

BAB 2

LANDASAN TEORI

2.1. Sistem Informasi

Sistem informasi merupakan seperangkat komponen yang saling terkait untuk mengumpulkan, memanipulasi, menyimpan, dan menyebarluaskan data dan informasi dan memberikan umpan balik untuk mencapai suatu tujuan. (Stair & Reynolds., 2016:4)

Terdapat tiga kata kunci dalam pengertian di atas: kumpulan (dikelompokkan, mengumpulkan), pengolahan (diproses, menyimpan, menganalisis), dan distribusi (menyebarkan). Sebuah alat pendukung yang dapat memenuhi kebutuhan tersebut di atas guna mendapat tujuan yang spesifik dikatakan sebagai sebuah sistem informasi.

Komponen yang berkaitan dengan sistem informasi yaitu data, informasi, proses, dan pengetahuan.

- Data mengacu pada fakta mentah, seperti nama peserta pelatihan, tanggal terjadinya pelatihan, teks tautan ke sebuah materi, dan sebagainya.
- Informasi merupakan kumpulan fakta disusun dan diproses sedemikian rupa sehingga mereka memiliki nilai di luar fakta-fakta individual.
- Proses yaitu sekumpulan tugas yang saling berkaitan secara logis yang dilakukan untuk mencapai hasil yang ditentukan. Dalam hal ini, misalnya nama seseorang digabungkan dengan data dan proses pendaftaran sehingga didapat informasi bahwa nama tersebut terdaftar sebagai peserta.
- Pengetahuan, yaitu kesadaran dan pemahaman tentang satu set informasi dan cara informasi tersebut dapat dibuat bermanfaat. Misalnya, dari informasi dan proses pendaftaran, diketahui bahwa dibutuhkan aksi lebih lanjut agar pelatihan dapat berjalan ke tahap selanjutnya.

Istilah-istilah lain yang berkaitan dengan sistem informasi yaitu:

- *Input*, yaitu kegiatan mengumpulkan dan menangkap data mentah.
- *Processing*, mengubah data menjadi output yang bermanfaat.
- *Output*, menghasilkan informasi yang berguna.
- *Feedback*, informasi dari sistem yang digunakan untuk membuat perubahan pada aktivitas input atau pemrosesan.

2.2. *Blended Learning*

Iskandar, dkk (2014:526) mengemukakan bahwa *blended learning* merupakan sebuah campuran yang mengkombinasikan beberapa cara penyampaian pembelajaran, seperti perangkat lunak kolaborasi, pembelajaran berbasis web, dan media komunikasi komputer dengan instruksi *face-to-face* tradisional.

Dalam jurnalnya, Iskandar, dkk (2014:526) juga menyampaikan prinsip dasar *blended learning* adalah perpaduan pembelajaran *face-to-face* dan online yang optimal sedemikian rupa sehingga kelebihan dari masing-masing berpadu menjadi pengalaman pembelajaran yang unik, sesuai dengan konteks dan tujuan pembelajaran. Jadi, *blended learning* menggabungkan pembelajaran *face-to-face* dan online yang memanfaatkan teknologi untuk mencapai hasil pembelajaran yang optimal bagi pesertanya.

2.3. *E-Learning*

Dalam Wahyuningsih & Makmur (2017:3), dikemukakan bahwa terdapat dua pandangan mengenai *e-learning*. Definisi pertama seperti yang disampaikan oleh Allen (2013) yang menjelaskan bahwa *e-learning* merupakan segala bentuk aktivitas pembelajaran yang memanfaatkan media elektronik untuk belajar. Definisi ini lebih menekankan pada penggunaan segala bentuk elektronik untuk membantu manusia belajar. Definisi kedua dikemukakan oleh Rosenberg (2001) yang lebih menjelaskan *e-learning* sebagai penggunaan teknologi internet dan komputer berjaringan untuk membantu proses belajar manusia.

