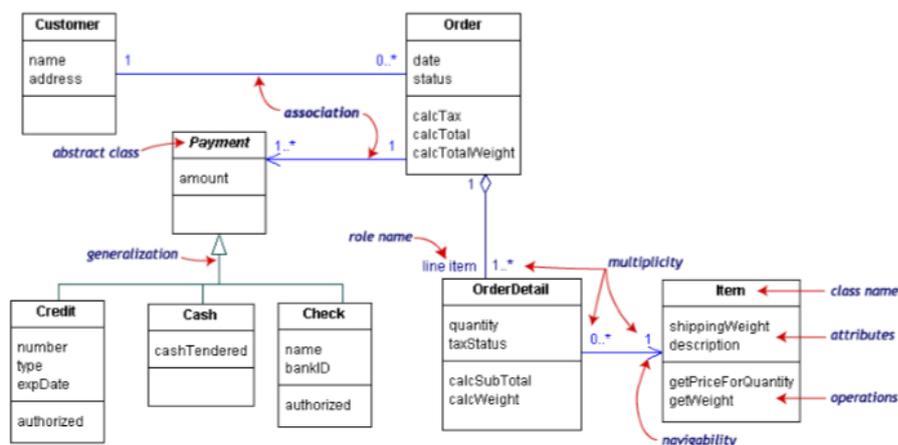
4. *Class Diagram*

Class diagram adalah *diagram* yang menggambarkan struktur sistem secara spesifik yang didalamnya ada nama kelas, atribut dan metode. *Class diagram* ini jika diinstansiasi akan menghasilkan berupa objek karena berhubungan dengan konsep berorientasi objek.

Area pokok yang ada di dalam *class diagram* ada tiga yaitu nama kelas, atribut dan metoda. Atribut dan metoda ini memiliki beberapa sifat yaitu *private* atribut atau metode tidak dapat dipanggil dari luar *class* dapat dipanggil di dalam *class* yang sama. *Protected* yaitu ketika atribut dan metode hanya bisa dipanggil di dalam *class* yang sama dan bisa mewariskan ke anak-anaknya, dan ketika atribut dan metode memiliki sifat *public* maka semua atribut dan metode baik yang ada di luar kelas atau di dalam *class* yang sama bisa dipanggil kapanpun dan oleh siapa saja.

Hubungan yang digunakan untuk menggambarkan *class diagram* adalah sebagai berikut :

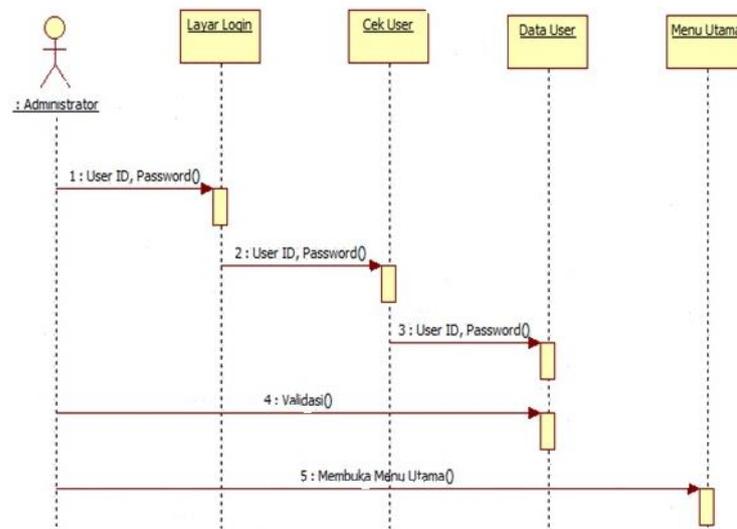
- Asosiasi adalah garis penghubung yang menghubungkan antar *class*. *Class* disini adalah *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain.
- Agregasi adalah garis di dalam *class diagram* yang menghubungkan pernyataan bahwa *class* menjadi atribut di *class* yang lain.
- Pewarisan adalah *class* dapat diturunkan dari *class* lain dan bisa mewarisi semua atribut dan metoda *class* lain sehingga bisa menambahkan fungsionalitas baru menjadi disebut anak dari *class* yang diwarisinya.
- Hubungan dinamis adalah suatu rangkaian pesan yang dikirim dari *class* pada *class* lain.



