

BAB II

LANDASAN TEORI

2.1. Sistem Pencatatan

Sistem pencatatan adalah sistem yang digunakan untuk mencatat setiap transaksi yang dilakukan, sistem ini terintegrasi dengan sistem lain pada perusahaan dengan tujuan untuk menunjang pelaksanaan proses bisnis suatu perusahaan secara menyeluruh [12].

2.2. Sistem Penjualan

Definisi dari sistem penjualan, suatu proses melalui prosedur-prosedur yang meliputi urutan kegiatan sejak diterimanya pesanan dari pembeli, pengecekan barang ada atau tidak ada dan diteruskan dengan pengiriman barang disertai dengan pembuatan faktur dan mengadakan pencatatan atas penjualan yang berlaku [13]. Sistem Informasi Penjualan adalah suatu sistem informasi yang mengorganisasikan serangkaian prosedur dan metode yang dirancang untuk menghasilkan, menganalisa, menyebarkan dan memperoleh informasi guna mendukung pengambilan keputusan mengenai penjualan [14].

2.3. Sistem Inventory

Sistem inventory adalah bagian yang disediakan dalam proses yang terdapat dalam suatu perusahaan untuk di produksi, serta barang jadi yang disediakan untuk memenuhi permintaan dari konsumen setiap waktu yang disimpan dan dirawat menurut aturan tertentu dalam keadaan siap pakai dan tersimpan dalam *database* [15].

2.4. Sistem Pembelian

Sistem pembelian adalah suatu sistem yang dilakukan untuk pemenuhan barang yang diperlukan oleh perusahaan Fungsi yang terkait dengan akuntansi pembelian, terdiri dari fungsi gudang, fungsi pembelian, serta fungsi akuntansi [16].

2.5. Agile Method

Menurut [17] dalam proses metodologi *agile* interaksi dan tim lebih diutamakan dari pada proses dan tools, software berfungsi lebih utama daripada dokumentasi, kerjasama dengan klien lebih utama dari pada negosiasi kontrak, dan sikap cepat tanggap dengan perubahan lebih utama dibandingkan mengikuti rencana. Metodologi *agile* juga bisa diartikan sama seperti metodologi pengembangan software lainnya yang berdasarkan prinsip-prinsip yang serupa atau pengembangan sistem yang memerlukan waktu singkat dengan adaptasi yang cepat. Metodologi pengembangan *agile* juga melihat pentingnya komunikasi antara anggota tim, anggota teknis dan klien, antara developer dan managernya. Ciri lain dari *agile* metodologi adalah klien menjadi bagian dari tim pengembangan software.

Berdasarkan Agile Alliance, 12 prinsip ini berguna untuk mereka yang menerapkan metodologi *agile* dengan baik [17]:

1. Kepuasan dari klien merupakan prioritas terutama yaitu dengan menghasilkan produk secepatnya dan konsisten.
2. Menerima terjadinya perubahan, meskipun proses pengembangan hampir selesai.
3. Hasil dapat diselesaikan dalam jangka waktu beberapa minggu hingga bulan.
4. Klien dan pengembang bekerjasama selama pengembangan.
5. Membangun proyek dengan orang-orang yang bermotivasi tinggi , ruang lingkup yang mendukung serta terpercaya untuk merampungkan proyek.

6. Komunikasi secara langsung merupakan hal yang wajib dilakukan untuk mencegah kesalahan informasi.
7. Jika Software bekerja itu adalah tolak ukur dari perkembangan proyek
8. Bantuan secara konsisten dari sponsor, tim, dan klien dibutuhkan untuk menjaga pengembangan.
9. Memperhatikan teknis dan desain merupakan tindakan mengembangkan sifat dari metode agile.
10. Kesederhanaan merupakan hal yang bernilai
11. Desain dan kebutuhan yang baik terdiri dari tim yang mengelola dirinya dengan baik pula
12. Secara berkala tim mencari jalan untuk berkembang menjadi lebih baik.

