

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.

2.1.1. Sistem

Definisi sistem adalah kumpulan dari bagian-bagian yang bekerja sama untuk mencapai tujuan yang sama atau sekumpulan objek-objek yang saling berelasi dan berinteraksi dan informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi si penerima dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang (Akbar & Dahlan, 2013: p.39).

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: p.321).

Sistem berasal dari bahasa latin (*systema*) dan bahasa yunani (*sustema*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Istilah ini sering

dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat (Husda, 2012: p.111).

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 (Husda, 2012: p.112) . :

1. Komponen (*component*)

Suatu sistem terdiri dari jumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. 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.

3. 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 energi 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.

4. 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.

5. 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.

6. Keluaran Sistem (*output*)

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

yang tidak berguna dan merupakan hasil sisa pembuangan, sedang informasi adalah keluaran yang dibutuhkan.

7. 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.

8. 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. Informasi

Informasi (*information*) adalah data yang telah diolah menjadi suatu bentuk yang penting bagi si 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: p.10).

Secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan. Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah sesuatu yang terjadi pada saat tertentu (Husda, 2012: p.117).

2.1.4. Konsep Dasar Informasi

Definisi umum untuk istilah informasi dalam pemakaian sistem informasi adalah data yang telah diolah menjadi suatu bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau masa mendatang. Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal atau data item. Data adalah kenyataan yang menggambarkan suatu kejadian dan kesatuan nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat tertentu (Ibrahim, 2011: p.322).

Informasi dikatakan bermutu atau berkualitas jika memenuhi kriteria berikut (Ibrahim, 2011: p.322):

1. Keakuratan (*accuracy*), artinya harus bebas dari kesalahan dan tidak bias atau menyesatkan, serta mencerminkan maksud yang sesungguhnya.
2. Ketepatan waktu (*timeliness*), artinya informasi yang diterima tidak boleh terlambat karena informasi yang telah usang tidak akan bernilai lagi.

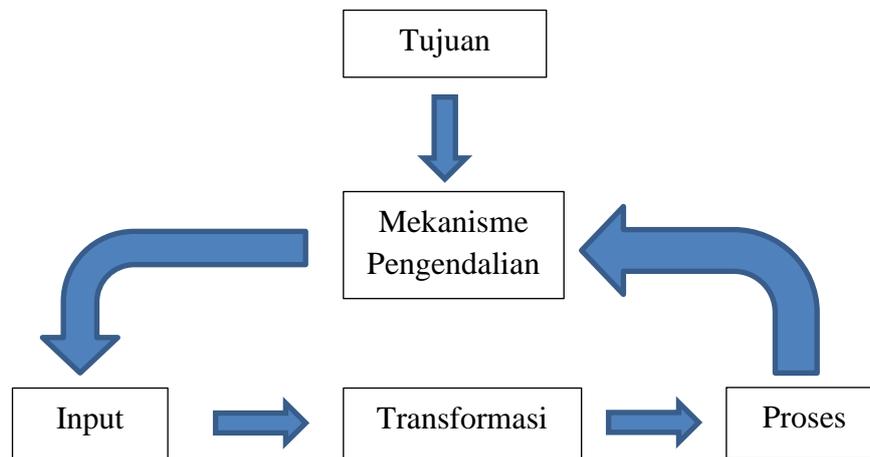
3. Relevan (*relevance*), artinya informasi tersebut bermanfaat bagi penerima atau pemakaiannya.
4. Tambahan, informasi harus dapat mempengaruhi atau memberikan tambahan baru pada informasi yang telah ada.
5. Korektif, informasi harus dapat menjadi suatu koreksi atas informasi yang salah atau palsu sebelumnya.
6. Penegasan, informasi harus dapat mempertegas informasi yang telah ada. Hal ini penting karena meningkatkan persepsi penerimaan atas kebenaran informasi tersebut.

