BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Tabel 2.1 Penelitian Terdahulu

NO	PENELITI	JUDUL	METODE	HASIL
1	Tri Septiar Syamfithriani (2017)	Sistem Informasi Inventori dan Pengelolaan Sapronak dengan pendekatan Supply Chain Management Studi Kasus di PT Aretha PS	**Supply Cahin Management**	Hasil yang di dapat adalah sistem mampu melakukan penyimpanan data yang lebih efektif dan efisien serta mengurangi resiko kerusakan/kehilangan data karena penyimpanan data sudah terkomputerisasi
2	Mhd Bustanur Rahmad (2014)	Perancangan Sistem Informasi *Inventory Spare Part* Elektronik Berbasis *Web PHP* dengan Studi Kasus CV Human Global Service Yogjakarta	**FIFO (First In First Out)**	hasil yang diperoleh adalah membuat sitem persediaan barang secara terkomputerisasi yang dapat dijadikan alat bantu alat bantu informasi dalam peningkatan melakukan pengolahan data

Tabel 2.1 Penelitian Terdahulu (Lanjutan)

NO	PENELITI	JUDUL	METODE	HASIL
				barang serta stok gudang dan mencatat penggunaan biaya *inventory*,
3	Ahmad Fathoni (2016)	Rancang Bangun Sistem *Extreeme Programming* sebagai Metodologi Pengembangan Sistem	*Extreeme Programming*	Hasil yang didapat dalam penelitian ini adalah metode ini mendukung waktu percepatan pembangunan suatu sistem serta memprioritaskan komunikasi yang baik antara klien maupun antar anggota sesama tim
4	Reynaldo Rendi Mulyono (2017)	Perancangan dan Implementasi Sistem *Inventory* sayur Organik dengan menggunakan *Framework CodeIgniter*	**Waterfall**	Hasil dari penelitian ini adalah penelitian tersebut adalah penulis mampu mengembangkan sistem secara baik dengan *Framework CodeIgniter* yang dapat menggantikan sistem sebelumnya

Tabel 2.1 Penelitian Terdahulu (Lanjutan)

NO	PENELITI	JUDUL	METODE	HASIL
5	Hendra Agusvianto (2017)	Sistem Informasi Inventori Gudang untuk Mengontrol Persediaan Barang Pada Gudang Studi Kasus: PT. Alaisys Sidoarjo	**Waterfall**	Dengan adanya aplikasi berbasis web ini dapat memberikan laporan pada pengolahan data barang dari gudang ke kantor pusat secara tepat, akurat dan tepat sasaran

1. Oleh Tri Septiar Syamfithriani, Almand Muhammad, M. Deri Eka P (2017) dari Fakultas Ilmu Komputer, Universitas Kuningan, dengan judul Sistem Informasi Inventori dan Pengelolaan Sapronak dengan pendekatan Supply Chain Management Studi Kasus di PT Aretha PS. Dalam penelitian ini penulis mengangkat masalah tentang pencatatan persedian barang pada perusahaan mengunakan aplikasi distro,yakni aplikasi yang hanya memberi informasi data pembelian dan penjualan yang belum terintegrasi sehingga masih harus mengolah data secara manual untuk pembuatan laporan dan pemesanan barang. Hasil yang di dapat adalah sistem mampu melakukan penyimpanan data yang lebih efektif dan efisien serta mengurangi resiko kerusakan/kehilangan data karena penyimpanan data sudah terkomputerisasi.
2. Oleh Mhd Bustanur Rahmad, Teddy Setiady (2014) dari Program Studi Teknik Informatika, Universitas Ahmad Dahlan dengan judul Perancangan

Sistem Informasi *Inventory Spare Part* Elektronik Berbasis *Web PHP* dengan Studi Kasus CV Human Global Service Yogjakarta. Disini penulis mengangkat masalah tentang masalah pencatatan barang masih manual. Adapun stok barang dan pembuatan laporannya masih menggunakan penulisan di buku, maka dari itu banyaknya waktu yang dibutuhkan untuk proses pembuatan laporan secara tepat dan transaksi penjualan barang yang akurat dan memperbesar kemungkinan kesalahan dalam pencatatan. Sehingga hasil yang diperoleh adalah membuat sitem persediaan barang secara terkomputerisasi yang dapat dijadikan alat bantu alat bantu informasi dalam peningkatan melakukan pengolahan data barang serta stok gudang dan mencatat penggunaan biaya *inventory*, sehingga pihak manajemen dapat mengambil keputusan berdasarkan rekapitulasi transaksi, sisa stok dan informasi lain.