Dua belas prinsip di atas merupakan fondasi untuk model-model proses yang memiliki sifat dari metodologi agile. Adanya prinsip tersebut merupakan usaha untuk mengkaji 3 asumsi mengenai proyek software yang pada dasarnya:

1. Kebutuhan software sulit untuk diprediksi dari awal dan akan selalu berubah. Selain itu, keinginan klien juga sering berubah seiring proyek berlangsung.
2. Pembangunan dan desain proyek sering terjadi tumpang tindih. Sulit diperkirakan seperti apa desain yang diperlukan dalam pembangunan.
3. Pembangunan, desain, analisis dan testing tidak dapat diperkirakan apakah akan menjadi seperti yang diinginkan.



Gambar 2. 1 Metodologi Agile

Sumber: [17]

Kelebihan dari metodologi agile:

1. Memaksimalkan kepuasan dari klien
2. Pembangunan sistem menjadi makin cepat selesai.
3. Meminimalkan resiko kegagalan dalam penerapan software dari sisi non-teknisnya.
4. Ketika pembangunan sistem mengalami kegagalan, kerugian dari sisi materi akan menjadi lebih rendah.

Perbandingan Metode *System Analysis and Design*:

Tabel 2. 1 Perbandingan Metode System Analysis and Design:

Pengembangan	<i>Waterfall</i>	<i>Prototyping</i>	<i>Agile</i>
<i>User Requirement</i> yang belum jelas	Tidak baik, karena requirement sudah ditentukan sebelum pembuatan sistem dilakukan.	Sangat baik, karena proses analisis dan design yang dilakukan secara berulang, hingga sistem selesai.	Sangat baik, karena proses pengembangan yang dilakukan terus menerus, dan penerimaan akan perubahan.

Pengembangan	<i>Waterfall</i>	<i>Prototyping</i>	<i>Agile</i>
Keandalan Sistem	Baik, karena pembuatan sistem terstruktur, sehingga sistem yang dibuat sesuai dengan kebutuhan.	Tidak baik, karena tidak memiliki analisis dan desain yang baik untuk sistem yang dapat diandalkan,	Sangat baik, karena terfokus pada testing, identifikasi requirement, dan pengembangan yang berulang.
Waktu yang singkat	Tidak baik, karena tidak dapat melakukan perubahan dengan mudah	Sangat baik, karena dapat menyediakan sistem yang dapat digunakan, walaupun belum dapat dipakai secara keseluruhan.	Sangat baik, karena dapat menyesuaikan fungsionalitas sistem dalam waktu tertentu dan dapat melakukan penyesuaian terhadap jadwal.

Sumber : [18]

2.6. Database

Database merupakan gabungan dari file-file yang berhubungan dan terikat satu sama lain, hubungan tersebut bila diperlihatkan dengan kunci dari setiap file yang tersedia. Sebuah database memperlihatkan gabungan data yang dipakai oleh sebuah perusahaan, instansi. Pengelolaan database adalah sebuah cara yang dipergunakan pada file-file yang ada pada instansi yang dimana file-file tersebut dapat diatur, disusun, dipergunakan pada waktu yang diinginkan, juga mampu ditampilkan ke dalam laporan sehingga mampu mengelola file-file yang terdapat informasi tersebut secara terstruktur [19]. Informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya serta database mampu mengurangi pemborosan tempat simpanan luar [15]. Database memiliki kategori-kategori yang digunakan sebagai pengelompokan data. Penggunaan database pada komputer dilakukan dengan menggunakan tabel-tabel [20].