Didalam suatu organisasi atau perusahaan, informasi merupakan sesuatu yang memiliki arti yang sangat penting didalam mendukung proses pengambilan keputusan oleh pihak manajemen. Secara umum informasi dapat didefinisikan sebagai data yang diolah menjadi bentuk yang lebih berarti bagi yang menerima (Husda, 2012: p.117).

2.1.5. Sistem Informasi

Sistem informasi adalah gabungan yang terorganisasi dari manusia, perangkat lunak, perangkat keras, jaringan komunikasi dan sumber data dalam mengumpulkan, mengubah, dan menyebarkan informasi dalam organisasi. Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan

pihak luar tertentu dengan laporan-laporan yang diperlukan (Akbar & Dahlan, 2013: p.40).



Gambar 2. 1 Sistem Informasi (Akbar & Dahlan, 2013: p.40)

Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (SeptaviaRd.Erwin, IntanGunadhi, & Kurniawati, 2015: p.2).

Setelah menguraikan definisi dari sistem informasi maka secara umum sistem informasi dapat diartikan sebagai kumpulan dari berbagai sub sistem, baik lainnya yang berjalan secara harmonis untuk mencapai tujuan yang sama yaitu mengumpulkan dan mengelola data menjadi suatu informasi yang bermanfaat dan memiliki sebuah nilai, komponen-komponen sistem informasi menurut (SeptaviaRd.Erwin, IntanGunadhi, & Kurniawati, p.2015). terdiri dari :

1. Hardware yaitu kumpulan dari perangkat lunak yang terlihat memungkinkan dapat membentuk sistem seperti computer, printer dan jaringan.
2. Software adalah kumpulan dari perintah-perintah atau fungsi yang ditulis dengan aturan tertentu, memerintahkan agar melaksanakan fungsi tertentu.
3. Data merupakan bahan dasar dari suatu informasi berupa fakta yang mengangkat kejadian-kejadian nyata dan dituangkan kedalam suatu symbol.
4. Prosedur adalah suatu tahapan yang berupa urutan kegiatan yang saling berhubungan untuk mencapai tujuan yang berupa suatu dokumen prosedur seperti : buku petunjuk operasional dan teknis.
5. Manusia merupakan pelaksana dari suatu sistem informasi seperti: Operator, Programmer, Analyst, Designer dan sebagainya.

2.1.6. Komponen Sistem Informasi

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*).

Keenam komponen ini harus ada bersama-sama dan membentuk satu kesatuan. Jika satu atau lebih komponen tersebut tidak ada, maka sistem informasi tidak akan dapat melakukan fungsinya (Husda, 2012: p.120).

Berikut komponen-komponen sistem informasi yang bakal dijelaskan dibawah ini:

1. Blok masukan (*input block*)

Input merupakan data yang masuk ke dalam 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*)

merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Blok kendali (*controls block*)

Beberapa pengendalian perlu dirancang dan diterapkan untuk menanggulangi gangguan-gangguan terhadap sistem.

2.1.7. 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: p.30).

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 wide web*. *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: p.30).

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

2.1.8. Web

Web atau lengkapnya *www* (*world wide web*) adalah sebuah koleksi keterhubungan dokumen–dokumen yang disimpan di internet dan diakses

menggunakan protocol (HTTP / *HyperText Transfer Protocol*).

Intinya bahwa pengguna internet bisa memanfaatkan berbagai macam fasilitas informasi dengan biaya murah tanpa arus datang secara langsung ke tempatnya. Informasi atau dokumen yang diakses dapat berupa data teks, gambar, animasi, video, suara, atau kombinasi diantaranya dan bahkan komunikasi bisa dilakukan secara langsung dengan suara dan video sekaligus (Khoirunnisa, Isnanto, & Martono, 2016: p.93).

