

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Teori Umum

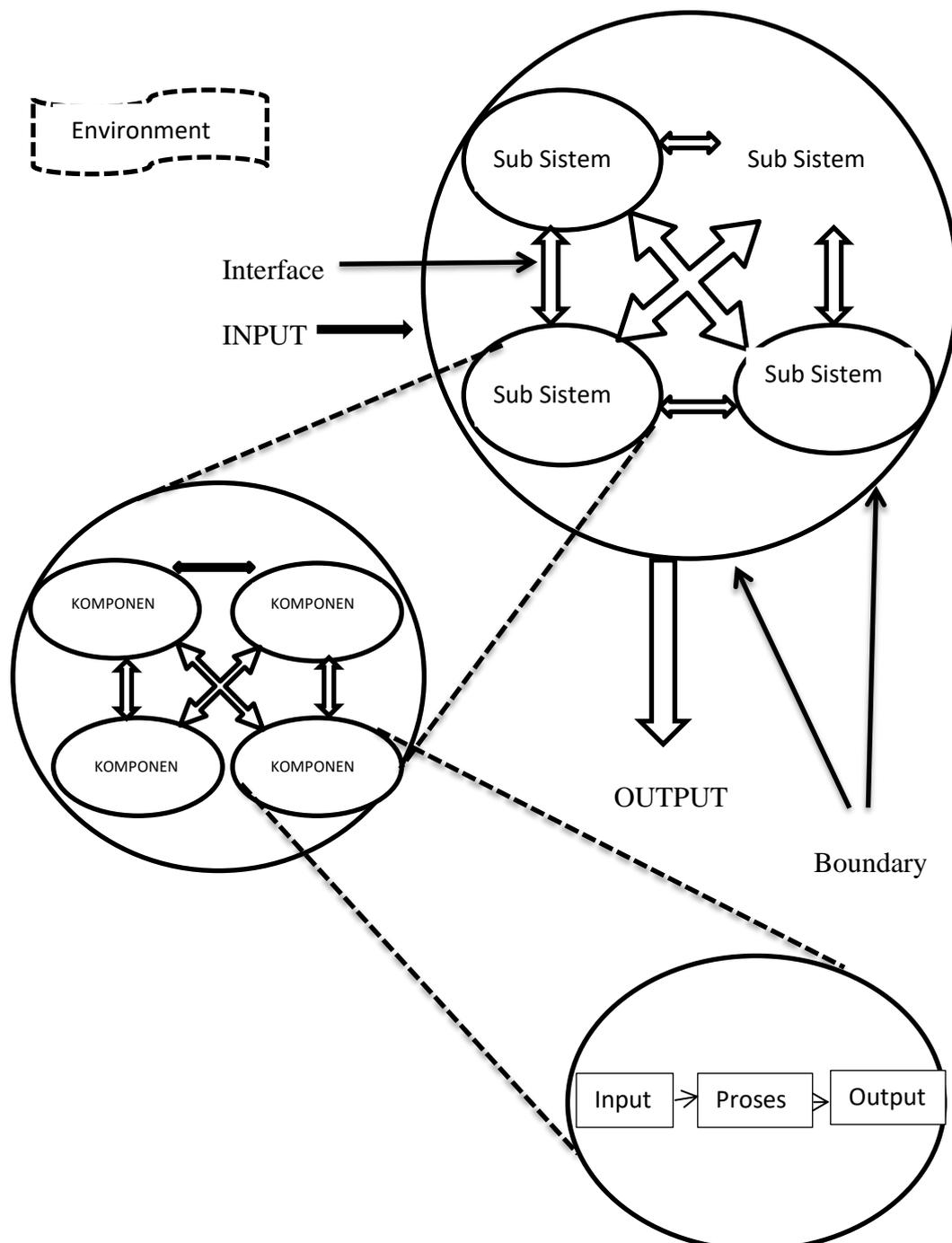
Dalam penulisan skripsi ini, penulis menggunakan teori umum dan teori khusus yang menjadi patokan dalam penulisan skripsi ini dan sebagai bahan acuan. Teori adalah seperangkat konsep, defenisi dan proposisi yang tersusun secara sistematis sehingga dapat digunakan untuk menjelaskan dan meramalkan fnomena (Sugiyono, 2012: 53).

2.1.1. Defenisi Sistem

Sistem (*system*) adalah kumpulan dari sub-sub sistem, elemen-elemen, prosedur-prosedur, yang saling berintegrasi untuk mencapai tujuan tertentu, seperti informasi, target atau *goal*. Karakter suatu sistem terdiri dari : komponen (*Components*), Batas Sistem (*Boundary*), Lingkungan luar Sistem (*Environments*), Penghubung (*Interface*), *input*, *process* dan *output*, Sasaran (*Objectives*), Tujuan (*Goal*).

Menurut buku *Conceptual, Structure and Development*, “Sistem dapat bersifat abstrak atau *fisis*. Sistem yang abstrak adalah susunan yang teraktur dari.

gagasan-gagasan atau konsep-konsep yang saling tergantung” (Hapzi Ali ; & Tonny, 2010: 8).



Gambar 2. 1 Karakteristik Sistem

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata adalah suatu objek nyata, seperti tempat, benda, dan orang-orang yang betul-betul ada dan terjadi (Rismawati, 2014: 321).

2.1.2. Karakteristik Sistem

Sebuah sistem memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut dapat dikatakan sebagai suatu sistem. Suatu sistem memiliki karakteristik atau sifat-sifat tertentu, yaitu (Muslihudin, 2016). :

1. Batas Sistem

Batas Sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

2. Lingkungan luar (*environment*)

Lingkungan luar (*environment*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energy dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup sistem.

3. Penghubung sistem (*interface*)

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Keluaran (*output*) untuk subsistem yang lainnya dengan melalui penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

4. Masukan Sistem (*input*)

Masukan (*input*) adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh di dalam komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan data adalah *signal input* untuk diolah menjadi informasi.