Gambar 2. 10 Contoh Class Diagram
(Sumber : Pengantar *Unified Modeling Language (UML)* [13, p.6])

5. Sequence Diagram

Adalah *diagram* yang menggambarkan interaksi antar objek yang ada di dalam dan diluar sistem sehingga menjadi dinamis. Kegunaan *diagram sequence* ini adalah untuk menunjukan rangkai pesan yang dikirim antar *object* atau antar sistem. Pesan ini digambarkan terhadap waktu. *Diagram* ini terdiri dari dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Untuk menghasilkan *output* tertentu dari sebuah sistem maka harus digambarkan melalui rangkaian langkah-langkah yang diawali dengan men-*trigger* aktivitas tertentu, kemudian apa saja proses yang berubah dan dihasilkan dari sistem yang dilakukan secara internal dan *output*.



Gambar 2. 11 Sequence Diagram
(Sumber : codepolitan.com)

2.2.6 Basis Data dan DBMS

Di dalam membuat suatu sistem baik itu aplikasi atau pun *platform* pasti ada yang dinamakan dengan basis data atau biasa disebut dengan *database*. *Database* ini sangat penting keberadaannya dalam membuat suatu aplikasi yang berbasiskan

teknologi komputer. Karena tanpa *database* data yang diperoleh dan diproses dimana akan disimpan. Semua data membutuhkan penyimpanan yang terstruktur yang disesuaikan dengan kebutuhan dari suatu sistem yang akan dibuat.

Basis data atau *database* adalah suatu kumpulan informasi dan data yang bisa disimpan secara sistematis dan terstruktur serta memiliki hubungan satu sama lainnya. Fungsi dari *database* ini adalah untuk mengelompokkan data dan informasi sehingga lebih mudah didapatkan, mempermudah menyimpan, mengakses data atau informasi, mencegah terjadinya duplikasi data dan bisa mampu meningkatkan kinerja aplikasi [18].

Manfaat dari penggunaan *database* adalah data atau informasi tidak akan mengalami redundansi atau pengulangan data yang sama. Kemudian integritas terjaga dengan baik dan menjaga keamanan data.

Nama *software* yang digunakan disebut dengan *Database Management System* (DBMS). Fungsi dari DBMS sendiri itu adalah agar basis data yang dibangun dapat digunakan oleh banyak pengguna secara bersamaan dan bisa menangani data dalam jumlah yang besar [18].

2.2.7 Testing Perangkat Lunak

Membuat suatu *platform* kursus *online* pasti ada tahap *testing* atau pengujian perangkat lunak supaya *platform* yang kita buat sesuai dengan apa yang kita harapkan dan pengguna inginkan.

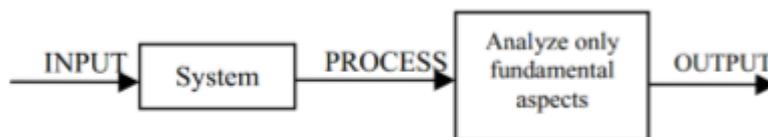
Tujuan dari melakukan pengujian perangkat lunak ini adalah untuk menemukan *error* yang ada dalam aplikasi sehingga bisa diperbaiki terlebih dahulu oleh pengembang sebelum di rilis kepada pengguna asli.

Di dalam pengujian perangkat lunak ini terbagi menjadi dua macam pendekatan yaitu :

1. *Black Box Testing*

Black box testing adalah suatu metode pengujian yang berfokus untuk menguji bagaimana cara aplikasi beroperasi dengan memasukan inputan dan melihat bagaimana *output*-nya apakah sesuai dengan apa yang diharapkan atau tidak sesuai dengan apa yang diharapkan [19].

Metode *black box* ini memungkinkan perangkat lunak mendapatkan serangkaian kondisi input yang dimana kondisi input ini ada persyaratan khusus untuk fungsional dari suatu program tertentu. Di dalam pengujian menggunakan metode *black box* ini dapat menemukan kesalahan-kesalahan seperti fungsi-fungsi yang tidak benar atau hilang, kesalahan dari antarmuka aplikasi, kesalahan dari struktur data atau akses basis data dan lain-lain.

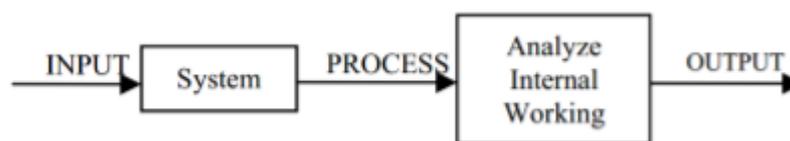


Gambar 2. 12 Sistem Kerja dari Pengujian *Black Box*
 (Sumber : Pengujian Perangkat Lunak dengan Menggunakan Metode *Black Box* dan *White Box* [15, p.4])

2. *White Box Testing*

White box testing adalah suatu metode pengujian yang berfokus pada bagaimana cara kerja perangkat lunak dari mulai melihat dari logika yang berjalan pada perangkat lunak yang diuji dengan menyediakan *test case* secara rinci dan spesifik. *White box testing* ini mengacu pada bagaimana program yang ada dalam perangkat lunak tersebut harus betul-betul benar secara 100% [19].