Website atau situs web adalah setiap komputer atau tempat dalam sebuah komputer yang terhubung dengan internet dan menjalankan fungsi dan proses sebagai *server web* yang berisi dokumen-dokumen dalam format HTML. Dengan browser, para pengguna computer dapat mencari dan menelusuri (*browse*) serta melihat isi dari dokumen web dan berpindah dari sebuah tempat (halaman) ke tempat lain di web. Contoh program *browser* yang populer misalnya Internet Explorer, Netscape, Opera, Mozilla, dan lain-lain (Husda, 2012: p.166).

2.1.9. 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: p.118).

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

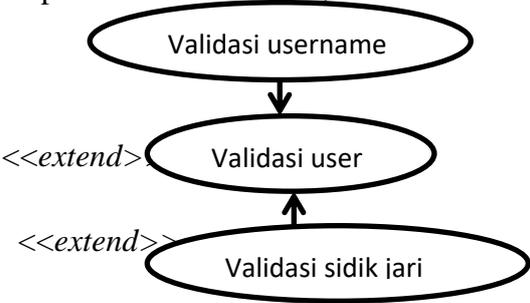
2.1.9.1. *Use Case Diagram*

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.

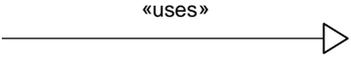
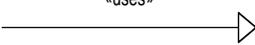
Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut *actor* dan *use case* (Rosa; & Shalahuddin, 2011: p.130).

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>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>
<p>Ekstensi/<i>extend</i></p> <p><<extend>></p> 	<p>Relasi <i>use case</i> tambahan 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>

Tabel 2. 2 Tabel Lanjutan Simbol-simbol use case

<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p> <p>Arah panah mengarah pada use case yang menjadi generalisasi (umum).</p>
<p>Menggunakan/<i>include/user</i></p> <p><<include>></p>  <p><<uses>></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 use case ini</p> <p>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.

2.1.9.2. *Class Diagram*

class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi (Rosa; & Shalahuddin, 2011: p.122).

1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

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.

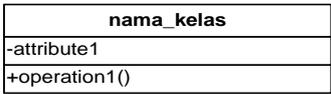
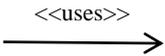
3. Kelas yang diambil dari pendefinisian use case

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

2.1.9.3. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem.(Rosa & Shalahuddin, 2011: p.134).

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut

1. Rancangan proses 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.

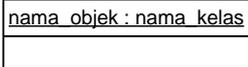
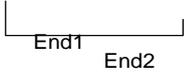
2.1.9.4. *Object Diagram*

Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Pada diagram objek harus dipastikan semua kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggung jawabkan

Untuk apa mendefinisikan sebuah kelas sedangkan pada jalannya sistem, objeknya tidak pernah dipakai. Hubungan *link* pada diagram objek merupakan hubungan memakai dan dipakai dimana dua buah objek akan dihubungkan oleh link jika ada objek yang dipakai oleh objek lainnya (Rosa; & Shalahuddin, 2011: p.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

2.1.9.5. *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: p.49).

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendiskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan *diagram* sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu (Rosa; & Shalahuddin, 2011: p.137).

2.1.9.6. *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 dijelankannya (Rosa; & Shalahuddin, 2011: p.140).

2.2. Tinjauan Teori Khusus

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

2.2.1. *Laundry*

Usaha *laundry* adalah suatu jenis usaha yang bergerak di bidang jasa cuci dan setrika. Bisnis ini termasuk dalam kategori bisnis dengan perputaran yang cepat. Maksudnya rentang waktu permintaan pelanggan antara permintaan pertama dan permintaan selanjutnya pada jasa ini yang memakan waktu relatif singkat.