5. Keluaran Sistem (*output*)

Keluaran (*output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keuaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem. Misal nya untuk sistem komputer, panas yang dihasilkan adalah keluaran yang tidak berguna dan merupakan hasil sisa pembuangan, sedang informasi adalah keluaran yang dibutuhkan.

6. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi. Sistem akuntansi akan mengolah data-data transaksi menjadi laporan-laporan yang dibutuhkan oleh manajemen.

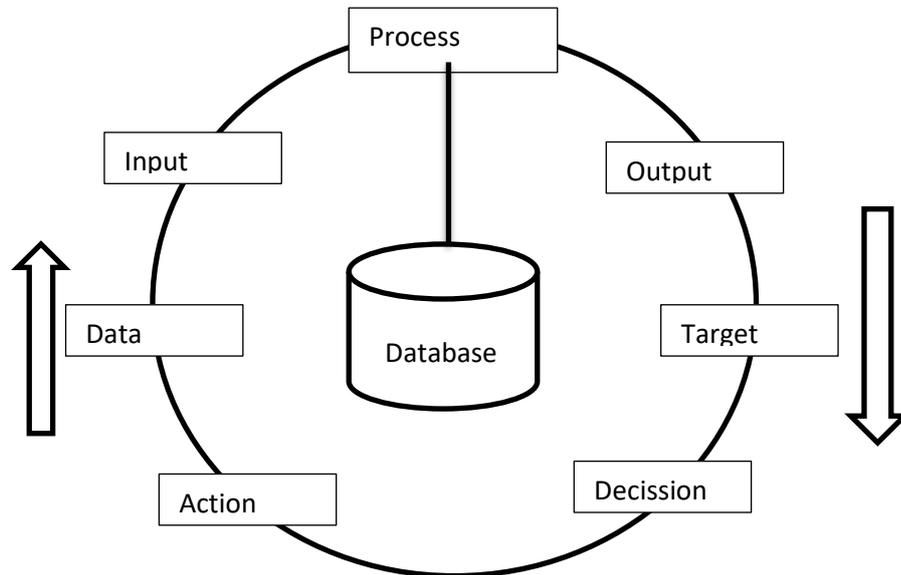
7. Sasaran Sistem (*goal*)

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

2.1.3. Defenisi Informasi

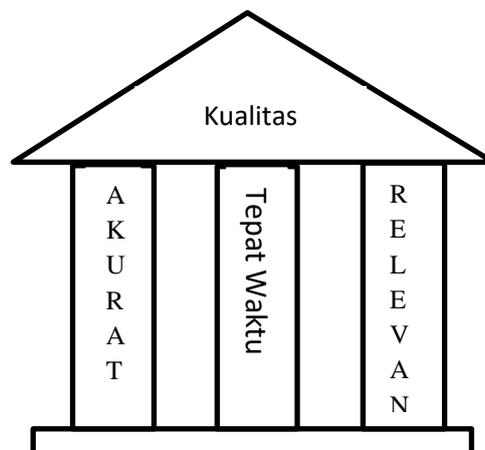
Informasi (*information*) adalah data yang telah diolah menjadi suatu bentuk yang penting bagi penerima dan mempunyai nilai yang nyata atau dapat dirasakan manfaatnya dalam keputusan-keputusan yang akan datang. *Output* informasi dari komputer digunakan oleh para Manager, non Manager, serta orang-orang dan organisasi-organisasi dalam lingkungan perusahaan (Hapzi Ali ; & Tonny, 2010: 10).

Menurut (Muslihudin, 2016; 9). informasi adalah data yang diolah menjadi bentuk yang lebih berguna untuk membuat keputusan.



Gambar 2. 2 Siklus Informasi

Kualitas dari suatu sistem informasi (*quality of information*) tergantung dari tiga hal, yaitu informasi harus akurat (*accurate*), tepat waktunya (*timelines*) dan relevan (*relevance*). John Burch dan Gary Grudnitski menggambarkan kualitas dari informasi dengan bentuk bangunan yang ditunjang oleh tiga buah pilar.



Gambar 2. 3 Pilar Kualitas Informas

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bisa menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusak informasi tersebut
2. Tepat waktu, berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan.
3. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda. Misalnya informasi mengenai sebab-musabab kerusakan mesin produksi kepada akuntan perusahaan adalah kurang relevan dan akan lebih relevan bila di tujukan kepada ahli teknik perusahaan.

2.1.4. Defenisi Sistem Informasi

Sistem informasi (*information system*) merupakan suatu kumpulan dari komponen-komponen dalam suatu perusahaan atau organisasi yang berhubungan dengan proses penciptaan dan pengaliran sistem informasi. Dalam hal ini, TI merupakan salah satu komponen dalam perusahaan. Komponen-komponen yang lainnya adalah prosedur, struktur organisasi, sumberdaya manusia, produk, pelanggan, rekanan dan sebagainya (Hapzi Ali ; & Tonny, 2010: 13).

Menurut (Muslihudin, 2016) sistem informasi adalah suatu alat untuk menyajikan informasi, sedemikian rupa, sehingga bermamfaat bagi penerimanya.

Sistem informasi adalah cara yang terorganisir untuk mengumpulkan, memasukkan, dan memroses data dan menyimpannya, mengelola, mengontrol dan melaporkannya sehingga dapat mendukung perusahaan atau organisasi untuk mencapai tujuan (Tantra, 2012: 2).

2.1.5. Defenisi Komponen Sistem Informasi

John Burch dan Gary Grudnitski mengemukakan bahwa sistem informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan (*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*), blok basis data (*database block*) dan blok kendali (*controls block*) (Muslihudin, 2016: 11).

1. Blok masukan (*input block*)

Input mewakili data yang masuk ke daam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok keluaran (*output block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen semua pemakai sistem.