E-learning tidak hanya menjadi pemecah masalah untuk pembelajaran jarak jauh. Namun *e-learning* juga diterapkan sebagai inovasi dan pendukung dari pembelajaran konvensional secara tatap muka.

2.4. **Basis Data**

Dalam bukunya *Basis Data*, Fatansyah (2007:2) mengemukakan beberapa definisi tentang *database* atau basis data dari sejumlah sudut pandang:

- a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
- c. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

DBMS atau *Database Management System* merupakan perantara bagi pemakai dengan basis data dalam *disk*. Pengguna dapat berinteraksi dengan basis data tersebut melalui suatu bahasa khusus yang ditetapkan oleh masing-masing perusahaan pembuat DBMS. Bahasa basis data terdiri atas sejumlah perintah (*statement*) yang diformulasikan dan dapat diberikan *user* dan dikenali/diproses oleh DBMS untuk melakukan suatu aksi/ pekerjaan tertentu.

Contoh-contoh bahasa basis data adalah SQL, dBase, QUEL, dan sebagainya. Pada pengembangan sistem aplikasi di sini, akan digunakan bahasa SQL dengan DBMS PostgreSQL.

Bahasa basis data biasanya dapat dipilah ke dalam dua bentuk yaitu:

1. *Data Definition Language* (DDL)

DDL merupakan skema basis data yang menggambarkan desain basis data secara keseluruhan. Dengan DDL, kita dapat membuat database baru, tabel baru, membuat indeks, mengubah tabel, menentukan struktur penyimpanan tabel, dan sebagainya.

2. *Data Manipulation Language* (DML)

DML merupakan bentuk bahasa basis data yang berguna untuk melakukan manipulasi dan pengambilan data pada suatu basis data. Manipulasi data antara lain berupa:

- Penyisipan/ penambahan data baru ke suatu basis data.
- Penghapusan data dari suatu basis data.
- Pengubahan data di suatu basis data.

2.5. *Agile Development*

Menurut Shore & Warden (2008:9) metode agile merupakan proses yang mendukung filosofi ketangkasan. Metode agile terdiri dari elemen-elemen individu yang disebut praktik. Praktik yang dilakukan dalam metode agile antara lain penggunaan kontrol versi, penggunaan standar kode, dan demo mingguan kepada *stakeholder*. Praktik agile sering melakukan tugas rangkap, memecahkan berbagai

masalah pengembangan perangkat lunak secara bersamaan dan saling mendukung dengan cara yang cerdas.

Metode agile secara resmi dimulai dengan *Agile Manifesto* pada tahun 2001. Manifesto Agile dapat dilihat di <http://agilemanifesto.org/iso/id/manifesto.html>.

Penulis manifesto agile mengemukakan bahwa mereka menghargai:

- Individu dan interaksi lebih dari proses dan sarana perangkat lunak
- Perangkat lunak yang bekerja lebih dari dokumentasi yang menyeluruh
- Kolaborasi dengan klien lebih dari negosiasi kontrak
- Tanggap terhadap perubahan lebih dari mengikuti rencana

Pengembangan agile dapat diterapkan dengan beberapa metode, antara lain *Extreme Programming*, *Kanban*, dan *Scrum*.

Menurut Measey & Radtac (2015:30) pola pikir Agile diartikan sebagai organisasi atau orang telah menyerap Agile pada taraf tertentu yang menjadi bagian dari identitas mereka. Agile bukan tentang 'melakukan' Agile, tetapi 'menjadi Agile' serta memiliki pola pikir Agile dan ada beberapa alat, proses, praktik yang dapat memfasilitasi hal ini.

Agile merupakan sebuah perjalanan bukan tujuan dan hal terbaik yang dapat diharapkan oleh sebuah organisasi adalah tim mereka menjadi lebih Agile dengan menanamkan pola pikir Agile lebih dalam pada diri mereka dan organisasi. Proses ini difasilitasi dengan menerapkan nilai, prinsip dan praktik Agile. Berikut merupakan karakteristik *Agile mindset* dan *Fixed mindset* (non-Agile).