Dengan menggunakan metode pengujian *white box* ini pengujian melakukan pengujian kepada seluruh program dengan melihat struktur program dan *coding* nya. *White box* ini menguji kasus uji yang menguji seluruh jalur atau percabangan dalam suatu program dapat berjalan minimal satu kali, setelah itu menguji bagaimana semua keputusan logika terjadi, dan menguji seluruh struktur internal maupun eksternal yang menjamin validitas.



Gambar 2. 13 Sistem Kerja dari Pengujian *White Box*
 (Sumber : Pengujian Perangkat Lunak dengan Menggunakan Metode *Black Box* dan *White Box* [15, p.6])

2.3 Software yang Digunakan

Dalam penelitian ini peneliti menggunakan beberapa *software* untuk menunjang terselesaikannya penelitian ini dengan baik. Berikut ini adalah *software* yang peneliti gunakan untuk membangun dan mengembangkan *platform* kursus *online* ini.

2.3.1 Pemrograman PHP

Pemrograman PHP adalah salah satu bahasa pemrograman yang digunakan untuk pengembangan *web* yang berbentuk *script server-side*. Pada awalnya PHP merupakan singkatan dari *Personal Home Page* yang digunakan untuk kebutuhan *website* pribadi. Tetapi dengan perkembangan sekarang nama PHP berubah menjadi singkatan dari PHP : *Hypertext Preprocessor* [20].

PHP disebut juga sebagai bahasa pemrograman *script* karena kode programnya bisa disisipkan ke dalam dokumen HTML. *Server side scripting* adalah bahasa pemrograman yang berjalan di *server*, PHP salah satunya. Untuk bahasa pemrograman *client side scripting* adalah bahasa pemrograman yang berjalan disisi *client* (*Web browser*). Contohnya seperti *Javascript*.

Di dalam membangun *platform* yang sedang penulis kerjakan adalah menggunakan pemrograman PHP karena beberapa keunggulannya. Keunggulan dari bahasa pemrograman PHP adalah mudah dipelajari, ringkas, *open source*, pengembangannya cepat, dan *maintenance*-nya mudah.

Bahasa pemrograman PHP itu sendiri memiliki perkembangan. Dari mulai PHP versi 5.0 sampai PHP versi 7.0 yang terbaru. Penulis menggunakan PHP versi 7.0 karena sudah banyak terintegrasi dengan *server* terbaru yaitu XAMPP yang mulai juga menggunakan PHP terbaru.

2.3.2 Cascading style sheet (CSS)

Cascading style sheet atau biasa disebut dengan CSS singkatnya adalah suatu rangkaian instruksi dalam membuat sebuah *website* yang menentukan bagaimana suatu text, *layout* dan bentuk dapat tampil di halaman *website* tersebut.

Kemudian pengertian CSS itu adalah salah satu bahasa desain *website* (*style sheet language*) yang digunakan untuk mengontrol tampilan dalam sebuah halaman *web* yang ditulis menggunakan penanda yaitu disebut *markup language*. CSS ini umumnya digunakan untuk mendesain halaman HTML atau XHTML, seiring dengan perkembangan teknologi sekarang CSS bisa juga diaplikasikan untuk

berbagai macam dokumen XML, seperti dalam sistem operasi *Android*, termasuk juga SVG.

Dalam teknisnya CSS ini memisahkan antara konten utama dengan tampilan seperti *layout*, *font* dan juga warna. Fungsi dari CSS ini adalah mendesain, merancang dan juga merubah bentuk serta merubah secara keseluruhan tampilan dari suatu *website* dengan menempelkannya pada *tag-tag html*, sehingga *tag-tag* tersebut bisa lebih fungsional dan juga lebih menarik. Cara kerja dari CSS ini juga sangat mudah, cukup menempelkan kelas yang telah kita buat dari *file* terpisah yang berformat *css* maka kita bisa menuliskannya pada *selector* dalam HTML yaitu *selector id* dan *class*.