4. Blok teknologi (*technology block*)

Teknologi merupakan “kotak alat” (*tool box*) dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari 3 bagian utama, yaitu teknisi (*human atau brainware*), perangkat lunak (*software*) dan perangkat keras (*hardware*). Teknisi dapat berupa orang-orang yang mengetahui teknologi dan membuatnya dapat beroperasi.

5. Blok basis data (*database block*)

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut DBMS (*Database Management System*)

6. Blok kendali (*controls block*)

Banyak hal yang dapat merusak sistem informasi, seperti misalnya bencana alam, apai, temperature, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri,kesalahan-kesalahan, ketidak efisienan, sabotase dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

2.1.6. Defenisi Internet

“*Internet* atau *internetwork* adalah sekumpulan jaringan berbeda yang saling terhubung bersama sebagai satu kesatuan dengan menggunakan berbagai macam protokol, salah satunya adalah protokol TCP/IP (*Transmission Control Protocol/Internet Protocol* (Muarie, 2015: 30).

FA adapun beberapa hal yang berkaitan dengan *web internet* adalah sebagai berikut:

1. *Web Browser* Merupakan perangkat lunak yang berfungsi menampilkan dokumen atau informasi *web* yang diambil dari *webserver*. Contoh *web browser* adalah *Internet explorer*, *Mozilla Firefox*, dan *Opera*.
2. *Web Server* Merupakan perangkat lunak yang berfungsi agar dokumen *web* yang disimpan di *server* dapat diakses oleh pemakai *internet* misalnya *XAMPP*.
3. *Situs Web/Homepage* *Situs web* adalah sebutan bagi sekelompok halaman *web* yang menyimpan informasi di *world wideweb*. *Homepage*

merupakan halaman awal dari sebuah situs *web*.

4. *Web Hosting* *Web hosting* adalah bentuk layanan jasa penyewaan tempat di *internet* yang memungkinkan perorangan ataupun organisasi menampilkan layanan jasa atau produknya di situs *web*. *Webhost* merupakan organisasi yang menyediakan tempat di *server*-nya untuk perseorangan atau organisasi meletakkan semua *file website*-nya dan menyediakan konektivitas dengan *internet* agar dapat diakses melalui *internet*.

Internet merupakan suatu jaringan komunikasi tanpa batas yang melibatkan jutaan komputer pribadi yang tersebar di seluruh dunia. Dengan menggunakan protokol *Transmission Control Protocol/ Internet Protocol* (TCP/IP) dan didukung oleh media komunikasi seperti satelit dan paket radio, internet telah memungkinkan komunikasi antar komputer dengan jarak yang tidak terbatas (Muarie, 2015: 30).

Dari penjelasan diatas, *internet* merupakan kumpulan dari beberapa komputer yang terhubung dalam satu jaringan dan dapat diakses dari tempat yang sangat jauh.

2.1.7. Defenisi WWW (World Wide Web)

World Wide Web lebih dikenal sebagai *web* adalah suatu layanan sajian informasi yang menggunakan konsep *hyperlink* (tautan), yang memudahkan *surfer* (sebutan para pemakai komputer yang melakukan *browsing* atau penelusuran informasi melalui internet) (Muarie, 2015: 31). Web terdiri dari dua komponen

dasar yaitu:

1. *Server Web*

Sebuah komputer (*server*) dan *software* yang menyimpan dan mendistribusikan data komputer lainnya melalui jaringan internet.

2. *Browser Web*

Web Browser adalah *software* yang digunakan untuk menampilkan informasi dari *server web*. Saat ini terdapat banyak macam *software* aplikasi browser antara lain *Internet Explorer* dan *Mozilla Firefox*.

Salah satu layanan aplikasi di internet adalah *World Wide Web*, pelayanan yang cukup baru dikembangkan di internet dan menjadi layanan aplikasi yang paling populer digunakan pemakai jaringan internet dan perkembangannya terus dilakukan sampai saat ini untuk menyempurnakan teknologi ini (Muarie, 2015: 31).

2.1.8. Defenisi UML (*Unified Modeling Language*)

UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML hanya berfungsi untuk melakukan pemodelan

UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

Secara fisik, UML adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG (*Object Management Group*). UML terbaru adalah UML 2.3 yang terdiri dari 4 macam spesifikasi, yaitu *Diagram Interchange Specification*, *UML Infrastructure*, *UML Superstructure*, dan *Object Constraint Language (OCL)* (Rosa; & Shalahuddin, 2011: 118).

Penjelasan pembagian kategori dan macam-macam diagram (Rosa; & Shalahuddin, 2011: 121). :

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

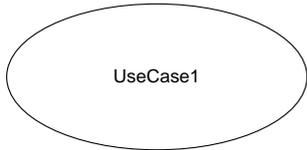
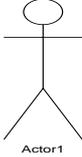
2.1.8.1. Definisi Use Case Diagram

Use Case atau diagram use case merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

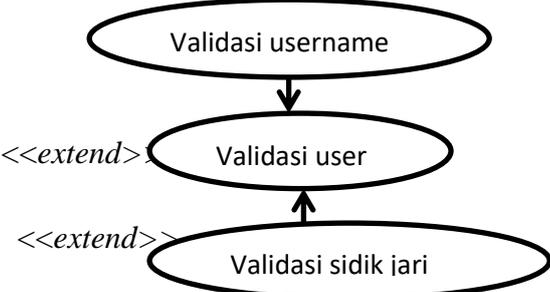
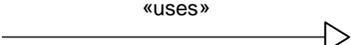
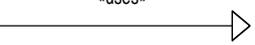
Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut *actor* dan *use case* (Rosa; & Shalahuddin, 2011: 130).

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 symbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang.
2. *Use Case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antarunit atau actor.