3. Oleh Ahmad Fathoni, Dhany Dwi (2016) dari Program Studi Sistem Komputer Universitas Serang Raya dengn Judul Rancang Bangun Sistem *Extreeme Programming* sebagai Metodologi Pengembangan Sistem. Dimana dalam penelitian ini penulis mengangkat masalah yaitu mengembangkan sebuah sistem dengan metode *Extreeme Programming.* Metode ini adalah metode yang dibuat untuk menghindari situasi yang suatu saat bisa berubah. Seperti keinginan konsumen dalam hal pembuatan perangkat lunak. *Extreeme Programming* berguna untuk mempercepat pekerjaan suatu tim dalam organisasi atau perusahaan. Karena dalam *Extreeme Programming Life cycle* menuntut ke suatu tim untuk menyelesaikan aktivitas *Planning,Analys, Design & Code, Test, Deploy*

dalam tempo waktu yang telah ditentukan. Hasil yang didapat dalam penelitian ini adalah metode ini mendukung waktu percepatan pembangunan suatu sistem serta memprioritaskan komunikasi yang baik antara klien maupun antar anggota sesama tim.

4. Oleh Reynaldo Rendi Mulyono, Hindriyanto D.P (2017) dari Fakultas Teknologi Informasi, Universitas Kristen Setya Wacana dengan judul Perancangan dan Implementasi Sistem *Inventory* sayur Organik dengan menggunakan *Framework CodeIgniter*. Dimana dalam penelitian ini penulis mengangkat masalah yaitu, sistem yang sebelumnya sudah ada namun belum begitu berkontribusi sepenuhnya dikarenakan kurang praktis dalam penggunaanya dan keamanan dari sistem yang tidak diperhatikan. Hasil dari penelitian ini adalah penelitian tersebut adalah penulis mampu mengembangkan sistem secara baik dengan *Framework CodeIgniter* yang dapat menggantikan sistem sebelumnya. Dan berdasarkan pengujian melalui *Blackbox Testing* disimpulkan fungsi sistem sudah bekerja sesuai dengan yang diharapkan.
5. Oleh Hendra Agusvianto (2017) dari program Studi S2 Pendidikan teknologi dan Kejuruan, Universitas Negeri Surabaya dengan judul Sistem Informasi Inventori Gudang untuk Mengontrol Persediaan Barang Pada Gudang Studi Kasus: PT. Alaisys Sidoarjo. Dimana dalam penelitian ini penulis mengangkat masalah pengolahan data gudang PT. Alaisys masih dilakukan secara manual. Pencatatan informasi dan pengolahan laporan masih dilakukan tulis tangan. Penulis membangun aplikasi sistem inventori dengan tujuan untuk memudahkan karyawan dalam melakukan

pengontrolan barang. Hasil yang didapat oleh penulis dalam penelitiannya adalah Sistem ini merupakan suatu aplikasi system inventori pada gudang yang berfungsi untuk pencatatan keluar masuk barang oleh karyawan PT.Alaisys dapat dilakukan dengan struktur sehingga dapat memberikan bantuan dalam hal efisiensi waktu kegiatan pencatatan dan penyusunan data pada barang. Dengan adanya aplikasi berbasis web ini dapat memberikan laporan pada pengolahan data barang dari gudang ke kantor pusat secara tepat, akurat dan tepat sasaran.

2.2 Konsep Dasar Rancang Bangun

Menurut Hidayah dkk. (2015) rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dan sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan. Sedangkan pengertian bangun atau pembangunan sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian. Jadi pengertian rancang bangun adalah membentuk dan membuat sistem dengan rencana atau proses melalui beberapa tahapan dan metode untuk dapat diterapkan guna memecahkan masalah.

2.3 Pengertian Sistem

Dalam Ermatita (2016) Sutabri menyatakan Sistem adalah suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu. Dengan kata lain sistem adalah kumpulan dari beberapa elemen yang membentuk satu kesatuan yang utuh untuk mencapai tujuan yang sama. Karakteristik dari sistem adalah :

1. Komponen Sistem (*Components*)

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem.

2. Batasan Sistem (*Boundary*)

   Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan Luar Sistem (*Environtment*)

   Lingkungan luar sistem adalah bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut.

4. Penghubung Sistem (*Interface*)

   Penghubung sistem atau interface adalah media yang menghubungkan sistem dengan subsistem yang lain.

