

BAB II KAJIAN PUSTAKA

II.1 Tinjauan Pustaka

Beberapa penelitian yang terkait sebagai bahan pertimbangan untuk penelitian yang akan dilakukan mengenai rancang bangun sistem inventory menggunakan metode first in first out antara lain :

Jurnal yang berjudul Telaah Sistem Informasi Persediaan Obat Menggunakan Metode FIFO Pada Apotek Adya Sukabumi, Penelitian ini membahas, Hasil penelitian ini adalah penggunaan sistem yang telah terkomputerisasi merupakan solusi yang tepat guna membantu dalam proses pengolahan data, dikarenakan sistem terkomputerisasi akan membantu Apotek Adya mempercepat pengolahan data termasuk penyimpanan data dan pencarian data serta mempercepat dalam pembuatan laporan. Sistem terkomputerisasi sedikitnya akan mengurangi kesalahan yang disebabkan oleh *human error*, data-data tersimpan dengan rapi, dan mempercepat proses pengaksesan data. Sistem komputerisasi akan membantu Apotek Adya dalam menangani proses pemesanan, penerimaan, dan pengeluaran obat yang setiap harinya selalu bertambah jumlahnya (Apriliani dkk, 2013)

Jurnal yang berjudul perancangan sistem informasi *follow up* paket *undelivery outbound* menggunakan metode FIFO di *customer service* PT. TIKI JNE Bandung, Penelitian ini membahas metode yang digunakan untuk melakukan proses *follow up* paket *undelivery outbound* oleh perusahaan adalah metode first in first out (FIFO), dimana metode ini akan mudah digunakan apabila hanya ada beberapa penerimaan informasi paket yang berkendala dari beberapa sumber yang diterima. Hasil penelitian ini adalah Perancangan ini merupakan peralihan dari sistem manual ke sistem komputer, sehingga proses *follow up* paket *undelivery outbound* akan dapat berjalan lancar sesuai dengan keperluan atau kebutuhan pemakai. Sistem Informasi *Follow Up* Paket *Undelivery Outbound* akan berjalan dengan baik jika data yang di input dengan benar serta penggunaannya dilakukan secara baik dan benar pula sesuai dengan kinerja dari sistem *follow up* paket

undelivery outbound yang telah diperbaharui, sehingga sistem informasi yang dihasilkan dapat lebih akurat (Victor dan Setia, 2014).

Telaah diatas digunakan sebagai bahan perbandingan antara penelitian yang sudah dilakukan dan yang akan dirancang oleh peneliti. . Penelitian ini kurang lebih seperti penelitian sebelumnya

II.2 Landasan Teori

II.2.1 Konsep Dasar Perancangan

Perancangan adalah proses pengembangan spesifikasi sistem baru berdasarkan rekomendasi hasil, Dalam tahap analisis sistem perancangan, tim kerja harus merancang dalam berbagai kertas dengan kebutuhan pengguna (*end user*) melalui alat perancangan sistem yang terstandarisasi. Kertas kerja yang dimaksud memuat berbagai uraian mengenai *input*, proses, dan *output* dari sistem yang akan diusulkan (Nasril & Saputra, 2016).

II.2.2 Sistem

Menurut (Yuliawan dkk, 2013) Sistem dapat didefinisikan dengan pendekatan proseduran pendekatan komponen. Dengan pendekatan prosedur, sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu. Contoh sistem yang didefinisikan dengan pendekatan prosedur ini adalah sistem akuntansi. Sistem ini didefinisikan sebagai kumpulan dari prosedur-prosedur penerimaan kas, pengeluaran kas, penjualan, pembelian dan buku besar. Sedangkan dengan pendekatan komponen, sistem dapat didefinisikan sebagai kumpulan dari komponen yang saling berhubungan satu dengan yanglainnya membentuk satu kesatuan untuk mencapai tujuan tertentu. Contoh sistem yang didefinisikan dengan pendekatan ini misalnya adalah sistem komputer yang didefinisikan sebagai kumpulan dari perangkat keras dan perangkat lunak.

II.2.3 Konsep Dasar Sistem

Definisi sistem adalah sekumpulan hal atau kegiatan atau elemen atau subsistem yang saling bekerja sama atau yang dihubungkan dengan cara-cara

tertentu sehingga membentuk satu kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan (Lemantar dkk, 2013).