Lebih jelasnya, pelanggan akan kembali menggunakan jasa ini ketika pakaian yang dikenakan sudah kotor. Selain itu bisnis *laundry* juga merupakan kategori bisnis yang berkelanjutan atau sebagai bisnis yang akan selalu dibutuhkan banyak orang. Manusia akan selalu mencuci pakaian selama kebutuhan sandang masih menjadi kebutuhan primer bagi manusia. Hal inilah yang membuat peluang bisnis *laundry* sangat menjanjikan (Setiyawati, Isnanto, & Martono, 2016: p.151)

Laundry Secara Umum mendefenisikan laundry ialah sebuah departemen housekeeping yang bertugas dan bertanggung jawab untuk memproses semua aktivitas pencucian baik untuk operasional hotel dan tamu hotel (Syawal & Yaqin, 2016: p.1).

2.2.2. PHP

PHP adalah bahasa skrip yang kuat yang dirancang untuk memungkinkan pengembang untuk membuat aplikasi Web yang sangat cepat dan fitur MySQL adalah yang cepat, basisdata handal yang terintegrasi dengan baik dengan PHP dan cocok untuk aplikasi berbasis internet yang dinamis (Khoirunnisa et al., 2016: p.94).

Jika diartikan PHP memiliki beberapa pandangan dalam mengartikannya, akan tetapi kurang lebih PHP dapat kita ambil arti sebagai PHP : Hypertext Preeprocessor. 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: p.32).

2.2.3. HTML

Hypertext Markup Language (HTML) adalah bahasa dari *World Wide Web* (www) yang dipergunakan untuk menyusun dan membentuk dokumen agar dapat

ditampilkan pada program browser. Tiap kali kita mengakses dokumen Web, maka sesungguhnya kita mengakses dokumen seseorang yang ditulis dengan menggunakan format HTML. Jadi HTML itu sendiri merupakan protokol yang digunakan untuk mentransfer data atau dokumen dari Web server ke browser (Khoirunnisa et al., 2016: p.94).

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 (Djaelangara et al., 2015: p.87).

2.2.4. 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: p.49).

Flowchart merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Gambaran ini dinyatakan dengan simbol dan setiap simbol menggambarkan proses tertentu. Sedangkan hubungan antara proses digambarkan dengan garis penghubung (Habibie et al., 2014: p.79).

2.2.5. 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: p.49).

XAMPP merupakan perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa *program*. XAMPP merupakan *tool* yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan menginstal XAMPP maka tidak perlu lagi melakukan *instalasi* dan *konfigurasi* web server Apache, PHP dan MySQL secara manual. XAMPP akan menginstalasi dan mengkonfigurasikannya secara *otomatis* untuk anda atau *auto konfigurasi* (Septa, Hamzah, & Andayati, 2014: p.27).

2.2.6. Notepad++

Notepad++ adalah sebuah aplikasi untuk menuliskan kode atau script web. Notepad++ memiliki nomor urut barisan kode program sehingga memudahkan anda mencari baris kesalahan program. Selain itu, Notepad++ juga dapat membagi antara script html, php yang berupa string, maupun numeric (Andi, 2012: p.3).

Untuk mendapatkan perangkat lunak Notepad++ download++ download melalui URL <http://notepad-plus-plus.org/download/v6.1.3.html>.

2.2.7. MySQL

MySQL adalah sebuah basisdata yang dapat digunakan baik sebagai klien maupun server. MySQL adalah sebuah basisdata server, dapat juga berperan sebagai klien sehingga sering disebut basisdata klien/server, yang terbuka dengan kemampuan dapat berjalan di Sistem Operasi manapun, dengan *Platform* Windows maupun Linux (Setiyawati et al., 2016: p.152).

MySQL merupakan program yang dapat mengakses suatu basisdata MySQL yang berposisi sebagai server. Pada saat itu berarti program kita berposisi sebagai klien. Jadi MySQL adalah sebuah basisdata yang dapat digunakan baik sebagai klien maupun server (Khoirunnisa et al., 2016: p.94).

2.2.8. 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: p.17).