5. Masukan Sistem (*Input*)

   Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

6. Keluaran Sistem (*Output*)

   Keluaran ini merupakan masukan bagi subsistem yang lain. Seperti contoh sistem informasi, keluaran yang dihasilkan adalah informasi, di mana informasi ini dapat digunakan sebagai masukan untuk

pengambilan keputusan atau hal-hal lain yang merupakan input bagi subsistem lain.

7. Pengolah Sistem (*Process*)

   Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem (*Objective*) Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik

2.4 Pengertian *Website*

Menurut Febrin (2012) *Website* adalah kumpulan halaman-halaman yang digunakan untuk menampilkan informasi atau teks, gambar diam tau bergerak, animasi, suara, dan atau gabungan dari semuanya baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman. Hubungan antara satu halaman *web* yang lainnya disebut *hyperlink*, sedangkan teks yang dijadikan media penghubung disebut *hypertext*. Fungsi website terdiri dari:

1. *Personal website*, *website* yang berisi informasi pribadi seseorang.
2. *Commercial website*, *website* yang dimiliki oleh sebuah perusahaan yang bersifat bisnis.
3. *Goverment webiste*, *website* yang dimiliki oleh instansi pemerintahan, pendidikan yang bertujuan memberikan pelayanan kepada pengguna.
4. *Non-Profit Organization website*, *website* yang dimiliki oleh organisasi yang bersifat non-profit atau tidak bersifat bisnis.

2.5 Pengertian Persediaan Barang

Dalam Ramdhany dan Hardianty (2015) Pernyataan Standar Akuntansi Pemerintahan (PSAP) 05 menyatakan Persediaan adalah aset lancar dalam bentuk barang atau perlengkapan yang dimaksudkan untuk mendukung kegiatan operasional pemerintah, dan barang-barang yang dimaksudkan untuk dijual dan/atau diserahkan dalam rangka pelayanan kepada masyarakat. Suatu aset digolongkan kedalam persediaan apabila:

1. Barang atau perlengkapan (*supplies*) yang digunakan dalam rangka kegiatan operasional pemerintah.
2. Bahan atau perlengkapan (*supplies*) yang digunakan dalam proses produksi.
3. Barang dalam proses produksi yang dimaksudkan untuk dijual atau diserahkan kepada masyarakat.
4. Barang yang disimpan untuk dijual atau diserahkan kepada masyarakat dalam rangka kegiatan pemerintahan.

2.5.1 Fungsi Persediaan

Sedangkan fungsi sistem persediaan atau *inventori* adalah: (Rangkuti, 2000)

1. Fungsi *Decoupling*

   Memungkinkan operasi-operasi perusahaan *internal* dan *eksternal* mempunyai kebebasan sehingga dapat memenuhi permintaan tanpa tergantung pada *supplier*.

2. Fungsi *Economic Lot Sizing*

   Adalah persediaan yang perlu memperhatikan penghematan

persediaan barang, biaya pengangkutan perunit menjadi lebih murah dan sebagainya.

3. Fungsi *Antisipasi*

   Adalah persediaaan yang pengadaannya karena permintaan musiman yang dapat diperkirakan atau diramalkan berdasarkan pengalaman atau data-data masa lalu, atau karena ketidakpastian jangka waktu pengiriman dan permintaan barang selama periode tertentu.

2.5.2 Jenis-Jenis Persediaan *(Inventory)*

Jenis-jenis persediaan terdiri dari (Rangkuti, 2000):

1. Persediaan barang mentah, yaitu persediaan barang-barang berwujud, seperti kayu, besi serta komponen–komponen lainnya yang digunakan dalam proses produksi.
2. Persediaan komponen rakitan, yaitu persediaan barang-barang yang terdiri dari komponen-komponen yang diperoleh dari perusahaan lain, dimana secara langsung dapat dirakit menjadi suatu produk.
3. Persediaan bahan pembantu atau penolong *(supplier)*, yaitu persediaan barang-barang yang diperlukan dalam proses produksi, tetapi tidak merupakan bagian atau komponen barang jadi.
4. Persediaan barang dalam proses *(work in proses)*, yaitu persediaan barang-barang yang merupakan keluaran dari tiap-tiap bagian dalam proses produksi atau yang telah diolah menjadi suatu bentuk, tetapi masih perlu diproses lebih lanjut menjadi barang jadi.

5. Persediaan barang jadi *(finished goods)*, yaitu persediaan barang-barang yang telah selesai diproses atau diolah dalam pabrik dan siap dijual atau dikirim kepada pelanggan.