Ada beberapa elemen yang membentuk sebuah sistem yaitu:

- 1) Tujuan artinya motivasi yang mengarahkan pada sistem, karena tanpa tujuan yang jelas sistem tidak terarah dan tidak terkendali.
- 2) Masukan artinya segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses.
- 3) Proses artinya bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna dan lebih bernilai.
- 4) Keluaran artinya hasil dari pemrosesan sistem dan keluaran dapat menjadi masukan untuk subsistem lain.
- 5) Batas sistem artinya pemisah antara sistem dan daerah di luar sistem.
- 6) Mekanisme pengendali dan umpan balik artinya mekanisme pengendalian diwujudkan dengan menggunakan umpan balik sedangkan umpan balik digunakan untuk mengendalikan masukan maupun proses.
- 7) Lingkungan artinya segala sesuatu yang berada di luar sistem.

II.2.4 inventory

Inventory atau persediaan adalah aktiva yang tersedia untuk dijual dalam kegiatan normal persediaan. Pada bisnis manufaktur, persediaan meliputi bahan mentah, barang dalam proses produksi, barang jadi (Dewayani & Wahyuningsih, 2016)

Inventory dapat di klasifikasikan, ada jenis perusahaan yang membeli barang akan dijual lagi, dan ada juga perusahaan yang mengolah bahan mentah menjadi bahan jadi. Beberapa macam klasifikasi *inventory* yang digunakan oleh perusahaan, antara lain: *Inventory* Produksi, *Inventory* MRO (*Maintenance, Repair, dan Operating supplies*), *Inventory In-Proces*, dan *Inventory Finished-goods* (Rohayati, 2014).

- 1) Klasifikasi *Inventory* produksi adalah bahan baku dan bahan-bahan lain yang digunakan dalam proses produksi dan merupakan bagian dari produk. Bisa terdiri dari dua tipe yaitu item spesial yang dibuat khusus untuk

spesifikasi perusahaan dan item standart produksi yang dibeli secara off-the-self.

- 2) Katagori *Inventory MRO (Maintenance, Repair, dan Operating supplies)* adalah barang-barang yang digunakan dalam proses produksi namun tidak merupakan bagian dari produk, seperti pelumas dan pembersih.
- 3) Katagori *Inventory In-Process* adalah produk setengah jadi. Produk yang termasuk dalam katagoriinventori ini bisa ditemukan dalam berbagai proses produksi Semua produk jadi yang siap untuk dipasarkan termasuk dalam katagori *inventory finished-goods*. Vio Hotel Indonesia adalah sebuah perusahaan yangmemiliki produk-produk yang siap untuk dipakai,tidak ada proses pengolahan.

II.2.5 Metode FIFO

metode FIFO adalah barang dalam persediaan yang pertama dibeli akan dijual atau digunakan terlebih dahulu sehingga yang ada dalam persediaan akhir adalah yang dibeli atau yang diproduksi kemudian (Dewayani & Wahyuningsih, 2016).

Dengan FIFO, biaya inventori diperhitungkan dalam barang yang siap dijual atau dikonsumsi yang sudah ada lebih lama dan hal itu berarti stock yang tersedia adalah pembelian yang paling lama atau paling dulu diproduksi dan unit yang digunakan akan dibebankan pada harga dari barang yang terlama.

Metode yang dapat digunakan ada 2 (Rahman & Bagio, 2016) yaitu :

- 1) Perpetual

Dengan cara ini setiap keluar atau masuknya barang dapat diketahui dengan pasti dan terinci, karena selalu dicatat setiap jenis barang yang keluar atau masuk, serta biaya bahan yang dikeluarkan. Dengan cara ini dapat memudahkan kita dalam melakukan pengecekan terhadap keluar masuknya barang.

- 2) Periodik

Dengan cara periodik persediaan barangdapat diketahui dalam satu periode tertentu, namun dengan cara ini keluar masuknya barang tidak dapat

diketahui dengan rinci, karena dalam pencatatan hanya masuknya barang saja yang dicatat. Untuk mengetahui berapa biaya bahan baku yang dipakai dalam produksi harus dilakukan dengan cara menghitung sisa persediaan bahan baku yang masih ada digudang pada akhir periode akuntansi. Harga pokok persediaan awal ditambah harga pokok bahan baku yang dibeli dikurangi harga pokok persediaan bahan baku yang masih ada pada akhir periode adalah biaya-biaya bahan baku yang dipakai.