Tabel 2.1 Perbandingan karakteristik *Agile mindset & Fixed Mindset*

	<i>Fixed Mindset</i>	<i>Agile Mindset</i>
Kemampuan	Statis, seperti tinggi	Dapat bertumbuh seperti otot
Tujuan	Agar terlihat baik	Untuk belajar
Tantangan	Menghindari	Melingkupi
Kegagalan	Mendefinisikan identitas	Memberikan informasi
Upaya	Bagi yang tanpa berbakat	Menjadi jalan untuk mahir
Reaksi menghadapi tantangan	Tidak berdaya	Memiliki ketahanan

Menurut Cline (2015:66) Agile menggunakan teknik yang dinamakan *Adaptive Planning*. Hanya kebutuhan dan gambaran kasar yang digunakan, lalu setiap tahap ke depan ditekuni lebih rinci untuk diimplementasikan. Dalam setiap iterasi, kebutuhan dan implementasi produk ditinjau sebelum mengumpulkan informasi yang lebih rinci dari customer.

Selama pengembangan produk, setiap perubahan yang dilakukan membutuhkan biaya yang lebih tinggi jika dilakukan di akhir. Oleh karena itu dengan sebagian produk yang ditinjau secara periodik, biaya perubahan akan lebih minimum karena pekerjaan yang lebih luas belum selesai dilakukan sehingga lebih sedikit perubahan.

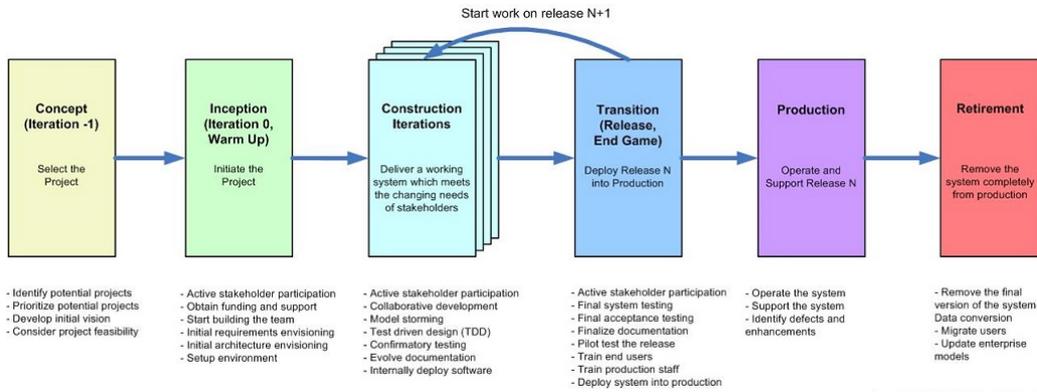
Kebutuhan pada Agile digambarkan melalui *use case*, potongan kecil dari ruang lingkup yang menggambarkan interaksi antara *user-system*, kebutuhan *user* bahkan bagian paling kecil dari ruang lingkup.

Developer menulis setiap skenario lalu meninjau kembali dengan user melalui demo. Proses ini mungkin terjadi beberapa kali dalam iterasi. Terkadang dalam siklus meninjau, user dapat mengajukan perubahan yang mereka inginkan dan pengembangan akan direvisi dengan cepat sehingga lebih murah.