2.6 Pengertian *Codeigniter*

CodeIgniter adalah sebuah *framework* yang digunakan untuk membuat sebuah aplikasi berbasis web yang disusun dengan menggunakan bahasa PHP. Di dalam CI ini terdapat beberapa macam kelas yang berbentuk *library* dan *helper* yang berfungsi untuk membantu pemrogram dalam mengembangkan aplikasinya.

Model View Controller merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web. Berawal dari bahasa pemrograman *Small Talk*, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface*, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC *pattern* dalam suatu aplikasi sebagai berikut.

1. *View*, merupakan bagian yang menangani *presentation logic*. Pada suatu aplikasi web bagian ini biasanya berupa *file template* HTML yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada *user*. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2. *Model*, biasanya berhubungan langsung dengan *database* untuk memanipulasi data (*insert, update, delete, search*), menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian view.

3. *Controller*, merupakan bagian yang mengatur hubungan antara bagian *view*, *controller* berfungsi untuk menerima *request* dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.

Dengan menggunakan prinsip MVC suatu aplikasi dapat dikembangkan sesuai dengan kemampuan *developer*, yaitu *programmer* yang menangani bagian model dan *controller*. Sedangkan, designer menangani bagian *view* sehingga penggunaan arsitektur MVC dapat meningkatkan *maintanability* dan organisasi kode. Walaupun demikian, dibutuhkan komunikasi yang baik antara *programmer* dan designer dalam menangani variabel-variabel yang akan ditampilkan.

Ada beberapa kelebihan Codeigniter (CI) dibandingkan dengan Framework PHP lain, sebagai berikut:

1. Performa Sangat Cepat
2. Konfigurasi Yang Sangat Minim
3. Banyak Komunitas
4. Dokumentasi Yang Sangat Lengkap

2.7 Pengertian MySQL

Menurut Nugroho (2004), MySQL merupakan database yang paling digemari dikalangan Programmer Web, dengan alasan bahwa program ini merupakan database yang sangat kuat dan cukup stabil untuk digunakan sebagai media penyimpanan data. Sebagai database server yang mampu untuk memanajemen database yang baik, MySQL terhitung merupakan database yang paling digemari dan paling banyak digunakan di banding database lainnya.

Selain MySQL masih terdapat beberapa jenis database server yang juga memiliki kemampuan yang jaga tidak bisa dianggap enteng, database itu adalah

Oracle dan PostgreSQL. Didalam dunia internet, MySQL dijadikan sebuah database yang paling banyak digunakan selain database yang bersifat share ware seperti Ms Access, pengguna MySQL ini biasanya dipadukan menggunakan program aplikasi php, karena dengan menggunakan kedua program tersebut diatas telah terbukti akan kehandalannya dalam menangani permintaan data. Kemampuan lain yang dimiliki MySQL adalah mampu mendukung Relasional Database Manajemen Sistem (DBMS), sehingga dengan kemampuan ini MySQL akan mampu menangani data-data sebuah perusahaan yang berukuran sangat besar hingga berukuran Giga Byte. MySQL adalah sebuah software database yang bersifat *free* (gratis) yang digunakan pada Aplikasi Perizinan Tenaga Kesehatan karena MySQL bebas lisensi dan sudah terbukti tangguh dan efisien.

2.8 Metode Pengembangan Perangkat Lunak

Menurut Pressman, (2002) *Extreme Programming (XP)* adalah metode pengembangan perangkat lunak yang ringan dan termasuk salah satu *agile methods* yang dipelopori oleh Kent Beck, Ron Jeffries, dan Ward Cunningham. *XP* merupakan *agile methods* yang paling banyak digunakan dan menjadi sebuah pendekatan yang sangat terkenal. Sasaran *XP* adalah tim yang dibentuk berukuran antara kecil sampai medium saja, tidak perlu menggunakan sebuah tim yang besar. Hal ini dimaksudkan untuk menghadapi *requirements* yang tidak jelas maupun terjadinya perubahan-perubahan *requirements* yang sangat cepat.

2.8.1 Nilai Utama Extreme Programming

Terdapat empat nilai utama pada metode Extreme Programming, yaitu:

1. Komunikasi (*Communication*)

Kurangnya komunikasi merupakan penyebab utama kegagalan pengembangan software. Oleh karena itu *Extreme Programming* (XP) memfokuskan diri pada hubungan komunikasi yang baik antar tim-klien, anggota tim, dan manajer proyek. Komunikasi dalam XP dibangun dengan melakukan pemrograman berpasangan (*pair programming*). Klien harus dilibatkan dalam proses pengembangan perangkat lunaknya dengan tujuannya untuk memberikan pandangan pengembang sesuai dengan pandangan pengguna sistem yang dibangun.