II.2.6 Basis Data

1) Pengertian Basis Data

Basis data merupakan kumpulan dari item data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, yang kelak dapat dimanfaatkan kembali dengan cepat dan mudah (Nugrahanti, 2015).

Data-data yang saling berkaitan satu sama lain disimpan dalam suatu tempat yang kemudian diolah sedemikian rupa sehingga dapat menghasilkan suatu informasi yang bermanfaat untuk pengguna. Data-data tersebut direpresentasikan dalam bentuk file-file, dimana file tersebut berisikan record-record yaitu rekaman yang menggambarkan suatu individu tertentu, sedangkan field merupakan suatu atribut dari record yang sejenis yang menunjukkan suatu item dari data.

2) *Database Management System* (DBMS)

Untuk mengelola basis data diperlukan suatu perangkat lunak yang disebut dengan DBMS. *Database Management System* (DBMS) atau Sistem Manajemen Basis Data adalah perangkat lunak yang di desain untuk membantu menangani koleksi data dalam jumlah besar yang dibutuhkan dalam sebuah sistem dengan pertumbuhan data yang sangat cepat. DBMS umumnya merupakan sebuah bagian dari komputer sains, tujuan penggunaan dan teknik penggunaannya sangat luas, seperti pada bahasa pemrograman, pemrograman berorientasi objek, sistem operasi, struktur data, pemrograman konkuren, kecerdasan buatan, dan masih banyak lagi.

Dengan adanya DBMS suatu basis data dapat dikelola dengan baik dan diproses dengan baik dalam sistem sesuai dengan kebutuhan pengguna.

Adapun keuntungan dari menggunakan system manajemen basis data atau DBMS (Maanari dkk, 2013), antara lain :

a) Independensi data

DBMS mampu mengatur independensi detail representasi dan penyimpanan data dari aplikasi, dengan cara mengabstraksikan data untuk mengisolasi kode program dalam mengakses detail-detail.

b) Efisiensi data

DBMS mampu memberikan akses data secara efisien, terlebih jika data tersebut berada pada penyimpanan eksternal, seperti pada sebuah server basis data.

c) Integritas dan Keamanan Data

Data yang diakses melewati sebuah DBMS harus mengikuti aturan yang ditentukan tentunya ini menjamin integritas dari data yang digunakan dalam aplikasi, demikian juga dengan hak akses data yang berbeda untuk tiap jenis pengguna data diatur oleh DBMS untuk menjamin keamanan data yang dikelola.

d) Administrasi Data

Ketika beberapa user memproduksi data, sangat diperlukan proses administrasi data yang tersentralisasi. Seorang profesional yang mengerti mengenai pentingnya data tersebut diatur dan bagaimana kelompok *user* mengaksesnya, sangat dibutuhkan dalam menangani administrasi data untuk mencegah redudansi data maupun untuk melakukan berbagai konfigurasi yang membuat pengambilan data menjadi lebih efisien.

e) Akses Konkuren dan Pemulihan Kerusakan

DBMS mengatur akses data konkuren sedemikian rupa sehingga user merasa mengakses data yang digunakan hanya oleh 1 *user*. Lebih jauh lagi sebuah DBMS juga mampu melindungi user dari sebuah kerusakan sistem.

f) Mengurangi Waktu Pengembangan Sebuah Perangkat Lunak

DBMS memiliki berbagai macam fungsi yang biasanya digunakan oleh aplikasi-aplikasi untuk mengakses data yang tersimpan. Dengan kemudahan tersebut pembangunan perangkat lunak tentunya menjadi lebih cepat dan lebih

handal, karena beberapa kegiatan pemrosesan data telah dilakukan oleh DBMS itu sendiri.