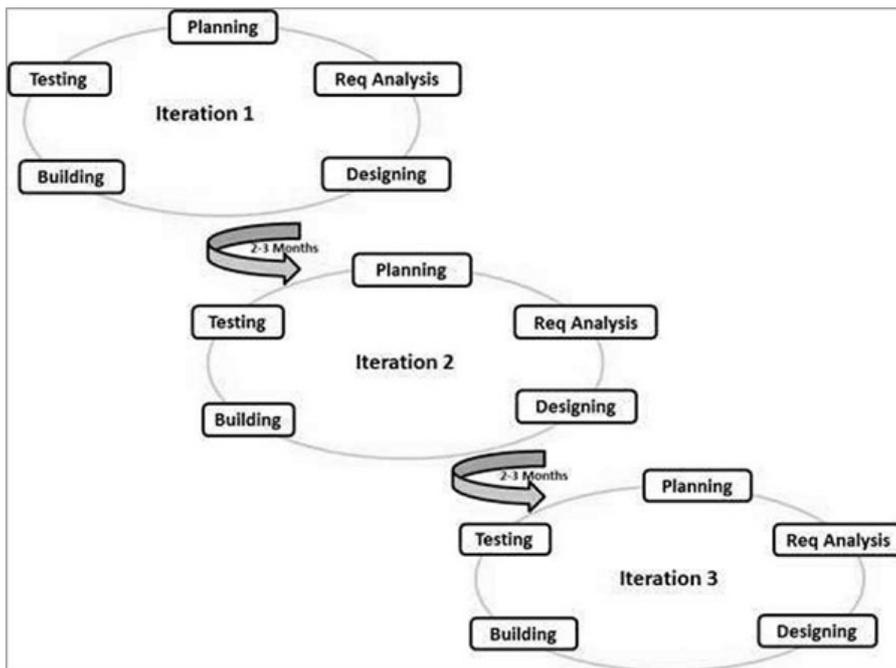
Menurut Ambler & Lines (2012:3) metode-metode Agile inti – seperti Scrum, XP, dan Agile Modeling (AM) – fokus pada kegiatan yang berorientasi konstruksi. Mereka ditandai oleh siklus hidup yang berorientasi produk/*value* di mana perangkat

lunak berpotensi *shippable* berkualitas tinggi diproduksi secara teratur oleh tim yang sangat kolaboratif dan *self-organized*.

Berikut diagram yang mengilustrasikan siklus hidup sistem dalam metodologi Agile.



Gambar 2.1 Siklus Tingkat Tinggi Agile (Ambler, 2012)

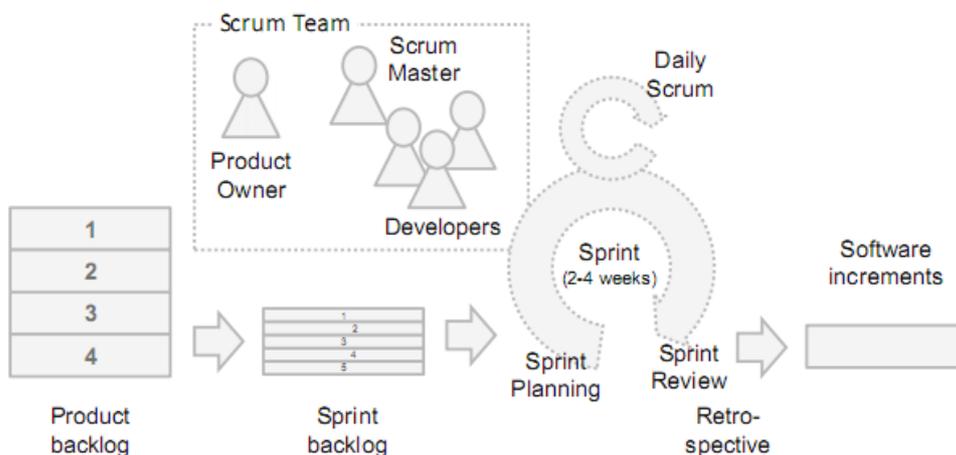


Gambar 2.2 Siklus Hidup Pengembangan Sistem pada Metodologi Agile (SDLC Agile Model, 2018)