Tabel 2. 1 Simbol-simbol use case

SIMBOL	DESKRIPSI
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p>Aktor/<i>actor</i></p>  <p>Nama actor</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi. Biasanya dinyatakan dengan menggunakan kata benda di awal frase nama actor</p>
<p>Asosiasi/<i>association</i></p> 	<p>Komunikasi antar actor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor</p>

Tabel 2. 2 Simbol-simbol use case Lanjutan

<p>Ekstensi/<i>extend</i> <code><<extend>></code> </p>	<p>Relasi <i>use case</i> tambahkan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan;</p>  <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan.</p>
<p>Generalisasi/<i>generalization</i> </p>	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. Arah panah mengarah pada <i>use case</i> yang menjadi generalisasi (umum).</p>
<p>Menggunakan/<i>include/user</i> <code><<include>></code>  <code><<uses>></code> </p>	<p>Relasi <i>use case</i> tambahkan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini Ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i> :</p> <ul style="list-style-type: none"> • Include berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan • Include berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan.

2.1.8.2. Definisi *Class Diagram*

class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem (Rosa; & Shalahuddin, 2011: 122). Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

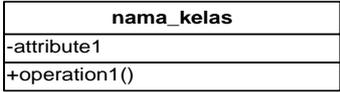
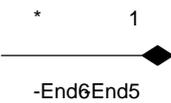
3. Kelas yang diambil dari pendefinisian use case

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian use case.

4. Kelas yang diambil dari pendefinisian data

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Tabel 2. 3 Simbol-simbol class diagram

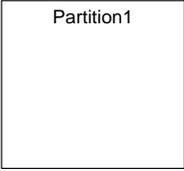
SIMBOL	DESKRIPSI
<p>Kelas</p> 	Kelas pada struktur system
<p>Antarmuka/interface</p> <p>nama_interface </p>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
<p>Asosiasi/association</p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Asosiasi berarah / <i>directed association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
<p>Ketergantungan/dependency</p> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
<p>Agregasi / <i>aggregation</i></p> <p>* 1</p> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

2.1.8.3. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut (Rosa; & Shalahuddin, 2011: 134).

1. Rancangan prose bisnis di mana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktiitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian di mana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

Tabel 2. 4 Simbol-simbol activity diagram

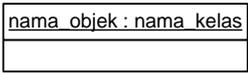
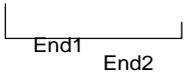
SIMBOL	DESKRIPSI
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan/join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.1.8.4. Definisi *Object Diagram*

Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem (Rosa; & Shalahuddin, 2011: 124).

Berikut ini-simbol-simbol yang ada pada diagram objek:

Tabel 2. 5 Simbol-simbol Diagram Objek

SIMBOL	DESKRIPSI
Objek 	Objek dari kelas yang berjalan saat sistem dijalankan
Link 	Relasi antar objek

2.1.8.5. Definisi *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan jumlah contoh objek dan message (pesan) yang diletakkan diantara objek-objek di dalam use case (Nasril: & Saputra, 2016: 49).

2.1.8.6. Definisi *Communication Diagram*

Diagram Komunikasi menggambarkan interaksi antarobjek/bagian dalam bentuk urutan pengiriman pesan. Diagram komunikasi mengelompokkan *message* pada kumpulan diagram sekuen menjadi sebuah diagram. Dalam diagram

komunikasi yang dituliskan adalah operasi/metode yang dijalankan antara objek yang satu dan objek lainnya secara keseluruhan, oleh karena itu dapat diambil dari jalannya interaksi pada semua diagram sekuen. Penomoran metode dapat dilakukan berdasarkan urutan dijalankannya (Rosa; & Shalahuddin, 2011: 140).

2.2. Tinjauan Teori Khusus

Memuat dan menjelaskan teori khusus yang berhubungan dengan objek pembahasan sesuai dengan judul penelitian skripsi

2.2.1. Defenisi Ujian

Pengertian Ujian sebagai berikut: “Ujian adalah cara (yang dapat dipergunakan) atau prosedur (yang perlu ditempuh) dalam rangka pengukuran dan penilaian di bidang pendidikan, yang berbentuk pemberian tugas atau serangkaian tugas baik berupa pertanyaan– pertanyaan (yang harus dijawab), atau perintah-perintah (yang harus dikerjakan) oleh testee, sehingga (atas dasar data yang diperoleh dari hasil pengukuran tersebut) dapat menghasilkan nilai yang melambangkan tingkah laku atau prestasi testee nilai yang mana dapat dibandingkan dengan nilai-nilai yang dicapai oleh testee. (Nasril: & Saputra, 2016).

2.2.2. Defenisi *Online*

Secara umum, sesuatu dikatakan online adalah bila ia terkoneksi/terhubung dalam suatu jaringan ataupun sistem yang lebih besar. Beberapa arti kata online lainnya yang lebih spesifik yaitu :

1. Dalam percakapan umum, jaringan/network yang lebih besar dalam konteks ini biasanya lebih mengarah pada internet, sehingga 'online' lebih pada menjelaskan status bahwa ia dapat diakses melalui internet.
2. Secara lebih spesifik dalam sebuah sistem yang terkait pada ukuran dalam satu aktivitas tertentu, sebuah elemen dari sistem tersebut dikatakan online jika elemen tersebut beroperasi. Sebagai contoh, Sebuah instalasi pembangkit listrik dikatakan online jika ia dapat menyediakan listrik pada jaringan elektrik.
3. Dalam telekomunikasi, Istilah online memiliki arti lain yang lebih spesifik. Suatu alat diasosiasikan dalam sebuah sistem yang lebih besar dikatakan online bila berada dalam kontrol langsung dari sistem tersebut. Dalam arti jika ia tersedia saat akan digunakan oleh sistem (on-demand), tanpa membutuhkan intervensi manusia, namun tidak bisa beroperasi secara mandiri di luar dari sistem tersebut.