II.2.7 Visual Basic

II.2.7.1 Pengertian visual basic

Visual basic adalah program untuk membuat aplikasi berbasis Microsoft Windows secara cepat dan mudah. Visual basic menyediakan tool untuk membuat aplikasi yang sederhana sampai aplikasi kompleks atau rumit baik untuk keperluan pribadi maupun untuk keperluan perusahaan/instansi dengan sistem yang lebih besar.(Akbar & Dahlan, 2013).

Visual basic berbasiskan prinsip pemrograman berbasis objek/OOP dan dikembangkan denganbasis visual yang berarti menggunakan sarana grafis untuk mengembangkannya. Dalam pemrogramanberbasis objek/OOP, sebuah program dibagi menjadi bagian-bagian kecil yang disebut dengan OBJEK. Setiap objek memiliki entiti yang terpisah dengan entiti objek-objek yang lain. Masing-masing Objek dapat diolah sendiri-sendiri, dan mempunyai sekumpulan sifat dan metode yang melakukan fungsi tertentu sesuai dengan yang telah diprogramkan kepadanya.(Akbar & Dahlan, 2013).

II.2.7.2 Keunggulan Visual Basic 6.0

Keunggulan menggunakan Microsoft Visual Basic 6.0 menurut (Sunandar dkk, 2012), antara lain :

- 1) Menggunakan platform pembuatan program yang diberi nama developer studio, yang memiliki tampilan dan sarana yang sama dengan Visual C++ dan Visual J++
- 2) Memiliki kompiler handal yang dapat menghasilkan file executable yang lebih cepat dan lebih efisien.
- 3) Tambahan sarana wizard. Wizard adalah sarana yang mempermudah di dalam pembuatan aplikasi dengan otomatisasi tugas-tugas tertentu.
- 4) Tambahan tombol-tombol baru yang lebih canggih serta meningkatkan kaidah struktur Bahasa Visual Basic.
- 5) Kemampuan membuat ActiveX dan fasilitas internet yang lebih banyak.

- 6) Sarana akses data yang lebih cepat dan andal untuk membuat aplikasi database yang berkemampuan tinggi.
- 7) Visual Basic 6.0 memiliki beberapa versi atau edisi yang disesuaikan dengan kebutuhan pemakainya
- 8) Visual Basic disertai dengan berbagai sarana untuk membuat aplikasi Database, sarana database Visual Basic yang menjadikannya lingkungan terbaik untuk mengembangkan aplikasi client/server.

II.2.8 Microsoft Acces 2010

Microsoft Access 2010 melanjutkan versi Microsoft Access 2007, dengan sedikit penambahan hal baru dan perubahan tampilan. Pada versi ini mulai diperkenalkan 'Web Database' dengan menggunakan *Microsoft sharePoint Server* dan *Access Services* (Nahlah & Amiruddin, 2015)

Fitur Microsoft Access 2010 dengan sejumlah perbaikan membuat proses pembuatan dan pengelolaan database menjadi lebih mudah dan cepat dari sebelumnya. Jika Anda sudah akrab dengan program Microsoft Access 2007, ribbon dan jendela navigasi bukan hal baru bagi pengguna karena Microsoft Access 2010 masih menggunakan fitur tersebut. *Ribbon* menggantikan menu dan *toolbar* dari versi sebelumnya, sedangkan jendela navigasi menggantikan dan memperluas fungsionalitas dari jendela database. Dan yang baru dari Microsoft Access 2010, *Backstage View* yang berisi semua perintah yang berlaku untuk seluruh database, seperti membuka dan membuat database, kompak dan perbaikan, atau perintah yang datang dari menu file.

II.2.9 Crystal Report

Crystal Report merupakan peranti standar untuk pembuatan laporan pada sistem operasi Windows, dimana cetakan/template laporan yang dihasilkan dapat disertakan pada banyak bahasa pemrograman (Elizabeth & Dermawan, 2015)

Crystal Report terdiri dari tiga bagian utama, yaitu:

- a. Toolbox, yang berfungsi untuk menambahkan objek-objek ke dalam report designer.
- b. Field Explorer, yang berfungsi untuk menampilkan daftar field, formula, dan pernyataan pernyataan SQLsertayanglainnya.

- c. Report Designer, yang berfungsi untuk meletakkan objek-objek yang digunakan pada laporan.