2. Kesederhanaan (*Simplicity*)

   Extreme Programming (XP) melakukan semua pekerjaan dengan sederhana dan praktis tanpa mengurangi fungsi utamanya. Dalam pengerjaan, metode yang dipilih adalah metode yang pendek dan simpel. Jangan terlalu rumit dalam membuat desain, hilangkan fitur yang tidak ada gunanya atau hapus fungsi yang tidak terpakai. Dengan kata lain lebih baik melakukan hal yang sederhana saat sekarang (sesuai kebutuhan) dan mengembangkannya nanti jika diperlukan.

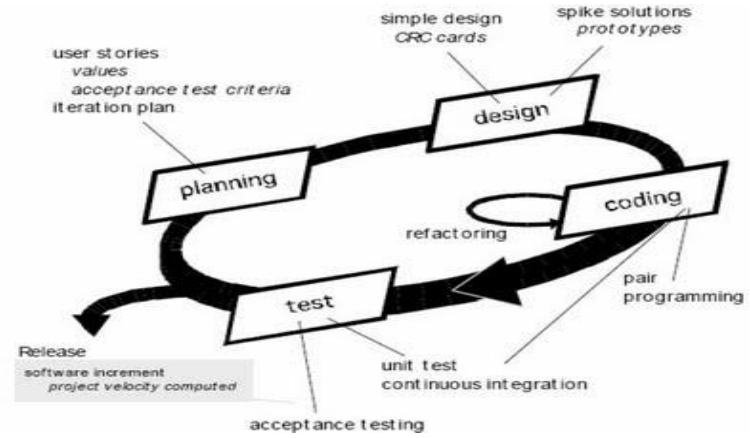
3. Umpan balik (*Feedback*)

   Selalu evaluasi perkembangan perangkat lunak yang sedang dikerjakan. Segala informasi harus dikumpulkan setiap interval waktu yang konsisten dan kesalahan-kesalahan yang muncul selama proses pengembangan harus dibahas dan dicari solusinya. Umpan balik tersebut berfungsi sebagai indikator kemajuan proyek dan menginformasikan pemimpin proyek apabila perubahan perlu dibuat.

4. Keberanian (*Courage*)

   Programmer Extreme Programming (XP) didorong untuk berani bereksperimen dan menulis ulang kode jika mereka tidak puas dengan kode atau desain yang sudah ada. Hal ini membantu mempertahankan moral serta integritas para pengembang proyek dan dapat mendukung lebih lanjut komunikasi dengan anggota proyek lainnya.

Preesman (2002) menjelaskan tahapan *Extreme Programming* yang dapat dilihat pada gambar berikut:



Gambar 2.1 Metode *Extreme Programming*

2.8.2 Tahapan Metode *Extreme Programming*

Terdapat empat langkah dalam metode *Extreme Programming*, yaitu:

1. *Planning.* Tahap *planning* dimulai dengan membuat *user stories* yang menggambarkan *output*, fitur, dan fungsi-fungsi dari *software* yang akan dibuat. *User stories* tersebut kemudian diberikan bobot seperti prioritas dan dikelompokkan untuk selanjutnya dilakukan proses *delivery* secara *incremental*.
2. *Design.* Design di *Extreme Programming* mengikuti prinsip *Keep It Simple* (KIS). Untuk design yang sulit, *Extreme Programming* akan

menggunaan *Spike Solution* dimana pembuatan design dibuat langsung ke tujuannya. *Extreme Programming* juga mendukung adanya *refactoring* dimana *software system* diubah sedemikian rupa dengan cara mengubah stuktur kode dan menyederhanakannya namun hasil dari kode tidak berubah.

3. *Coding.* Proses *coding* pada XP diawali dengan membangun serangkaian unit test. Setelah itu pengembang akan berfokus untuk mengimplementasikannya. Dalam *Extreme Programming* diperkenalkan istilah *Pair Programming* dimana proses penulisan program dilakukan secara berpasangan. Dua orang programmer saling bekerjasama di satu komputer untuk menulis program. Dengan melakukan ini akan didapat *real-time problem solving* dan *real-time quality assurance.*
4. *Testing.* Tahap ini dilakukan pengujian kode pada *unit test*. Dalam *Extreme Programming*, diperkenalkan XP *acceptance test* atau biasa disebut *customer test*. Tes ini dilakukan oleh *customer* yang berfokus kepada fitur dan fungsi sistem secara keseluruhan. *Acceptance test* ini berasal dari *user stories* yang telah diimplementasikan.