2.6. Scrum Methodology

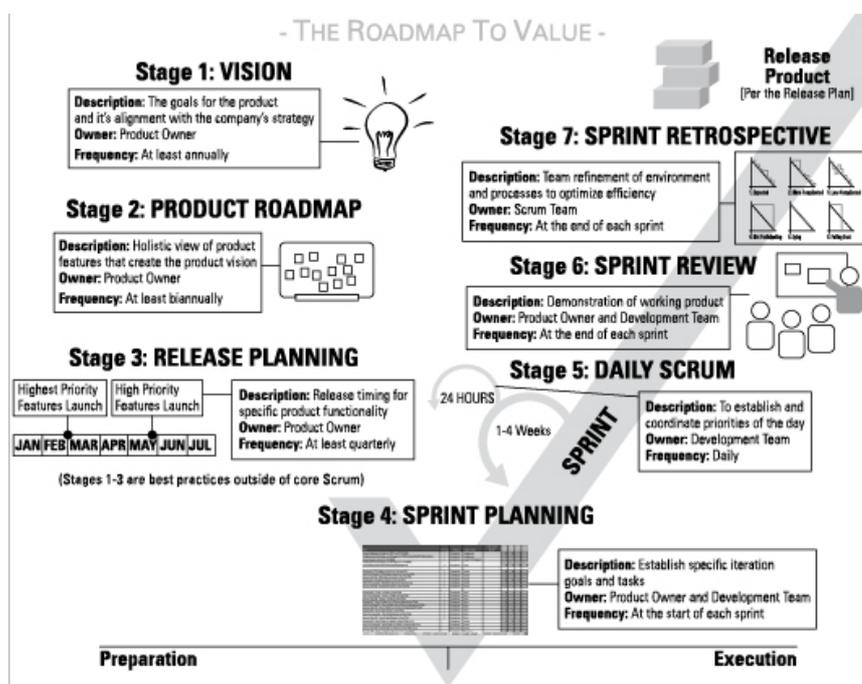
Metode pengembangan yang digunakan dalam proses pengembangan aplikasi ini adalah Scrum.

Berikut merupakan gambaran dari seluruh elemen pada Scrum menurut Schmidt (2016:17):



Gambar 2.3 Elemen Scrum

Pada Scrum ada 2 peranan spesifik yaitu *Scrum Master* dan *Product Owner*. *Scrum Master* berperan sebagai fasilitator yang bertanggung jawab untuk mengawasi proses pada Scrum dan menghindari segala permasalahan yang mungkin dapat menghambat pekerjaan tim. Sementara *Product Owner* merepresentasikan *customer* dalam tim dan menyuarakan kebutuhan *customer*. Anggota tim yang lain masuk ke dalam tim *development* dan melakukan analisis kebutuhan, perancangan, pengembangan dan validasi *software*.



Gambar 2.4 Iterasi pada Agile (Layton & Ostermiller, 2017)

Dalam Scrum, dilakukan iterasi dan membagi project menjadi siklus pengembangan yang lebih pendek, yang dikenal dengan Sprints. Setiap Sprint diawali dengan *meeting Sprint Planning* dimana tim akan menentukan fitur yang akan diimplementasikan. Kemudian anggota tim akan menspesifikasikan sub-pekerjaan dan menugaskan pada *developer* secara individu. Semua anggota tim akan bertemu setiap hari sekitar 15 menit untuk sinkronisasi pekerjaan dan agar progress pekerjaan dalam tim berjalan transparan. Setiap developer mengemukakan pencapaiannya, pekerjaan yang sedang dilakukan saat ini dan masalah-masalah yang ada dalam tim.

Tim mengatur pekerjaan-pekerjaan development dengan menggunakan *product backlog*. *Backlog* berisi daftar pekerjaan yang diprioritaskan oleh *Product Owner*. Tim *development* membagi *backlog* menjadi *sprint backlog items* dan mengikuti *progress* nya di setiap Sprint.

Setiap sprints akan diakhiri dengan *meeting Sprint Review* pada waktu tim menyampaikan progress pada *Product Owner* atau langsung pada *customer*.