Dengan Internet kita dapat menerima dan mengakses informasi dalam berbagai format dari seluruh penjuru dunia. Kehadiran internet juga dapat memberikan kemudahan dalam dunia pendidikan, hal ini terlihat dengan begitu banyaknya situs web yang menyediakan media pembelajaran yang semakin interaktif serta mudah untuk dipelajari (Dewi & Cahyadi, 2014: 121).

2.2.3. Defenisi PHP

Jika diartikan PHP memiliki beberapa pandangan dalam mengartikannya, akan tetapi kurang lebih PHP dapat kita ambil arti sebagai PHP : Hypertext Preprocessor. Ini merupakan bahasa yang hanya dapat berjalan pada server dan hasilnya dapat di tampilkan pada client. PHP singkatan dari PHP Hypertext Preprocessor yang digunakan sebagai bahasa skrip serverside dalam pengembangan web yang dapat disisipkan pada dokumen HTML. Ketika seseorang mengunjungi web berbasis PHP, webserver akan memproses kode-kode PHP. Beberapa perintah atau kode dari PHP tersebut selanjutnya ada yang diterjemahkan ke dalam HTML dan beberapa ada yang disembunyikan (misalnya proses operasi). Setelah diterjemahkan ke dalam HTML, webserver akan mengirim kembali ke webbrowser pengunjung tersebut (Muarie, 2015: 32).

PHP atau kependekan dari Hypertext Preprocessor adalah salah satu bahasa pemrograman open source yang sangat cocok atau dikhususkan untuk pengembangan Web dan dapat di tanamkan pada sebuah skrip HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, Java, dan Perl serta mudah untuk dipelajari. PHP diciptakan untuk mempermudah pengembang web dalam menulis halaman web dinamis dengan cepat, bahkan lebih dari itu kita dapat mengeksplorasi hal-hal yang luar biasa dengan PHP. Sehingga dengan demikian PHP sangat cocok untuk para pemula, menengah maupun expert sekalipun (Djaelangara, Sengkey, & Lantang, 2015: 87).

2.2.4. Defenisi HTML

Hypertext Markup Language (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web internet dan pemformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format ASCII normal sehingga menjadi halaman web dengan perintah-perintah HTML (Djaelangkara et al., 2015: 87).

merupakan bahasa yang digunakan untuk menulis halaman web. Biasanya mempunyai ekstensi .htm, .html, atau .shtml. HTML tersusun atas tag-tag, digunakan untuk menentukan tampilan dari dokumen HTML yang diterjemahkan oleh browser. Tag HTML tidak case sensitive. Jadi bisa menggunakan <HTML> atau <html>. Keduanya menghasilkan output yang sama (Muarie, 2015: 31).

2.2.5 Defenisi JavaScript

JavaScript adalah kode untuk menyusun halaman *web* yang memungkinkan dijalankan disisi klien. Karena dijalankan disisi klien, maka *JavaScript* dapat digunakan untuk membuat tampilan lebih bersifat dinamis. Jadi, *JavaScript* adalah bahasa skrip yang diproses pada sisi klien (Muarie, 2015: 33).

“*Javascript* adalah suatu bahasa pemograman yang dikembangkan untuk dapat berjalan pada web browser” (Nasril: & Saputra, 2016: 50).

2.2.6. Defenisi CSS (*Cascading Style Sheets*)

CSS (Cascading Style Sheet) merupakan salah satu bahasa pemrograman *web* untuk mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam. Pada umumnya *CSS* dipakai untuk memformat tampilan halaman *web* yang dibuat dengan bahasa *HTML* dan *XHTML*. *CSS* dapat mengendalikan ukuran gambar, warna bagian tubuh teks, warna tabel, ukuran border, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya (Muarie, 2015: 33).

Cascading Style Sheet (*CSS*) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. *CSS* bukan merupakan bahasa pemograman. Pada umumnya *CSS* dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa *HTML* dan *XHTML* (Djaelangkara et al., 2015: 88).

2.2.5. Definisi *Flowchart*

”*Flowchart* adalah gambaran secara fisik atau bagian yang memperhatikan urutan dan hubungan antar proses beserta instruksinya dan merupakan bagian dari alur dokumen, suatu *flowchart* dapat menggambarkan secara global *system*” (Nasril: & Saputra, 2016: 49).

2.2.6. Definisi XAMPP

“XAMPP (*X Apache MySQL PHP Perl*) merupakan paket PHP dan MySQL berbasis *open source* yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis PHP” (Nasril: & Saputra, 2016: 49).

2.2.7. Definisi Bootstrap

“Bootstrap adalah framework ataupun *tools* untuk membuat aplikasi web ataupun website yang bersifat *responsive* secara cepat, mudah, dan gratis”. (Nasril: & Saputra, 2016: 50).

2.2.8. Definisi MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya, SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukkan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. (Djaelangkara et al., 2015: 8).