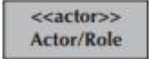

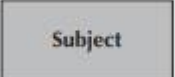

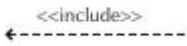
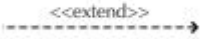

2.7. Unified Modelling Language (UML)

Menurut buku [18] tujuan UML adalah untuk menyediakan kosakata umum berorientasi objek istilah dan teknik diagram untuk memodelkan setiap proyek pengembangan sistem dari analisis hingga implementasi. Pada November 1997, Manajemen Objek Group (OMG) secara resmi menerima UML sebagai standar untuk semua pengembang objek. Penggunaan UML tergantung pada proses pengembangan sistem, diagram yang berbeda memainkan peran yang lebih penting, diantaranya yaitu:

2.7.1. Use Case Diagram

Use Case Diagram menyediakan cara komunikasi yang sederhana dan langsung ke pengguna persis apa yang akan dilakukan sistem, *use case diagram* digambar saat mengumpulkan dan mendefinisikan persyaratan untuk sistem. Dengan cara ini, *use case diagram* dapat mendorong pengguna untuk memberikan persyaratan tingkat tinggi. *Use case diagram* menggambarkan secara sangat sederhana cara fungsi utama sistem dan berbagai jenis pengguna yang akan berinteraksi dengan itu . Berikut ini merupakan beberapa simbol yang digunakan dalam membuat *use case diagram*:

Tabel 2. 2 Use Case Diagram





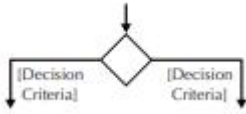
Keterangan	Simbol
<p><i>Actor</i> adalah seseorang atau sistem yang memperoleh manfaat dari dan bersifat eksternal terhadap subjek. Dapat dikaitkan dengan aktor lain menggunakan asosiasi spesialisasi / superclass, dilambangkan dengan panah dengan panah berongga. Ditempatkan di luar batas subjek</p>	 <p>Actor/Role</p> 
<p><i>Use-Case</i> adalah bagian utama dari fungsionalitas sistem. Dapat memperpanjang use case lain. Dapat menyertakan use case lain. Ditempatkan di dalam batas sistem..</p>	
<p><i>Subject</i> termasuk nama subjek di dalam atau di atas. Mewakili ruang lingkup subjek, misalnya, sistem atau individu proses bisnis.</p>	
<p><i>Association Relationship</i> menghubungkan aktor dengan use case yang digunakan untuk berinteraksi</p>	
<p><i>Include relationship</i> Merupakan penyertaan fungsionalitas dari satu use case di dalam yang lain.</p>	
<p><i>Extend relationship</i> merupakan ekstensi dari use case untuk memasukkan perilaku opsional</p>	
<p><i>Generelaization relationship</i> memaparkan kasus penggunaan khusus menjadi lebih umum</p>	

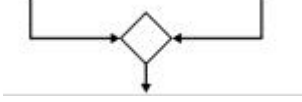

Sumber:[18]

2.7.2. Activity Diagram

Activity diagram digunakan untuk memodelkan perilaku dalam proses bisnis yang tidak bergantung pada objek. Activity diagram dapat digunakan untuk memodelkan segala sesuatu mulai dari alur kerja bisnis tingkat tinggi yang melibatkan banyak kasus penggunaan yang berbeda, hingga detail kasus penggunaan individual, sampai ke detail spesifik dari metode individual. Singkatnya, diagram aktivitas dapat digunakan untuk memodelkan segala jenis proses. Berikut ini merupakan beberapa simbol yang digunakan dalam membuat *activity diagram*:

Tabel 2. 3 Activity Diagram


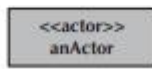
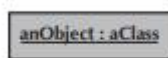
Keterangan	Simbol
<i>Action</i> merupakan perilaku yang sederhana dan tidak dapat dikompromikan. Diberi label berdasarkan namanya.	
<i>Activity</i> digunakan untuk mewakili serangkaian tindakan. Diberi label berdasarkan namanya.	
<i>Initial node</i> menggambarkan awal dari serangkaian tindakan atau kegiatan.	
<i>Final-activity node</i> digunakan untuk menghentikan semua aliran kontrol dan objek mengalir dalam suatu kegiatan (atau tindakan).	
<i>Decision node</i> digunakan untuk mewakili kondisi pengujian untuk memastikan bahwa aliran kontrol atau aliran objek hanya turun satu jalur. Diberi label dengan kriteria keputusan untuk melanjutkan ke jalur tertentu.	