II.2.10 Metodologi Pengembangan Sistem

Metodologi pengembangan yang digunakan pada sistem pendukung keputusan ini adalah menggunakan metodologi pengembangan model *Waterfall*. Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, desain, coding, testing/verification, dan maintenance. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan (Pascapraharastyan dkk, 2014). Sebagai contoh tahap desain harus menunggu selesainya tahap sebelumnya yaitu tahap *requirement*.

Tahap-tahap pengembangan sistem dengan model *waterfall* (Romadhoni dkk, 2015) adalah :

1) *Analysis*

Pada tahap ini, merupakan proses analisa kebutuhan sistem. Pengembang mengumpulkan data-data sebagai bahan pengembangan sistem. Pengumpulan data dapat dilakukan dengan teknik wawancara, teknik observasi, dan teknik kuisioner. Untuk menganalisis sistem yang telah ada agar dapat disusun sistem baru yang lebih baik. Analisis terstruktur biasanya akan menggunakan alat (*tool*) untuk merancang sistem baru. *Tool* yang digunakan untuk merancang sistem secara logis dapat digambarkan dengan UML (*Unified Modeling Language*).

2) *Design*

Proses desain adalah proses multi langkah yang berfokus pada empat atribut, yaitu : struktur data, arsitektur perangkat lunak, *representasi interface*, dan detail prosedural. Proses desain menterjemahkan hasil analisis ke dalam representasi perangkat lunak.

3) *Code*

Pada tahap ini desain diterjemahkan ke dalam program perangkat lunak. Pada tahap pengimplementasian ke dalam kode program akan bergantung pada hasil desain perangkat lunak pada tahap sebelumnya.

4) Test

Setelah pengkodean, dilanjutkan dengan pengujian terhadap sistem yang telah dibuat. Pengujian dilakukan untuk mengetahui kesesuaian hasil *output* dari sistem dengan kebutuhan yang telah dirancang pada tahap analisis. Salah satu jenis pengujian yang dilakukan kepada pengguna adalah pengujian *black box*. Pengertian pengujian *black box* dan aspek yang akan diujikan kepada pengguna akan dijelaskan lebih rinci sebagai berikut :

a) Pengujian *Black Box*

Pengujian *Black Box* lebih berfokus pada spesifikasi fungsional dari perangkat lunak. Penguji dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengujian pada spesifikasi fungsional program (Mustaqbal dkk, 2015).

Pengujian *Black Box* cenderung untuk menemukan hal-hal berikut:

- 1) Fungsi yang tidak benar atau tidak ada.
- 2) Kesalahan antarmuka (*interface errors*).
- 3) Kesalahan pada struktur data dan akses basis data.
- 4) Kesalahan performansi (*performance errors*).
- 5) Kesalahan inisialisasi dan terminasi.

Pengujian didesain untuk menjawab pertanyaan-pertanyaan berikut:

- 1) Bagaimana fungsi-fungsi diuji agar dapat dinyatakan valid?
- 2) Input seperti apa yang dapat menjadi bahan kasus uji yang baik?
- 3) Apakah sistem sensitif pada input-input tertentu?
- 4) Bagaimana sekumpulan data dapat diisolasi?
- 5) Berapa banyak rata-rata data dan jumlah data yang dapat ditangani sistem?
- 6) Efek apa yang dapat membuat kombinasi data ditangani spesifik pada operasi sistem?

b) Kualitas Perangkat Lunak

Kualitas perangkat lunak didefinisikan sebagai konfirmasi terhadap kebutuhan fungsional dan kinerja yang dinyatakan secara eksplisit, standar perkembangan yang didokumentasikan secara eksplisit dan karakteristik implisit

yang diharapkan bagi semua perangkat lunak yang dikembangkan secara profesional.

Kualitas perangkat lunak adalah keberadaan karakteristik dari suatu produk yang dijabarkan dalam kebutuhannya, artinya kita harus melihat terlebih dahulu karakteristik-karakteristik apa yang berhubungan atau tidak dengan kebutuhan-kebutuhan yang diinginkan oleh pemakai. Karakteristik yang dimaksud yaitu *contra-productive characteristics* dan *neutral characteristic* (Hapsari dan husen, 2015).