2.2.9. *Extreme Programming (XP)*

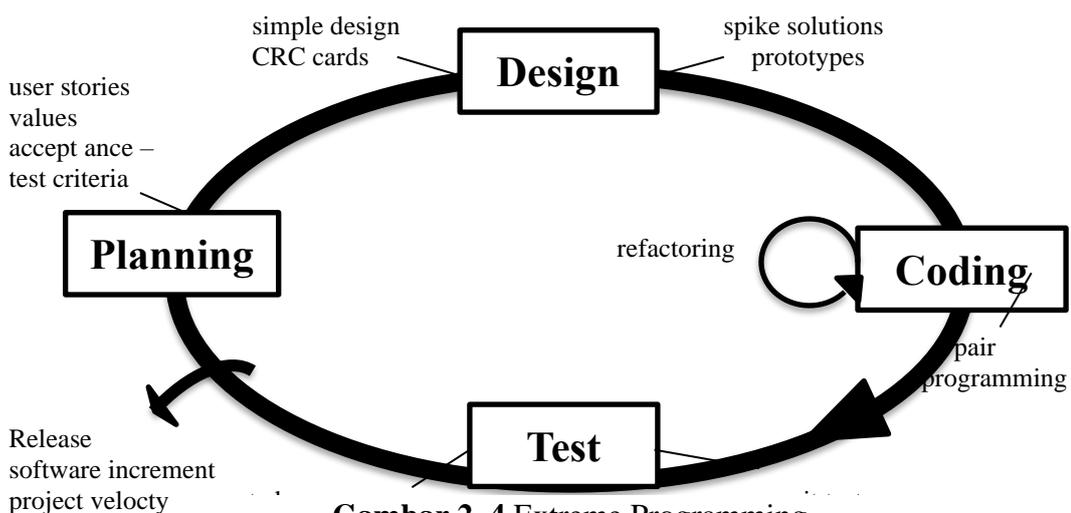
Metode Extreme Programming sering juga dikenal dengan metode XP. Metode ini dicetuskan oleh Kent Beck, seorang pakar software engineering. Extreme programming adalah model pengembangan perangkat lunak yang menyederhanakan berbagai tahapan pengembangan sistem menjadi lebih efisien, adaptif dan fleksibel (Fatoni & Dwi, 2016: 17). Nilai dasar metode extreme programming :

1. Communication : Memfokuskan komunikasi yang baik antara programmer dengan user maupun antar programmer.
2. Courage : Pengembang perangkat lunak harus selalu memiliki keyakinan, keberanian dan integritas dalam melakukan tugasnya.
3. Simplicity : Lakukan semua dengan sederhana.
4. Feedback : Mengandalkan feedback sehingga dibutuhkan anggota tim yang berkualitas.
5. Quality Work : Proses berkualitas berimplikasi pada perangkat lunak yang berkualitas sebagai hasil akhirnya.

Extreme Programming (XP) merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium serta metode ini juga sesuai jika tim dihadapkan dengan requirement yang tidak jelas maupun terjadi perubahan – perubahan requirement yang sangat cepat (Prabowo & Artwodini, 2013: 477).

Untuk mencapai kesederhanaan, XP membatasi pengembang perangkat lunak melakukan perancangan hanya untuk kebutuhan-kebutuhan yang sifatnya mendesak alih-alih melakukan perancangan kebutuhan-kebutuhan yang diperlukan di masa depan. Tujuannya adalah untuk menciptakan rancangan yang sederhana yang dapat dengan mudah diimplementasikan dalam bentuk kode-kode program secara cepat. jika rancangan tersebut selanjutnya harus ditingkatkan, rancangan yang bersangkutan dapat di-refaktorisasi di waktu yang lain (Korowotjeng, Sengkey, Paturusi, Tuturoong, & Kom, 2014: 4).

Pemrograman Ekstreme menggunakan suatu pendekatan 'berorientasi objek' sebagai paradigma pengembangan yang diinginkan dan mencakup di dalamnya seperangkat aturan dan praktik-praktik yang terjadi dalam konteks empat kegiatan kerangka kerja: perencanaan, perancangan, pengkodean, dan pengujian (Korowotjeng et al., 2014: 4).



Gambar 2. 4 Extreme Programming

Kunci kegiatan XP diringkas dalam paragraf-paragraf berikut :

1. Perencanaan Kegiatan

Perencanaan biasanya dimulai dengan mendengarkan suatu kegiatan yang bertujuan untuk mengumpulkan kebutuhan-kebutuhan yang memungkinkan anggota teknis tim XP memahami konteks bisnis untuk perangkat lunak yang akan dikembangkan dan untuk merasakan perlunya output, fitur-fitur utama, dan fungsionalitas. Aktivitas-aktivitas mendengarkan pada dasarnya mengarah kepada pembuatan serangkaian "cerita" (juga disebut user stories) yang menggambarkan keluaran yang diperlukan, fitur-fitur, dan fungsionalitas-fungsionalitas yang akan dibangun menggunakan perangkat lunak yang akan dikembangkan. Setiap cerita umumnya ditulis oleh para pelanggan dan diletakkan pada kartu indeks. pelanggan-pelanggan memberikan suatu nilai (yaitu, suatu prioritas) pada cerita-cerita tertentu berdasarkan seluruh nilai bisnis dari fitur.

atau Fungsi Anggota tim XP kemudian menilai setiap cerita dan menetapkan biayanya yang diukur dalam bentuk minggu-minggu yang diperlukan untuk melakukan pengembangan cerita tersebut. jika cerita ini diperkirakan membutuhkan lebih dari tiga minggu pengembangan, pelanggan akan diminta untuk membagi cerita tersebut ke dalam cerita-cerita kecil dan perhitungan-perhitungan nilai dan biaya terjadi lagi. Penting untuk diperhatikan bahwa cerita baru dapat ditulis setiap saat.

Pelanggan dan pengembang kemudian bekerja sama untuk memutuskan bagaimana mengelompokkan cerita kedalam kelanjutannya untuk dikembangkan oleh tim XP. Ketika komitmen dasar (kecocokan atas cerita yang dimasukkan, tanggal pengiriman, dan hal-hal lainnya yang terkait dengan proyek perangkat lunak) ini dirilis, tim XP mengatur cerita yang akan dikembangkan dalam satu dari tiga cara : (1) semua cerita akan segera dilaksanakan, (2) cerita yang memiliki nilai bisnis tertinggi akan dipindahkan ke dalam jadwal dan dilaksanakan pertama kali, atau (3) cerita yang paling berisiko akan dijadwalkan dan akan dilaksanakan pertama kali.