2.8.3 Kelebihan dan kelemahan Metode *Extreme Programming*

1. Kelebihan *Extreme Programming*, yaitu:
    a. Meningkatkan kepuasan kepada klien
    b. Pembangunan system dibuat lebih cepat
    c. Menjalin komunikasi yang baik dengan klien

d. Meningkatkan komunikasi dan sifat saling menghargai antar *developer*

2. Kelemahan *Extreme Programming*, yaitu:
   a. Cerita-cerita yang menunjukkan *requirements* dari pelanggan kemungkinan besar tidak lengkap sehingga *Developer* harus selalu siap dengan perubahan karena perubahan akan selalu diterima.
   b. Tidak bisa membuat kode yang detail di awal (prinsip *simplicity* dan juga anjuran untuk melakukan apa yang diperlukan hari itu juga).
   c. *XP* tidak memiliki dokumentasi formal yang dibuat selama pengembangan. Satu-satunya dokumentasi adalah dokumentasi awal yang dilakukan oleh user.

2.9 Unified Modelling Language (UML)

Menurut Rosa dan Shalahuddin, (2013) 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. UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori yaitu:

1. ***Stucture Diagrams***

   Kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan, seperti *class diagram*,

object diagram, component diagram, composite structure diagram, package diagram, dan *deployment diagram.*

2. ***Behavior Diagrams***

   Kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem, seperti *use case diagram, activity diagram,* dan *state machine diagram.*

3. ***Interactions Diagrams***

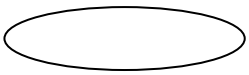
   Kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem, seperti *sequence diagram, communication diagram, timing diagram* dan *interaction overview diagram.*

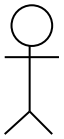
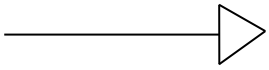
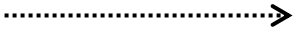
Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada UML *(Unified Modelling Language):*

2.9.1. Use Case Diagram

Use case diagram atau diagram *use case* merupakan pemodelan untuk kelakuan *(behavior)* sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Tabel 2.1 Simbol *Use Case Diagram*

No.	Simbol	Keterangan
1.		*Use Case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor, biasanya menggunakan kata kerja.

No.	Simbol	Keterangan
2.		Aktor merupakan seseorang/sesuatu yang berinteraksi dengan yang akan dibuat diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda.
3.		Asosiasi/*association* merupakan komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
4.		Generalisasi (*generalization*) merupakan *hupromosin* (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
5.	<< Include >> 	*Include* berarti *use case* yang ditambahkan akan dipanggil saat *use case* tambahan dijalankan.
6.	<<Extend>> 	Ekstensi (*extend*) merupakan *use case* tambahan ke sebuah *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu.

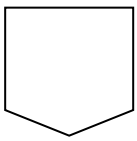
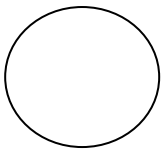
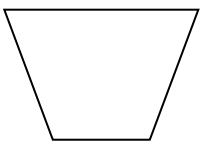
Sumber: Rosa dan Shalahuddin, (2014:156)

2.9.2 Flowchart Diagram

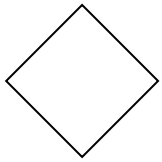
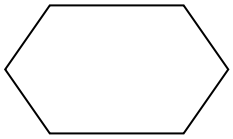
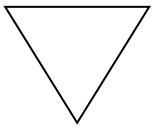
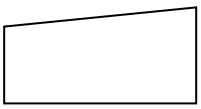
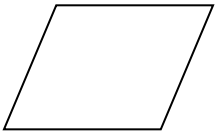
Flowchart merupakan gambaran secara grafik dari prosedur dan langkah-langkah yag dijalani dalam suatu program. *Flowchart* dapat mempermudah penyelesaian masalah yang khusunya perlu dipelajari dan dievaluasi lebih lanjut (Indrajani, 2011). Menurut Sahid (2010) *Flowchart* adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi.

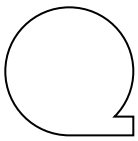
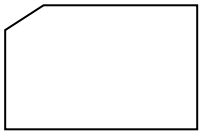
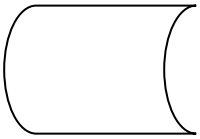
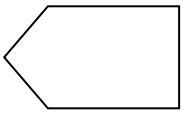
Berikut ini merupakan notasi atau simbol-simbol dalam penggambaran *flowchart*.