Faktor-faktor kualitas perangkat lunak (Hapsari dan Husen, 2015) terdiri dari :

- 1) Kebenaran adalah tingkat dimana program memenuhi spesifikasinya dan memenuhi sasaran misi karyawan.
- 2) Reliabilitas adalah tingkat dimana sebuah program dapat diharapkan melakukan fungsi yang diharapkan dengan ketelitian yang diminta.
- 3) Efisiensi adalah jumlah sumber daya penghitungan kode yang diperlukan oleh program untuk melakukan fungsinya.
- 4) Integritas adalah tingkat dimana akses ke perangkat lunak atau data oleh orang yang tidak berhak dapat di kontrol.
- 5) Usabilitas adalah usaha yang dibutuhkan untuk mempelajari, mengoperasikan, menyiapkan input, dan menginterpretasikan output suatu program.
- 6) Maintanabilitas adalah usaha yang diperlukan untuk mencari dan membetulkan kesalahan pada sebuah program.
- 7) Fleksibilitas adalah usaha yang diperlukan untuk memodifikasi program operasional.
- 8) Testabilitas adalah usaha yang diperlukan untuk menguji sebuah program untuk memastikan apakah program melakukan fungsi-fungsi yang dimaksudkan.
- 9) Portabilitas adalah usaha yang diperlukan untuk memindahkan program dari satu perangkat keras dan atau lingkungan.

10) Reusabilitas adalah tingkat dimana sebuah program (bagian dari suatu program) dapat digunakan kembali di dalam aplikasi lain.

11) Interperabilitas adalah usaha yang diperlukan untuk merangkai satu sistem dengan yang lainnya.

II.2.11 UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. (Hendini,2016).

UML dapat digunakan untuk membuat model segala jenis aplikasi perangkat lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Karena konsep dasar UML menggunakan *class* dan *operation*, maka lebih cocok untuk penulisan perangkat lunak dalam bahasa berorientasi objek.

Notasi UML adalah sekumpulan bentuk khusus yang menggambarkan berbagai diagram piranti lunak, dan tiap-tiap bentuk memiliki makna tertentu. Notasi UML merupakan gabungan dari 3 notasi yang telah ada sebelumnya yaitu *Grady Booch OOD*, *Jim Rumbaugh OMT*, dan *Ivar Jacobson OOSE*.

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan merupakan notasi dari metode yang digunakan untuk mendesain secara cepat. Bagian utama dari UML adalah *view*, diagram, model elemen dan *general mechanism*.

1. View

Berfungsi untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan digunakan untuk melihat grafik, tetapi *view* merupakan suatu abstraksi yang berisi sejumlah diagram. Jenis-jenis *view* antara lain *use*

case view, logical view, komponen view, concurrennses view dan deployment view.

a. *Use case view*

Use case diagram berfungsi untuk memahami dan menggunakan sistem yang dimodelkan. *View* ini melihat pada bagaimana *actor* dan *use case* berinteraksi.

b. *Logical view*

Logical view merupakan persyaratan (*requirements*) fungsional sistem. *View* ini melihat pada kelas-kelas dan hubungan antar kelas-kelas tersebut.

c. *Komponen view*

Componen view bertujuan untuk pengaturan software. *View* ini mengandung informasi mengenai komponen-komponen *software*, komponen tereksekusi (*executable*) dan *library* dan sistem yang dimodelkan.

d. *Deployment view*

Deployment view memperlihatkan pemetaan setiap proses ke dalam *hardware*. *View* ini sangat bermanfaat ketika membuat model suatu sistem yang diterapkan dalam lingkungan arsitektur yang terdistribusi yang diterapkan pada aplikasi dan server pada lokasi yang berbeda.

2. *Diagram*

Diagram berbentuk grafik menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan jika digambar biasanya dialokasikan untuk *view* tertentu.

Jenis-jenis diagram antara lain *use case diagram, class diagram, activity diagram*.

a. *Use case diagram*

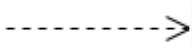
Use case diagram mendiskripsikan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah apa yang diperbuat sistem, dan bukan bagaimana. Sebuah *use case* mempresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan

tertentu, misalnya *login* ke sistem. Seorang/ sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram sangat membantu pada saat penyusunan *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat menginclude fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *usecase* yang di *include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal.

Sebuah *use case* dapat di *include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviournya* sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari *use case* yang lain.