Setelah proyek pertama telah dijalankan, tim XP selanjutnya akan menghitung kecepatan proyek. Sederhananya, kecepatan proyek merupakan jumlah cerita pelanggan yang diimplementasikan selama rilis yang pertama. Kecepatan proyek kemudian dapat digunakan untuk (1) membantu memperkirakan tanggal pengiriman ke para pelanggan dan membantu memperkirakan jadwal rilis berikutnya dan (2) menentukan apakah terdapat komitmen yang berlebihan untuk semua cerita di seluruh proyek pengembangan perangkat lunak. Jika hal tersebut terjadi, maka isi rilis dimodifikasi atau tanggal pengiriman akhir diubah. Ketika kerja pengembangan perangkat lunak cepat berjalan, para pelanggan dapat saja menambahkan cerita, mengubah nilai sebuah cerita yang ada, memilah-milah cerita, atau menghilangkannya. Tim XP kemudian mempertimbangkan semua sisa rilis dan memodifikasi rencana.

2. Perancangan

Perancangan XP dengan ketat mengikuti prinsip "tetap sederhana". Sebuah hasil perancangan yang sederhana selalu lebih disukai daripada gambar-gambaran yang lebih kompleks. Selain itu, perancangan XP akan memberikan panduan implementasi untuk suatu cerita ketika cerita itu ditulis, tidak kurang, tidak lebih. Rancangan-rancangan dan fungsionalitas-fungsionalitas tambahan (karena pengembang menganggap nantinya akan diperlukan) tidak terlalu disarankan.

XP mendorong penggunaan kartu CRC sebagai mekanisme yang efektif untuk berpikir tentang perangkat lunak dalam konteks berorientasi objek. Kartu CRC (class-responsibility- collaborator) digunakan untuk mengidentifikasi dan mengatur kelas-kelas dalam konteks "pemrograman berorientasi objek" yang relevan dengan peningkatan perangkat lunak saat ini. Tim XP selanjutnya akan melakukan latihan perancangan. Kartu CRC adalah satu-satunya produk kerja perancangan yang dihasilkan sebagai bagian dari proses pengembangan perangkat lunak cepat: XP.

Jika masalah perancangan yang sulit ditemui sebagai bagian dari perancangan suatu cerita, metode pengembangan cepat PX menyarankan pembuatan langsung dari prototipe operasional dari bagian perancangan tersebut. Disebut sebagai solusi spike, prototipe perancangan diimplementasikan dan selanjutnya dievaluasi. Tujuannya adalah untuk mengurangi risiko-risiko yang akan timbul ketika pelaksanaan yang

sesungguhnya dimulai dan untuk memvalidasi keaslian perkiraan untuk cerita yang berisi masalah-masalah perancangan.

Refaktorisasi pada dasarnya adalah proses mengubah sistem perangkat lunak sedemikian rupa sehingga tidak mengubah perilaku eksternal kode, namun memperbaiki struktur internal yang ada di dalamnya. Ini merupakan cara yang disiplin untuk membersihkan kode dan memodifikasi/menyederhanakan rancangan internal, yang pada gilirannya akan meminimalkan kemungkinan akan munculnya kesalahan-kesalahan program. Pada intinya, ketika Anda melakukan refactor atau refaktorisasi, Anda pada prinsipnya memperbaiki rancangan kode-kode program Setelah kode-kode program itu ditulis. Karena perancangan perangkat lunak pada rekayasa perangkat lunak cepat (XP) hampir-hampir tidak menggunakan notasi dan, walaupun ada, hanya menghasilkan beberapa saja, tidak menghasilkan produk-produk kerja selain kartu-kartu CRC dan solusi-solusi spike, maka perancangan pada XP dipandang sebagai buatan sementara yang dapat dan harus terus menerus diubah wiring majunya pelaksanaan konstruksi perangkat lunak. Tujuan dari refaktorisasi pada dasarnya adalah untuk mengendalikan modifikasi-modifikasi yang perlu dilakukan dengan cara menyarankan perubahan perancangan kecil yang "secara radikal dapat meningkatkan kualitas rancangan". Tetapi, dalam hal ini perlu diperhatikan bahwa upaya yang diperlukan untuk melakukan refaktorisasi dapat tumbuh secara drastic ketika ukuran aplikasi bertambah. Sebuah gagasan central dalam XP

adalah rancangan yang terjadi baik sebelum dan sesudah pengkodean dimulai. Refaktorisasi berarti bahwa perancangan perangkat lunak terjadi terusmenerus ketika sistem/perangkat lunak dikonstruksi. Bahkan, kegiatan konstruksi itu sendiri akan memberikan tim XP pedoman bagaimana caranya meningkatkan kualitas rancangan.

3. Pengkodean.

Setelah cerita dikembangkan dan karya rancangan awal dilakukan, tim perangkat lunak cepat (XP) tidak langsung beralih ke kode-kode program, tetapi lebih dulu akan mengembangkan serangkaian unit pengujian yang akan menjalankan setiap cerita yang akan disertakan pada rilis yang ada (peningkatan perangkat lunak)."

Setelah unit pengujian dibuat, pengembang perangkat lunak akan lebih mampu berkonsentrasi pada apa yang harus diimplementasikan supaya lulus dari unit pengujian tersebut. Tidak ada yang ditambahkan (tetap sederhana). Ketika kode-kode program telah selesai dituliskan, kode-kode program tersebut dapat langsung diuji menggunakan unit pengujian yang telah dirancang sebelumnya, sehingga bisa langsung memberikan umpan balik kepada para pengembang.

Konsep kunci selama kegiatan pengkodean adalah pemrograman berpasangan (pair programming). XP menyarankan bahwa dua orang pemrogram seharusnya bekerja sama pada satu komputer workstation untuk menuliskan kode-kode program untuk suatu cerita. Hal ini menyajikan mekanisme untuk pemecahan masalah yang dapat dilakukan

secara real-time dan jaminan kualitas real-time (kode-kode program langsung ditinjau ketika ia dibuat). Hal ini juga membuat para pengembang perangkat lunak dapat berfokus pada masalah yang dihadapi. Dalam praktiknya, setiap orang mengambil peran yang sedikit berbeda. Misalnya, satu orang mungkin berpikir tentang rincian pengkodean dari bagian tertentu dari suatu perancangan, sementara orang yang lain memastikan standar pengkodean (merupakan bagian yang diharuskan dari XP) diikuti atau kode untuk cerita memenuhi unit pengujian yang telah dikembangkan untuk melakukan validasi kode terhadap cerita.