Lima nilai- nilai yang membentuk dasar bagi semua pekerjaan yang dibentuk sebagai bagian dari XP yaitu komunikasi, kesederhanaan, umpan balik, keberanian, dan rasa hormat. Masing- masing nilai yang penting ini digunakan sebagai pendorong untuk kegiatan, tindakan, dan tugas XP yang spesifik. Untuk mencapai komunikasi yang efektif yang seharusnya terjadi antara rekayasawan

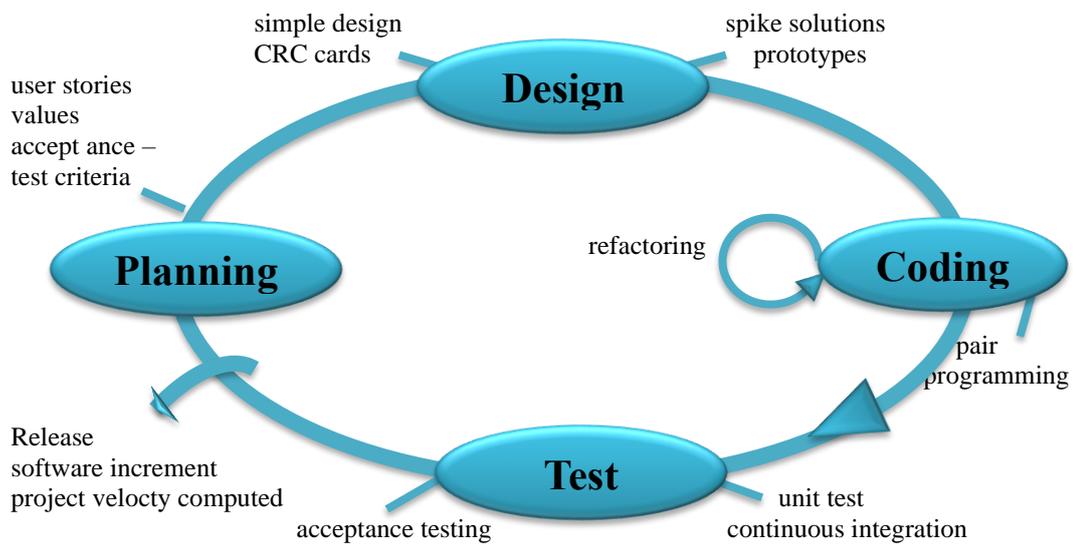
perangkat lunak dan para stakeholder lainnya (contoh: untuk membangun fitur-fitur dan fungsi-fungsi tertentu yang dibutuhkan bagi perangkat lunak), , menekankan pentingnya adaptasi terhadap umpan balik yang berkesinambungan, dan menekankan pentingnya dokumentasi yang produktif sebagai suatu media komunikasi (Korowotjeng, Sengkey, Paturusi, & Tuturoong, 2014: p.4).

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: p.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, 2014: p.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: p.4).



Gambar 2. 2 Proses XP (Korowotjeng et al., 2014: p.4)

Kunci kegiatan XP diringkas dalam paragraf-paragraf berikut (Korowotjeng et al., 2014: p.4) :

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.

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 sate 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.

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.