Tabel II.1 Daftar simbol *Use Case Diagram*

No	Gambar	Nama	Keterangan
1		Actor	Actor adalah pengguna system. Actor tidak terbatas pada manusia saja, jika sebuah system berkomunikasi dengan aplikasi lain dan membutuhkan input atau meberikan output, maka aplikasi tersebut bisa juga dianggap sebagai actor.
2		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan

			mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		Generalization	Hubungan dimana objek anak (desendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk.
4		Include	Menspesifikasikan bahwa usecase sumber secara eksplisit.
5		Extend	Menspesifikasikan bahwa usecase target memperkuat perilaku dari usecase sumber pada suatu titik diberikan.
6		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		System	Memspezifikasikan paket yang menampilkan system secara terbatas.
8		Use case	Deskripsi dari urutan aksi-aksi yang ditampilkan system yang menghasilkan suatu hasil yang terukur bagi suatu actor.
9		Collaboration	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (bersinergi).

10		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.
----	---	------	--

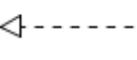
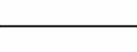
b. *Class diagram*

Class adalah sebuah spesifikasi yang apabila di instansiasikan akan menghasilkan objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan atribut atau properti suatu sistem dan memberikan layanan untuk memanipulasi keadaan tersebut (fungsi/metoda). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi dan lain-lain.

Class dapat berupa implementasi sebuah *interface*, yaitu kelas abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian interface mendukung resolusi metoda pada saat *run-time*.

Table II.2 Daftar simbol *class diagram*

No	Gambar	Nama	Keterangan
1		Generalization	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang berada di atasnya objek induk (ancestor).
2		Nary association	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.

3		Class	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		Collaboration	Deskripsi dari urutan aksi-aksi yang ditampilkan system yang menghasilkan suatu hasil yang terukur bagi suatu actor.
5		Realization	Operasi yang benar-benar dilakukan oleh suatu objek.
6		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7		Association	Apa yang menghubungkan antara objek satu dengan objek.

c. *Activity diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state* khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar transisi di *trigger* oleh selesainya *state* sebelumnya (*internal processing*).

Table II.3 Daftar simbol *Activity Diagram*

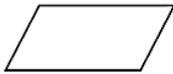
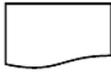
SIMBOL	NAMA	KETERANGAN
	<i>Start state</i>	Titik awal atau permulaan
	<i>End state</i>	Titik akhir atau akhir dari aktivitas
	<i>Activity</i>	<i>Activity</i> atau aktivitas yang dilakukan oleh actor
	<i>Decision</i>	Pilihan untuk mengambil keputusan
	<i>Interaction</i>	Alur
	<i>Fork</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu

II.2.12 System Flow

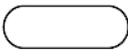
System Flow merupakan urutan-urutan langkah kerja suatu proses yang digambarkan dengan menggunakan simbol-simbol yang disusun secara sistematis. *System Flow* menjelaskan urutan prosedur yang akan diterapkan dalam sistem meliputi media input, output, dan jenis media penyimpanan dalam proses pengolahan data (Iswandy, 2015). Tujuannya adalah agar tahap-tahap penyelesaian dapat digambarkan dengan sederhana, jelas dan rapi. *System flow* biasanya menggunakan simbo-simbol tertentu, namun tidak menutup kemungkinan bagi pemrogram untuk membuat simbol sendiri jika simbol yang telah tersedia dirasa kurang, namun harus dilengkapi dengan kamus simbol untuk menjelaskan arti dari masing-masing simbol.

Simbol-simbol dalam *system flow* antara lain terdiri dari simbol *input* dan *output*, proses, dan simbol penghubung (Taupik dkk, 2013).