CSS itu seiring dengan pesatnya perkembangan teknologi juga semakin diperbarui. CSS3 adalah pengembangan yang lebih baru dari CSS. CSS3 ini sudah mendukung banyak efek yang bisa langsung digunakan dengan cepat. Efek ini tentu memudahkan untuk pengembang dalam hal mempercantik tampilan sebuah *website*. Efek yang dihasilkan dari CSS3 ini adalah seperti efek tulisan, gambar maupun objek. Kelebihan dari CSS3 ini juga bisa menghemat *bandwith website* karena aksesnya lebih ringan dan dapat membuat *website* lebih interaktif.

2.3.3 Template Bootstrap

Penulis menggunakan *template bootstrap* sebagai tampilan halaman *website* yang dibuat agar lebih cepat dan lebih mudah karena *template*-nya dapat didownload secara gratis.

Bootstrap adalah *front-end framework* atau *framework* (kerangka kerja) yang digunakan untuk membuat tampilan *website* menjadi lebih mudah, lebih

bagus, dan lebih menarik yang mengedepankan tampilan yang bisa dibuka dalam semua perangkat atau *device* baik itu *handphone*, *smartphone*, *laptop* atau komputer. Untuk mempermudah dan mempercepat pengembangan sebuah *website* maka bootstrap menyediakan HTML, CSS, dan *Javascript* yang siap pakai dan mudah digunakan untuk para pengembang *website*.

Secara teknisnya adalah kita sebagai pengembang tinggal menggunakan nama kelas yang sudah dibuat oleh *bootstrap* lalu kita tinggal menggunakannya dalam *tag-tag* HTML. Peneliti mengembangkan *platform* ini menggunakan *template* bawaannya bootstrap agar mudah digunakan dan juga agar *website* terlihat lebih menarik. Peneliti menggunakan bootstrap versi 4.0 yang terbaru dan bisa di download dari laman resminya yaitu <https://getbootstrap.com/>.

2.3.4 XAMPP

Dalam membangun *platform* kursus *online* ini peneliti menggunakan XAMPP sebagai *local server* untuk membuat *platform* berjalan dengan baik ketika dikembangkan dalam komputer peneliti.

Menurut situs resmi XAMPP, XAMPP adalah *software open source* yang digunakan untuk membuat *server* sendiri pada komputer kita (istilahnya adalah *Localhost*) dapat di *download* secara gratis dari situs resminya yaitu <https://www.apachefriends.org/> dan *software* ini dapat berjalan di semua sistem operasi.

Peneliti menggunakan XAMPP ini karena terdapat banyak kemudahan dalam membuat aplikasi sebelum nanti di *online* kan. Jadi peneliti membuat dulu

aplikasinya menggunakan *server localhost* pada komputer lalu kemudian di-online-kan dengan memiliki *domain* dan *hosting*.

Di dalam XAMPP ada beberapa hal penting yang harus diketahui fungsinya yaitu adalah *Htdocs*, *Phpmyadmin*, dan *Control Panel*.

1. *Htdocs* adalah salah satu *folder* yang ada dalam XAMPP yang berfungsi untuk menyimpan seluruh *file-file* dari aplikasi yang akan kita buat nantinya. Kemudian *folder Htdocs* ini akan ditampilkan di *browser* menggunakan bantuan *server localhost* sehingga aplikasi yang kita buat bisa tampil dalam *browser*.
2. *Phpmyadmin* adalah salah satu fitur yang ada dalam XAMPP yang berfungsi untuk mengolah data dalam *database website* yang akan kita kembangkan.
3. *Control Panel* adalah sama seperti Cpanel pada *website* yang sudah *online*, kalau di XAMPP *Control Panel* ini juga fungsinya kurang lebih sama yaitu mengatur *website*, *database*, *FTP*, *modul*, dan lain-lainnya.

2.3.5 Database MySQL

MySQL adalah suatu sistem manajemen pengelolaan *database* yang relasional dengan menggunakan *client-server* model. *Software* yang digunakan untuk membuat dan mengelola *database* adalah RDBMS menggunakan model relasional. Berikut ini adalah *software* yang terkait dengan RDBMS :

1. *Database* adalah sekumpulan data yang terstruktur yang mana bisa dengan mudah untuk menyimpan data dan mengambil data.
2. *Open Source, Software RDBMS* ini adalah *software* yang bebas digunakan oleh siapa saja, bebas menginstal dan bebas memodifikasinya.

3. *Client server model* adalah *software* RDBMS ini bersifat sebagai *client*, yang artinya berjalan dari sisi *client* agar bisa mengakses data dan terhubung dengan *server*.
4. SQL adalah salah satu *brand* yang terkenal dari *software* DBMS yang sudah menerapkan *client-server model*. Jadi singkatnya ketika *client* bisa berkomunikasi dengan *server* yaitu harus melewati bahasa yang spesifik dengan domain *Structured Query Language* (SQL).