2.7. UML

Perencanaan Agile merupakan dasar yang digunakan untuk membantu mengontrol perubahan (Fowler, 2018). Tim yang Agile merencanakan sama cermatnya dengan tim tradisional, tetapi rencananya terus berubah untuk mencerminkan hal-hal yang dipelajari selama proyek. Sukses didasarkan pada nilai yang diberikan oleh produk.

Menurut Fenn (2017) metode pengembangan yang Agile membantu pelanggan menyempurnakan atau menyetujui desain sistem sambil memastikan semua orang setuju pada hasil akhirnya. Memperluas pendekatan visual ini ke titik awal pengembangan tangkas dengan menggunakan diagram bahasa pemodelan terpadu (UML) dapat membantu menghasilkan hasil nyata untuk sebuah proyek.

UML atau *Unified Modeling Language* diciptakan pada tahun 1995 oleh Grady Booch, Ivar Jacobson, dan James Rumbaugh. UML merupakan sekumpulan standar teknik pembuatan diagram yang bertujuan untuk menyediakan kosakata umum dari istilah-istilah metode berorientasi objek. UML menyediakan teknik diagram yang cukup lengkap untuk memodelkan proses pengembangan sistem dari tahap analisis hingga tahap implementasi (Dennis, Wixom, & Tegarden., 2015)

UML terdiri dari beberapa diagram untuk menggambarkan 2 jenis pemodelan: *structure diagrams & behavioral diagrams*.

2.7.1. Structure Diagrams

Structure diagrams menyediakan cara untuk merepresentasikan data dan hubungan statis dalam suatu sistem informasi. Diagram struktur meliputi diagram *class*, *object*, *package*, *deployment*, *component*, *composite structure*, dan diagram *profile*.

- a. *Class diagram* menggambarkan hubungan antar class yang dimodelkan di dalam sistem, biasanya dibuat pada tahap analisis dan desain.
- b. *Object diagram* menggambarkan hubungan antara objek yang dimodelkan dalam sistem, digunakan saat instance yang sebenarnya dari class dapat mengkomunikasikan model dengan lebih baik. *Object diagram* juga dibuat pada tahap analisis dan desain.
- c. *Package diagram* mengelompokkan elemen-elemen UML untuk membentuk konsep dengan level lebih tinggi. *Package diagram* dibuat dalam tahap analisis, desain, dan implementasi.
- d. *Deployment diagram* menunjukkan arsitektur fisik dari sistem, dibuat pada tahap desain fisik dan implementasi. *Deployment diagram* juga dapat digunakan untuk menunjukkan komponen-komponen perangkat lunak yang akan di-*deploy* ke dalam arsitektur fisik sistem.
- e. *Component diagram* menggambarkan relasi fisik antara komponen-komponen perangkat lunak, dibuat pada tahap desain fisik dan implementasi sistem.
- f. *Composite structure design diagram* menggambarkan struktur internal dari sebuah *class*. *Composite structure design diagram* dibuat pada tahap analisis dan desain.
- g. *Profile diagram* digunakan untuk mengembangkan ekstensi dari UML.

2.1.1. Behavioral Diagrams

Behavioral diagrams menyediakan cara untuk menggambarkan hubungan dinamis antara objek-objek atau instance-instance yang melambangkan sistem informasi bisnis. Diagram-diagram ini juga memungkinkan pemodelan tingkah laku dinamis dari objek-objek secara individual sepanjang masa hidupnya

(*lifetime cycle*). *Behavioral diagrams* meliputi diagram *activity*, *sequence*, *communication*, *interaction overview*, *timing*, *behavior state machine*, *protocol state machine*, dan *use-case*.