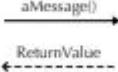
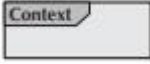
Keterangan	Simbol
<i>Merge node</i> digunakan untuk menyatukan kembali jalur keputusan yang berbeda yang dibuat menggunakan <i>decision node</i> .	
<i>Swimlane</i> digunakan untuk memecah <i>activity diagram</i> menjadi baris dan kolom untuk menetapkan <i>activity</i> individu (atau tindakan) untuk individu atau objek yang bertanggung jawab untuk menjalankan aktivitas (atau tindakan)	

Sumber: [18]

2.7.3. Sequence Diagram

Sequence diagram adalah salah satu dari dua jenis diagram interaksi. Mereka mengilustrasikan objek yang berpartisipasi dalam *use case* dan pesan yang lewat di antara mereka dari waktu ke waktu untuk satu *use case*. *Sequence diagram* adalah model dinamis yang menunjukkan urutan eksplisit pesan yang dikirimkan antara objek dalam interaksi yang ditentukan. Karena *sequence diagram* menekankan urutan aktivitas berbasis waktu yang terjadi antara satu set objek, mereka sangat membantu untuk memahami spesifikasi waktu nyata dan kasus penggunaan yang kompleks. Berikut merupakan beberapa simbol yang digunakan dalam membuat *sequence diagram*:

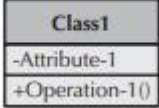
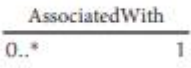


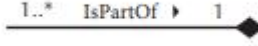
Keterangan	Simbol
<i>Actor</i> Adalah orang atau sistem yang memperoleh manfaat dari sistem dan berada di luar sistem	 
<i>Object</i> berpartisipasi dengan mengirim dan/atau menerima pesan.	

Keterangan	Simbol
<i>Lifeline</i> menunjukkan adanya suatu objek pada <i>sequence</i> .	
<i>Execution occurrence</i> merupakan persegi panjang yang ada pada <i>lifeline</i> . Menunjukkan ketika suatu objek mengirim atau menerima pesan	
<i>Message</i> menyampaikan informasi dari satu objek ke objek lainnya.	
<i>Frame</i> menunjukkan konteks dari <i>sequence diagram</i>	

2.7.4. Class Diagram

Class diagram adalah model statis yang menunjukkan kelas dan hubungan antar kelas yang tetap konstan dalam sistem dari waktu ke waktu. *Class diagram* menggambarkan kelas, yang meliputi: baik perilaku dan status, dengan hubungan antar kelas. Berikut ini merupakan beberapa simbol yang digunakan dalam membuat *class diagram*:

Tabel 2. 4 Class Diagram

Keterangan	Simbol
<i>Class</i> merupakan gambaran dari orang, tempat atau hal tentang sistem	
<i>Attribute</i> merupakan gambaran dari keadaan suatu objek	<p style="text-align: center;">attribute name /derived attribute name</p>
<i>Operation</i> merupakan tindakan atau fungsi yang dapat dilakukan oleh <i>class</i>	<p style="text-align: center;">operation name ()</p>
<i>Association</i> merupakan gambaran dari hubungan antara beberapa <i>class</i> atau <i>class</i> dengan dirinya sendiri	<p style="text-align: center;">AssociatedWith</p> 
<i>Generalization</i> merupakan gambaran dari hubungan antara beberapa <i>class</i>	
<i>Aggregation</i> merupakan gambaran dari hubungan logis antara beberapa <i>class</i> dan dirinya sendiri	
<i>Composition</i> merupakan gambaran dari hubungan fisik dari beberapa <i>class</i> dan dirinya sendiri	