Tabel 2.2 Simbol *Flowchart*

Simbol	Keterangan
	*Symbol Off-line Connector* (Simbol untuk keluar/masuk prosedure atau proses dalam lembar/halaman yang lain)
	*Symbol Connector* (Simbol untuk keluar/masuk prosedur atau proses dalam lembar/halaman yang sama)
	*Symbol Process* (Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer)
	*Symbol Manual Operation* (Simbol yang menunjukkan pengolahan yang tidak dilakukanoleh komputer)

Tabel 2.2 Simbol *Flowchart* (Lanjutan)

Simbol	Keterangan
	*Symbol Decision* (Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban/aksi)
	*Symbol Predefined Process* (Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage)
	*Symbol* Terminal (Simbol untuk permulaan atau akhir dari suatu program)
	*Symbol Off-line Storage* (Simbol yang menunjukkan bahwa data di dalam symbol ini akan disimpan)
	*Symbol* Manual *Input* (Simbol untuk pemasukan data secara manual *on-line keyboard*)
	*Symbol input-output* (*Symbol* yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya)

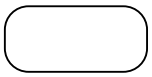
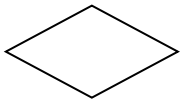
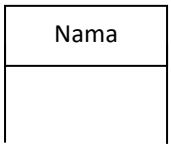
	*Symbol magnetic-tape unit* (*Symbol* yang menyatakan input berasal pita *magnetic* atau output disimpan ke pita *magnetic*)
	*Symbol punched card* (Simbol yang menyatakan input berasal dari kartu atau *output* ditulis ke kartu)
	*Symbol disk and on-line storage* (Simbol untuk menyatakan input berasal dari *disk* atau output disimpan ke *disk*)
	*Symbol display* (Simbol yang menyatakan peralatan output yang digunakan yaitu layar, *plotter*, printer, dan sebagainya)
	*Symbol* dokumen (Simbol yang menyatakan *input* berasal dari dokumen dalam bentuk kertas)

2.9.3 Activity Diagram

Menurut Rosa dan Shalahuddin (2013) *Activity* Diagram adalah diagram aktivitas atau menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Rosa dan Shalahuddin (2013) menjelaskan simbol-simbol *activity diagram* yang ditampilkan pada tabel 2.3

Tabel 2.3 Simbol *Activity* Diagram

Simbol	Keterangan
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 percabanagn dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan / *join*	Asosiasi percabangan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Starus akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane  Atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.9.4 Class Diagram

Menurut Rosa dan Shalahuddin (2013) *class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program membuat kelas-kelas sesuai rancangan didalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

Rosa dan Shalahuddin (2013) menjelaskan simbol-simbol class diagram yang ditampilkan pada tabel 2.4.

Tabel 2.4 Simbol *Class Diagram*

Simbol	Keterangan
Kelas Nama_kelas +atribut +operasi()	Kelas pada struktur sistem.
Antarmuka / *interface* nama_interface	Sama dengan konsep *interface* dalam pemrograman berorientasi objek.
Asosiasi / *association*	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicty.*
Asosiasi berarah / *directed cassociation*	Relasi antarkelas dengan makna yang satu digunakan oleh kelas yang lain; asosiasi biasanya juga disertai dengan *multiplicity.*

Tabel 2.4 Simbol *Class Diagram* (Lanjutan)

Simbol	Keterangan
Generalisasi	Relasi antar kelas dengan maka generalisai-spesialisai (umum khusus).
Kebergantungan / *dependecy* 	Kebergantungan antar kelas.
Agregasi / *aggregation*	Relasi antarkelas dengan makna semua bagian (*whole-part).*

2.10 Analisis PIECES

Untuk mengindentifikasi masalah, maka harus dilakukan analisis terhadap PIECES *(Performance, Information, Economy, Control, Efeciency, dan Service)* (Hanif. Al fatta, 2007).

1. Analisis Kinerja (*performance*)

   Masalah Kinerja terjadi ketika tugas-tugas yang dijalankan oleh sistem mencapai sasaran. Kinerja diukur dengan jumlah produksi dan waktu tanggap. Jumlah produksi adalah jumlah pekerjaan yang dilaksanakan selama jangka waktu tertentu. Waktu tanggap adalah keterlambatan rata-rata antara suatu transaksi dengan tanggapan yang diberikan kepada transaksi tersebut.

2. Analisis Informasi (*information***)**

   Informasi merupakan komoditas yang penting bagi pemakai akhir.