- a. *Activity diagram* menggambarkan alur bisnis, alur aktivitas dalam sebuah *use case*, atau desain detail dari sebuah metode. *Activity diagram* dibuat pada tahap analisis dan desain.
- b. *Sequence diagram* memodelkan tingkah laku objek dalam sebuah *use case*, berfokus pada urutan aktivitas sesuai waktu pelaksanaannya. *Sequence diagram* dibuat pada tahap analisis dan desain.
- c. *Communication diagram* memodelkan tingkah laku objek dalam sebuah *use case*, berfokus pada jalur komunikasi antar sekumpulan objek yang bekerja sama dalam sebuah aktivitas bisnis. *Communication diagram* dibuat pada tahap analisis dan desain.
- d. *Interaction overview diagram* menggambarkan alur kontrol dari sebuah process. *Interaction overview diagram* dibuat pada tahap analisis dan desain.
- e. *Timing diagram* menggambarkan interaksi antara sekumpulan objek dan perubahan *state*-nya seiring waktu. *Timing diagram* dibuat pada tahap analisis dan desain.
- f. *Behavioral state machine diagram* menjelajah tingkah laku dari sebuah *class*. *Behavioral state machine diagram* dibuat pada tahap analisis dan desain.
- g. *Protocol state machine diagram* menggambarkan dependensi antara *interface* yang berbeda dari sebuah *class*. *Protocol state machine diagram* dibuat pada tahap analisis dan desain.
- h. *Use-case diagram* menangkap kebutuhan bisnis untuk sistem dan menggambarkan interaksi antara sistem dan lingkungannya. *Use-case diagram* dibuat pada tahap analisis.

2.8. Wireframes

Wireframing merupakan alat penting untuk perancangan dan pengembangan produk (Bank & Zuberi). *Wireframes* biasanya dibuat saat proses perancangan produk dan proses pengembangan dengan satu atau lain cara, baik itu menggunakan coretan cepat pada kertas gores atau desain terstruktur pada buku sketsa rancangan.

Desainer sistem atau produk, *developer*, dan manajer produk biasanya yang paling sering membuat dan memanfaatkan wireframes dalam pekerjaan sehari-hari mereka. Namun, selain mereka, banyak orang mendapat manfaat dari *wireframes*. Termasuk *business analyst*, *information architect*, *UI/ UX designer*, *graphic designer*, *developer*, dan *product manager*.

Dengan kata lain, *wireframes* merupakan alat yang praktis dan bermanfaat bagi siapa pun yang terlibat dalam pengembangan produk dalam kapasitas apa pun.

Wireframes merupakan cetak biru untuk desain yang menghubungkan struktur konseptual dasar (arsitektur informasi) ke permukaan (desain visual) dari situs web atau aplikasi seluler. Lebih khusus lagi, mereka representasi visual dari sebuah antarmuka, digunakan untuk mengkomunikasikan detail-detail sebagai berikut:

- Struktur. Bagaimana potongan-potongan situs ini disatukan?
- Konten. Apa yang akan ditampilkan di situs?
- Hirarki informasi. Bagaimana informasi ini diatur dan ditampilkan?
- Fungsionalitas. Bagaimana cara kerja antarmuka ini?
- Perilaku. Bagaimana cara berinteraksi dengan pengguna? Dan bagaimana cara kerjanya?

2.9. Ruby dan Ruby on Rails

Ruby merupakan salah satu bahasa pemrograman berbasis objek. Dalam Ruby, semua benda adalah objek. Segala sesuatu yang dapat dimanipulasi merupakan suatu objek dan hasil dari manipulasi tersebut juga merupakan sebuah objek.