Sumber: [18]

2.8. PHP

PHP adalah sebuah skenario untuk pemrograman script web server-side, skenario tersebut yang membangun HTML secara on the fly, tujuannya adalah HTML yang dibuat berdasarkan aplikasi bukan merupakan HTML yang dibuat dengan memakai teks editor atau HTML editor [19]. Penggunaan PHP memungkinkan Web dapat dibuat dinamis sehingga maintenance situs

Web tersebut menjadi lebih mudah dan efisien. PHP merupakan software Open-Source yang disebar dan dilisensikan secara gratis serta dapat didownload secara bebas dari situs resminya [21]. Keuntungan PHP, kode yang menyusun program tidak perlu diedarkan ke pemakai sehingga kerahasiaan kode dapat dilindungi [22].

2.9. HTML

HTML (Hypertext Markup Language) merupakan suatu script yang bias menampilkan informasi dan daya kreasi kita melalui internet. HTML sendiri adalah suatu dokumen teks biasa yang mudah untuk dimengerti dibandingkan bahasa pemrograman lainnya, dan karena bentuknya itu maka HTML dapat dibaca oleh platform yang berlainan seperti windows, unix dan lainnya [23].

2.10. MySQL

MySQL merupakan sistem basis data yang dasarnya hanya beroperasi di sistem Liunx dan Unix. Bersamaan dengan perkembangan zaman dan meningkatnya peminat yang memakai sistem basis data ini, MySQL meluncurkan versi yang bisa diinstall diberbagai macam platform, termasuk Windows [19]. MySQL ini memiliki sifat Open Source dan merupakan model dari software DBMS. Karena sifatnya yang open source, hal ini menunjukkan bahwa MySQL menyediakan source codenya yang mampu didownload secara di internet secara gratis. Perusahaan konsultan TcX yang berlokasi di Swedia merupakan perusahaan yang menciptakan MySQL , dan dialihkan menjadi perusahaan MySQL AB. Lalu MySQL AB dialihkan lagi menjadi Sun Microsystems, yang pada 27 Januari 2010 resmi menjadi perusahaan dari Oracle Corporation yang terdaftar pada perjanjian tanggal 20 April 2009 [24].

2.11. XAMPP

XAMPP adalah perangkat lunak bebas yang mendukung banyak sistem operasi yang berfungsi sebagai server yang berdiri sendiri (localhost) [25]. XAMPP adalah salah satu paket instalasi apache sebagai *web server*, PHP sebagai bahasa pemrograman, dan MySQL sebagai database secara instan

yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut [26]. XAMPP merupakan paket PHP dan MySQL berbasis open source, yang digunakan sebagai alat pembantu pengembangan aplikasi berbasis PHP [20]. Melalui aplikasi ini, developer dapat menguji aplikasi secara langsung dari komputer tanpa perlu terkoneksi dengan internet. XAMPP juga dilengkapi fitur manajemen database PHPMyAdmin, sehingga pengembangan web berbasis database dapat dilakukan dengan mudah [25].

2.12. Penelitian Terdahulu

Tabel 2. 5 Penelitian Terdahulu

Tentang Jurnal	Hasil Penelitian	Adopsi dari Penelitian
<p>Judul Jurnal: <i>Supply Chain Management</i> Pabrik Roti ABC dengan Metode <i>Wighted Moving</i> <i>Avwrage</i> (WMA)</p> <p>Penulis: Herlina, Yohansen, Fransiska Prihatini Sihotang</p> <p>Tahun: 2020</p> <p>Nama Jurnal: Jurnal Teknik Informatikan dan Sistem Informasi Vol. 7, No. 2, Agustus 2020, Hal. 349- 363</p>	<p>Hasil akhir dari penelitian ini adalah berupa sebuah sistem informasi manajemen rantai pasok yang dapat membantu perusahaan dalam merencanakan banyak produksi dengan lebih baik dan cepat, lalu memudahkan perusahaan dalam mengefisiensikan ketersediaan bahan baku dan hasil produksi, serta dapat meningkatkan manajemen perusahaan dengan memproses laporan bahan baku, penjualan, dan retur.</p>	<p>Digunakan sebagai referensi dalam pemodelan sistem.</p>