2.2.9. Penelitian Terlebih Dahulu

Tabel 2. 6 Penelitian Terdahulu

ISSN	PENULIS / JUDUL	HASIL PENELITIAN
2338-0403	Yenita Dwi Setiyawati, R. Rizal Isnanto , Kurniawan Teguh Martono / Pembuatan Aplikasi Antar-Jemput Laundry Berbasis Web Service pada Platform Android	Aplikasi antar-jemput laundry pada platform Android yang dibuat menggunakan bahasa pemrograman Java dan akses basisdata melalui web service sudah dapat digunakan untuk mendata pakaian yang masuk dan keluar laundry maupun mendata pelanggan yang akan menggunakan jasa laundry secara tepat waktu.
2407-1846	Husnil Kamil, Audiah Duhani / Pembangunan Sistem Informasi Pelayanan Jasa Laundry Berbasis Web Dengan Fitur Mobile Pada 21 Laundry Padang	Dapat menghemat waktu dalam pencarian data transaksi pada saat mengelola data transaksi. Aplikasi mampu menghasilkan <i>output</i> berupa rekap transaksi bulanan, data master yang terbaru, serta penyerahan nota <i>order laundry</i>
2549-0796	Arif Rakhman, Achmad Sutanto / Analisa Sistem Informasi Geografis Tempat Laundry Berbasis Android	Sistem Informasi tempat <i>laundry</i> di Kota Tegal berbasis <i>Android</i> , dapat disimpulkan bahwa: Sulitnya mencari informasi tempat <i>laundry</i> secara tepat dan efisien menjadi latar belakang utama dalam proses pembuatan sistem informasi tempat laundry di Kota Tegal. Media promosi bagi pemilik <i>laundry</i> yang masih menggunakan brosur atau <i>banner</i> yang tidak dapat tersebar secara luas dengan cepat.
2087-1224	Dennise Adrianto, Violitta Yesmaya, Albertus Agung, Davin Timothy Ivander, Elysia, Nico Natalie / <i>Qr Code Reader</i> Pada <i>Smartphone</i>	Dengan menggunakan sistem yang terintegrasi antara aplikasi pada <i>smartphone</i> dengan aplikasi backend proses operasional restoran dalam mengelola data menu, karyawan, dan transaksi menjadi lebih baik dan rapi

Tabel 2. 7 Tabel Lanjutan Penelitian Terdahulu

2302-3805	Muhammad Syawal Ainul Yaqin / Analisis Perancangan Aplikasi Layanan Laundry Berbasis Visual Basic 2010 Dan Sql Server 2000 Pada Gajayana Laundry	Aplikasi laundry sangat membantu para pelaku usaha laundry dalam melakukan proses transaksi dan pencatatan pendapatan yang diperoleh. Penerapan aplikasi laundry dalam pencatatan pendapatan sangat mudah dilakukan dan dapat meminimalisasi kesalahan dalam pencatatan pendapatan
2354-6654	Harisuddin, Hilman Niroha, Muhammad Fadillah, Muhammad Izzuddin Al Qassam, Nadhira Puspa Diamanta, Nourma Dwi Safitri / Sistem Informasi Laundry Berbasis Metode OOAD	Sistem informasi ini diterapkan pada Bibi's Laundry untuk memberikan beberapa layanan seperti pemesanan secara online, komunikasi dengan staff secara online, dan beberapa layanan lain yang dapat memberikan kemudahan bagi pelanggan dan pegawai dari laundry tersebut.
2549-6824	Lydia Salvina Helling / Perancangan Sistem Informasi Pelayanan Pelanggan Pada Citra Laundry Bogor	Sistem Informasi Pelayanan Pelanggan di CITRA LAUNDRY membantu pihak manajemen dalam menangani penyimpanan data yang besar. Data yang besar juga dapat terorganisir dengan baik dalam tabel-tabel yang dihasilkan sehingga dengan mudah dapat diakses secara cepat dan baik. Pihak manajemen juga bisa mendapatkan laporan pelayanan bulanan dengan lebih terinci dan rapi.

Tabel 2. 8 Tabel Lanjutan Penelitian Terdahulu

2406-7733	Ahmad Fatoni, Dhany Dwi, / Rancang Bangun Sistem <i>Extreme Programming</i> Sebagai Metodologi Pengembangan Sistem	<i>Extreme Programming</i> bisa di terapkan pada perusahaan atau organisasi yang mempunyai tim khusus untuk pengembangan suatu sistem. Perusahaan atau organisasi yang ingin menerapkan <i>Extreme Programming</i> diharapkan untuk bisa memahami dan mengerti <i>Extreme Programming practice</i> , dan <i>scrum</i> .
2337-3539	Sonny Ariyanto Prabowo, Sholiq dan Feby Artwodini Muqtadiroh / 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	Immanuel Korowotjeng, Rizal Sengkey, Sary, Paturusi / 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