Dalam menginstruksikan *server* untuk menjalankan operasi tertentu maka data *query* meminta informasi yang spesifik dari *database* yang sudah ada, kemudian melakukan manipulasi data seperti menambah, menghapus, melakukan operasi bilangan, dan juga bisa mengubah data [21].

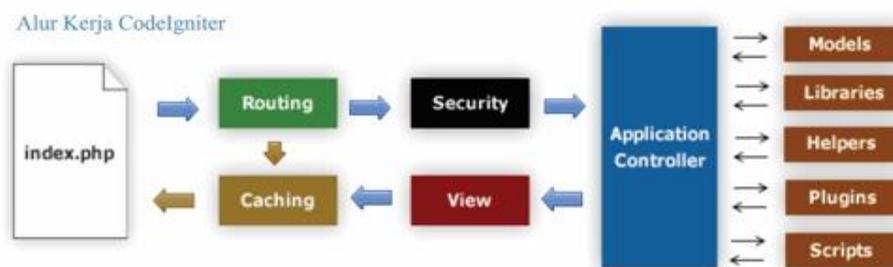
2.3.6 Framework Codeigniter

Framework Codeigniter adalah sebuah kerangka kerja atau *framework* untuk mengembangkan *website* agar lebih cepat dan mudah karena sudah disediakan dokumentasinya dan *source code* nya untuk memudahkan para pengembang *website*.

Peneliti menggunakan *framework* dalam membangun *platform* ini karena ingin membuat suatu *platform* yang cepat dan mudah dalam pembangunan dan pengembangannya. Ketika menggunakan *framework* dalam hal ini *framework* PHP dapat dengan mudah melakukan *maintenance* karena sudah ada pola terstruktur guna membantu pengembang dalam membuat *website* [22].

Codeigniter ini adalah sebuah *framework* PHP yang dibangun untuk membuat suatu *website* agar lebih mudah digunakan oleh pengembang. Codeigniter

ini sebuah *web application network* yang sifatnya *open source* atau terbuka untuk digunakan oleh siapapun guna membangun aplikasi PHP yang dinamis.



Gambar 2. 14 Alur Kerja *Framework* CodeIgniter
(Sumber : <https://idcloudhost.com/panduan/mengenal-apa-itu-framework-codeigniter/>)

2.3.7 *Visual Studio Code*

Salah satu *software* yang peneliti gunakan untuk menuliskan kode-kode program dalam membangun *platform* ini adalah *Visual Studio Code*. *Visual Studio Code* ini adalah aplikasi *editor* pertama yang bisa berjalan di banyak *platform* perangkat lunak yaitu seperti OS X, Ubuntu dan windows.

Visual studio code ini dibuat oleh Microsoft pada tahun 2015 dan sekarang sudah banyak yang menggunakannya. Editor ini keunggulannya bisa *support* dengan berbagai macam bahasa pemrograman.

Peneliti begitu menyukai editor ini karena banyak *plug-in* yang bisa membantu dalam pengkodean program, salah satunya adanya *support debugging*, dan juga arsitektur *editor* yang sangat mudah dan enak digunakan sangat cocok. Berikut ini peneliti cantumkan keunggulan *Visual Studio Code* sebagai aplikasi editor yang terbaik menurut versi penulis yaitu :

1. *Cross platform* atau bisa digunakan di berbagai sistem operasi dari mulai windows, linux, macOS yang otomatis bisa memudah kita dalam melakukan pengkodean program dimanapun.
2. *Lightweight* adanya fitur ini membuat pengembang jadi lebih dimanja untuk bisa lebih mengontrol sepenuhnya tema yang digunakan, bahasa, *commands* dan juga *debugger*.
3. *Editor* yang sangat *powerful* karena memberikan fitur yang sangat begitu produktif, seperti adanya *snippets*, *IntelliSense*, *autocorrect* dan *formatting*.
4. Melakukan *code debugging* yang sangat mudah dengan cara mengawasi kode, variabel, *expression* dan juga *call stack* yang mana saja.
5. *Control source code* yang bisa terintegrasi seperti melakukan integrasi dengan Git. Tentu ini sangat memudahkan pengembang dalam melakukan *controlling code* bersama tim.
6. Dan yang terakhir adalah pengembang bisa membuat banyak terminal dalam satu *editor*, tentu ini menjadi keunggulan yang jarang ada di *editor* yang lain.