Karena Informasi yang akan dihasilkan dapat memenuhi keinginan dari pengguna dan juga dapat mengatasi masalah-masalah yang ada. Informasi yang ada ini pun dapat dimanfaatkan oleh pihak internal atau pihak external.

3. Analisis ekonomi (*economy*)

   Ekonomi merupakan motivasi paling umum bagi suatu lembaga. Pijakan dasar bagi kebanyakan manajer adalah biaya yang murah.

4. Analisis Pengendalian (*control*)

   Tugas-tugas dari sustu sistem informasi perlu di monitor dan dibetulkan jika ditemukan adanya kinerja yang di bawah standar. Kontrol dipasang untuk meningkatkan kinerja sistem, mencegah atau mendeteksi penyalahgunaan atau kesalahan sistem dan menjamin keamanan data.

5. Analisis efisiensi (*efficiency*)

   Efisiensi berhubungan dengan bagaimana sumber tersebut digunakan dengan pemborosan yang minimal. Oleh karena itu, masalah efisiensi membutuhkan peningkatan output/hasil. Karena sistem yang ada telah dapat di daya gunakan dengan baik dan juga telah dapat menghasilkan output seusai dengan yang diharapkan.

6. Analisis Pelayanan (*service*)

   Pelayanan yang baik dapat mencerminkan suatu lembaga itu baik atau tidak baik, sehingga pelayanan harus juga diperhitungkan secara baik.

Analisis pieces juga di gunakan sebagai bahan pengujian untuk mengetahui tingkat kepuasan pengguna terhadap sistem yang di buat. Hasil di dapat

berdasarkan kuesioner yang telah di berikan penulis dengan kriteria PIECES dan diberikan nilai berdasarkan Skala Likert dan mendapatkan hasil nilai berdasarkan skala Kaplan dan Norton (Yuli Asbar).

2.11 Pengujian Sistem (Metode *Blackbox Testing*)

Blackbox testing dilakukan tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. Juga disebut sebagai behavioral testing, *specification-based testing input/output testing* atau *functional testing* (Jadnika & Irwan). Dengan adanya *blackbox testing*, perekayasa *software* dapat menggunakan sekumpulan kondisi masukan yang dapat secara penuh memeriksa keseluruhan kebutuhan fungsional pada suatu program. Kategori *error* yang akan diketahui melalui *blackbox testing*:

1. Fungsi yang hilang atau tidak benar
2. *Error* dari antar muka
3. *Erorr* dari struktur data atau akses eksternal database
4. *Error* dari kinerja atau tingkah laku
5. *Error* dari insialisasi dan terminasi

2.12 Sejarah Singkat CV Simpur

Simpur Poultry Shop pertama kali didirikan pada tanggal 12 April 1989 dengan nama perusahaan CV Simpur dan nama dagang Simpur Poultry Shop dengan tujuan untuk memperoleh laba yang sebesar-besarnya dan menjaga kelangsungan hidup perusahaan. Oleh karena itu perusahaan mengutamakan hubungan baik dengan pelanggan.

Perusahaan ini didirikan karena adanya perkembangan dalam hal permintaan pasar akan kebutuhan bahan makanan daging ayam broiler maupun peternak telur ayam ras pada saat itu sejak masuknya industri pabrik peneglolaan makanan sehingga memungkinkan untuk membuka distributor peternakan sebagai jasa penyalur antara pabrik makanan ternak dengan para pengusaha ayam.

Simpur poultry Shop dalam melakukan penjualan selalu menambahkan produk produk yang dijualnya dengan produk jenis baru yang kualitasnya terjamin. Sampai saat ini,barang yang dijual atau didistribusikan oleh perusahaan adalah:

1. Pakan ternak (ayam potong dan petelur)
2. Pakan burung
3. Pakan udang
4. Pakan ikan
5. Pakan babi
6. Pakan sapi
7. Obat, vitamin dan vaksin (untuk ayam potong dan petelur)
8. Telur ayam ras dan telur ayam kampung
9. Bibit ayam broiler (DOC) dan bibit bebek (DOD)

Dari masing-masing barang tersebut dibagi menjadi beberapa bagian yang berbeda untuk mengelola stoknya. Seperti pada bagiang makanan, maka hanya akan mengelola data stok barang makanan seperti pakan ikan, pakan ayam, pakan sapi dan babi. Lalu pada bagian lain seperti obat, vitamin dan vaksin hanya mengelola data tersebut.

2.12.1 Logo Perusahaan

Simpur Poltry Shop memiliki logo sebagai berikut:

Gambar 2.1 Logo Simpur Poultry Shop

Sumber: CV Simpur