Ketika pasangan pemrogram tadi berhasil menyelesaikan pekerjaan mereka, kode-kode program yang mereka kembangkan kemudian diintegrasikan dengan hasil pengembangan sebelumnya. Dalam beberapa kasus, hal ini dilakukan biasanya oleh tim integrasi. Dalam kasus lain, pasangan pemrogram tadi sekaligus memiliki tanggung jawab untuk melakukan integrasi. Strategi "integrasi berkesinambungan" membantu untuk menghindari masalah kompatibilitas dan interfacing dan memberikan lingkungan "smoke-testing" yang membantu menemukan kesalahan lebih dini.

4. Pengujian

bahwa pembuatan unit pengujian sebelum pengkodean dimulai merupakan elemen kunci dari pendekatan pengembangan perangkat lunak cepat XP. Unit pengujian yang harus dibuat dan kemudian dijalankan menggunakan kerangka kerja yang memungkinkan mereka untuk

diotomatisasi (sehingga mereka dapat dijalankan dengan mudah dan dapat dijalankan berulang kali). Hal ini mendorong strategi pengujian regresi terhadap kode- kode program setiap kali kode-kode program tersebut diubah (yang seringkali dikaitkan dengan falsafah refactoring XP). Sebagai unit pengujian individu yang diatur dalam "rangkaian pengujian universal", pengujian integrasi dan validasi atas sistem/perangkat lunak pada umumnya dapat terjadi. Hal ini menyajikan bagi tim perangkat lunak XP suatu indikasi kemajuan dan juga dapat memberikan peringatan awal jika terjadi sesuatu yang keliru. Wells menyatakan: "Memperbaiki masalah-masalah kecil setiap beberapa jam membutuhkan waktu lebih sedikit daripada memperbaiki masalah besar sebelum batas waktunya."

Uji kelayakan XP, sering juga disebut uji pelanggan, dirincioleh para pelanggan dan pada dasarnya berfokus pada fitur- fitur dan fungsionalitas-fungsionalitas sistem/perangkat lunak secara keseluruhan yang dapat terlihat dan ditinjau kembali oleh para pelanggan. Uji kelayakan berasal dari cerita pengguna yang telah diimplementasikan sebagai bagian dari suatu rilis perangkat lunak.

2.2.10. Penelitian Terlebih Dahulu

Tabel 2. 6 Penelitian Terdahulu

ISSN	JUDUL	HASIL PENELITIAN
2338-3143	RANCANG BANGUN SISTEM INFORMASI UJIAN <i>ONLINE</i>	Sistem ujian yang sudah dibangun atau dikomputerisasikan dapat mengoreksi hasil ujian dengan cepat sehingga dapat menghemat waktu dalam pengerjaan ujiannya maupun dalam penilaian ujiannya tanpa harus dikoreksi guru.
2407-2192	RANCANG BANGUN SISTEM UJIAN <i>ONLINE</i> PADA SMP NEGERI 8 SEKAYU	Sistem ujian <i>online</i> ini dapat memberikan kemudahan dalam mengetahui data guru, siswa dan nilai hasil belajar siswa.
2085-1588	Implementasi <i>E-learning</i> sebagai <i>Komplemen</i> dan <i>Blanded Learning</i> Untuk Meningkatkan Motivasi dan Hasil Belajar Pada Matakuliah <i>Enterprise Resources Planning</i>	materi ajar dengan animasi dan video animasi interaktif yang telah digunakan mahasiswa dalam proses pembelajaran mandiri (<i>blanded learning</i>). Hasil pembelajaran mandiri mahasiswa tersebut dapat meningkatkan motivasi, kualitas serta perluasan materi belajar (<i>komplemen learning</i>) mahasiswa terhadap matakuliah ERP.
1978 - 8282	SISTEM UJIAN CALON MAHASISWA BARU BERBASIS ILEARNING EDUCATION MARKETING PADA PERGURUAN TINGGI	Sistem yang dibangun yaitu sistem bersifat online dan terpusat, proses untuk melakukan USM (Ujian Saringan Masuk) bisa secara insite maupun offsite, presenter tidak perlu lagi mengoreksi satu persatu dari hasil USM, nilai USM otomatis akan masuk dalam gradebook. Selain itu memudahkan CMB (Calon Mahasiswa Baru)) untuk mengikuti USM jika berada di luar daerah maupun kota

Tabel 2.7 Penelitian Terdahulu Lanjutan

2087-1224	<i>QR CODE READER PADA SMARTPHONE ANDROID PADA LAYANAN RESTORAN.</i>	Dengan menggunakan sistem yang terintegrasi antara aplikasi pada smartphome dengan aplikasi backend proses operasional restoran dalam mengelola data menu, karyawan, dan transaksi menjadi lebih baik dan rapi
2337-3539	Rancang Bangun Aplikasi Web Informasi Eksekutif Pada Pemerintah Kabupaten XYZ	Aplikasi web informasi eksekutif ini merupakan sebuah aplikasi yang memiliki fasilitas untuk dapat menampilkan tampilan grafik untuk data – data yang dihasilkan dari laporan pertanggung jawaban pada suatu kabupaten
2301-8402	Sistem Informasi Pengarsipan Berbasis Web Di Kantor Jurusan Teknik Elektro Unsrat	Hasil penelitian menunjukkan bahwa manajemen arsip digital mampu memberikan kemudahan dalam proses pencarian arsip serta memberikan kemudahan dalam pengelolaan dan penyimpanan surat masuk dan keluar, selain itu arsip digital juga mampu mengurangi biaya untuk perawatan dan penyimpanan karena arsip sudah didigitalisasi atau disimpan dalam bentuk digital