Di mana Ruby merupakan bahasa pemrograman, Ruby on Rails merupakan sebuah *framework* atau kerangka kerja, seperangkat struktur dan konvensi untuk membangun aplikasi web menggunakan bahasa Ruby. Ruby on Rails, atau biasa disebut Rails, juga merupakan pustaka (*library*) atau kumpulan *gems* (program yang dibuat dengan bahasa Ruby) yang digunakan pengembang sebagai inti aplikasi web Rails apa pun. Dengan menggunakan Rails, kita mendapatkan kode teruji yang mengimplementasikan banyak fitur yang paling dibutuhkan dari situs web dinamis. (Kehoe., 2013:10)

2.10. Minimum Viable Product (MVP)

MVP didefinisikan sebagai produk dengan fitur yang cukup untuk mengumpulkan pengetahuan yang valid tentang produk, yang merupakan fokus

utama pada tahap awal. MVP memainkan peran yang penting tidak hanya untuk *startup*, tetapi juga untuk *stakeholder* eksternal seperti user yang berpotensi, investor dan mentor. Sekarang ini MVP menjadi kunci untuk dipresentasikan pada saat meeting dengan investor. Terdapat beberapa tipe MVP, bervariasi berdasarkan upaya pengembangan, tujuan dan tahapan yang sering muncul. Sebagai contoh, *landing page* sebagai salah satu MVP yang dapat dibuat dengan cepat untuk mengomunikasikan proposal dari sebuah produk ke publik. Sebuah prototipe dengan satu fitur sebagai MVP mungkin membutuhkan beberapa bulan untuk pembuatan dan integrasi ke produk akhir. Selain itu, MVP yang berbeda mungkin digunakan untuk memberikan tujuan yang sama, sebagai contoh untuk berkomunikasi dengan investor. Sedikit yang tahu bahwa MVP juga digunakan untuk karya, dari komunitas praktisi dan peneliti. (Duc & Abrahamsson., 2016:1)

2.11. Object Oriented System Analysis and Design (OOSAD)

Pendekatan *object oriented* untuk merancang sistem informasi dapat digunakan oleh berbagai metodologi tradisional. Namun bagaimana pun, pendekatan objek oriented sangat berhubungan dengan tahapan pengembangan RAD atau metodologi Agile. Pada pendekatan tradisional, proses dekomposisi masalah dilakukan dengan *process-centric* atau *data-centric*. Namun proses dan data saling berkaitan sehingga sulit untuk menentukan fokus utama. Berdasarkan kurangnya keselarasan dengan dunia nyata, metodologi *object oriented* telah memunculkan fase SDLC tapi mencoba menyeimbangkan penekanan pada proses dan data dengan memfokuskan pada dekomposisi masalah sebagai objek yang berisi data dan proses. (Dennis, Wixom, & Tegarden., 2015:23)

2.12. Penelitian Kualitatif

Penelitian kualitatif, kriteria mutunya ditentukan oleh kredibilitas peneliti, yang mencakup pengetahuan yang cukup, pengalaman dan pemahaman konteks yang mendalam. Selain itu juga, data dan informasi yang diperoleh benar-benar berasal dari orang yang mengalami langsung peristiwa, gejala, fakta atau realita tersebut dan mampu mengungkapkan dan menceritakannya kembali secara jelas kepada peneliti. (Semiawan, 2010)

2.13. Metode Pengumpulan Data

Pengumpulan data merupakan suatu prosedur yang sistematis dan standar untuk memperoleh data yang diperlukan, ada hubungan antara metode pengumpulan data dengan masalah penelitian yang ingin dipecahkan. Pengumpulan data dapat dilakukan salah satunya dengan cara wawancara / *interview*. Wawancara atau *interview* merupakan proses memperoleh keterangan untuk tujuan penelitian dengan cara tanya jawab, sambil bertatap muka antara pewawancara dengan responden menggunakan alat yang dinamakan *interview guide*. (Nazir, 2013:179)

2.14. Metode Studi Pustaka

Studi pustaka merupakan teknik pengumpulan data dengan mengadakan studi penelaah terhadap buku-buku, literatur-literatur, catatan-catatan dan laporan-laporan yang ada hubungannya dengan masalah yang dipecahkan. Teknik ini digunakan untuk memperoleh dasar-dasar dan pendapat secara tertulis yang dilakukan dengan cara mempelajari berbagai literatur yang berhubungan dengan masalah yang diteliti. (Nazir, 2013:94)