Tentang Jurnal	Hasil Penelitian	Adopsi dari Penelitian
<p>Judul Jurnal: Penerapan Agile Unified Process dalam Pembangunan Sistem Informasi Akuntansi Pendapatan Jasa</p> <p>Penulis: Nurfia Oktaviani Syamsiah</p> <p>Tahun: 2016</p> <p>Nama Jurnal: Jurnal Prespektif, Vol. 13, No. 2</p>	<p>Hasil akhir dari penelitian ini adalah pembuktian akan metode <i>agile</i> bahwa metode ini adalah metode yang paling cocok untuk kebutuhan pengembangan suatu sistem.</p>	<p>Digunakan sebagai referensi dalam pengembangan sistem dengan menggunakan metode <i>agile</i>.</p>
<p>Judul Jurnal: Sistem Informasi Penunjang Proses Pemesanan dan Desain Kue pada Toko Kue Artisan Online Berbasis Web</p> <p>Penulis: Tania Jovita Wibowo, Suryasari, Arnold Aribowo, Andre E. Widjaja</p> <p>Tahun: 2019</p> <p>Nama Jurnal: ULTIMA InfoSys, Vol. X, No. 1</p>	<p>Hasil yang diperoleh dari penelitian ini adalah sistem berbasis web yang mendukung proses pemesanan dan desain kue pada toko kue artisan online. Sistem ini dapat mempermudah proses pemesanan dan kustomisasi pesanan pada desain kue yang dipesan.</p>	<p>Digunakan sebagai referensi untuk fitur yang akan dibutuhkan di dalam sistem</p>

Berdasarkan tabel 2.5, pada penelitian terdahulu yang pertama dengan judul “*Supply Chain Management* Pabrik Roti ABC dengan *Metode Weghted Moving Average (WMA)*” dimana pada penelitian ini dibuatnya sebuah sistem untuk melakukan kontrol terhadap stok bahan baku, hasil produksi dan pembuatan laporan yang diperlukan oleh perusahaan. Pada penelitian ini diambilnya sebagai referensi dalam pembuatan model dari sistem baik itu *use case*, *activity diagram*, dan juga *class diagram*.

Untuk penelitian kedua dengan judul “Penerapan *Agile Unified Process* dalam Pembangunan Sistem Informasi Akuntansi Pendapatan Jasa” dimana penelitian ini mampu menjelaskan penggunaan metode *agile* yang baik dimana terjadinya beberapa perubahan requirement selama proses pembuatan sistem, dilakukannya revisi terhadap sistem berdasarkan umpan balik dari *client* yang mampu dilakukan tanpa adanya kesulitan, sehingga mampu membuktikan bahwa metode *agile* ini merupakan metode yang sederhana, mudah diterapkan, memiliki visibilitas tinggi, dan semua fitur sesuai dengan tujuan dari *client*. Penelitian ini digunakan sebagai referensi dalam penggunaan metode *agile* dalam pengembangan sistem.

Penelitian terakhir yaitu penelitian dengan judul “Sistem Informasi Penunjang Proses Pemesanan dan Desain Kue pada Toko Kue Artisan Online Berbasis Web” dimana penelitian ini membuat sebuah sistem yang dapat membantu mempermudah proses pemesanan pada desain kue yang dipesan dari toko kue artisan. Penelitian ini diambil sebagai referensi karena dianggap memiliki fitur yang hampir mirip dengan sistem yang akan dibuat, sehingga dijadikan sebagai referensi dalam pembuatan beberapa fitur dari sistem.