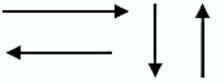
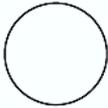
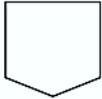
Tabel II.4 Simbol *Input dan Output Flow*

Simbol Input dan Output		
	<i>Input-Output</i>	Simbol yang menandakan proses input dan output
	<i>Document</i>	Simbol yang menyatakan input dan output berupa dokumen dalam bentuk cetak/kertas
	<i>Disk and On-line Storage</i>	Simbol untuk menyatakan input berasal dari disk atau output di simpan ke disk

Tabel II.5 Simbol Proses *System Flow*

Simbol Proses		
	<i>Process</i>	Simbol yang menunjukkan proses yang dilakukan komputer
	<i>Decision</i>	Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban / aksi
	<i>Predefined Process</i>	Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan didalam storage
	<i>Terminal</i>	Simbol untuk permulaan atau akhir dari suatu program
	<i>Manual Input</i>	Simbol untuk pemasukan data secara manual on-line keyboard

Tabel II.6 Simbol Penghubung *System Flow*

Simbol Penghubung		
	Arus / Flow	Penghubung antar proses
	<i>Connector</i>	Simbol keluar / masuk prosedur atau proses dalam lembar / halaman yang sama
	<i>Off-line Connector</i>	Simbol keluar / masuk prosedur atau proses dalam lembar / halaman yang lain

II.3 Tinjauan Umum PT. ESSEI PERBAMA

II.3.1 Sejarah Perusahaan

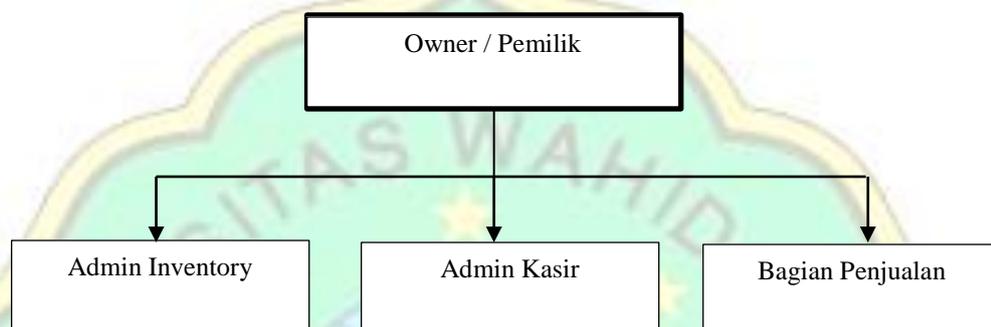
PT Essei Perbama adalah sebuah perusahaan yang sedang berkembang yang bergerak dibidang distribusi Sari roti dan jasa rekrutment tenaga kerja tanpa adanya pungutan biaya.

Pada tahun 2008 PT. Essei Perbama berdiri sebagai perusahaan rekrutment tenaga kerja yang bekerja sama dengan PT. Nippon Indosari Corpindo TBK (Sari Roti). Untuk memenuhi permintaan konsumen yang terus meningkat, PT. Essei Perbama mengembangkan usahanya dengan mendirikan distributor Sari Roti di Bekasi pada tahun 2012. Besarnya permintaan masyarakat atas produk Sari Roti membuat PT. Essei Perbama kembali melebarkan sayapnya untuk membuka beberapa cabang distributor sari roti di Provinsi Jawa Barat pada tahun 2013. Pada tahun 2014, PT. Essei Perbama mendirikan distributor baru yang berlokasi di provinsi Jateng & DIY.

II.3.2 Struktur Organisasi

Suatu usaha yang diorganisir dan dikelola untuk menyediakan barang dan jasa bagi masyarakat untuk mendapatkan keuntungan, diperlukan suatu struktur organisasi yang baik. Karena dengan struktur organisasi yang baik dapat merencanakan pembagian tugas serta wewenang kepada karyawan perusahaan sesuai dengan bidang dan tanggung jawabnya.

Struktur organisasi merupakan gambaran struktur kerja dan sebuah organisasi, dimana didalamnya menunjukkan hubungan wewenang dan tanggung jawab serta deskripsi pekerjaan yang harus dilakukan. Jadi struktur organisasi menggambarkan kedudukan masing masing jabatan yang terdapat dalam perusahaan sehubungan dengan tanggung jawab yang ada pada dirinya. Struktur organisasi Distributor Sari Roti Semarang adalah sebagai berikut:



Gambar II.1 Struktur Organisasi

II.3.3 Job Description

Secara umum tugas dan fungsi organisasi dari masing-masing bagian adalah sebagai berikut :

1. Pimpinan

Pimpinan dimana segala keputusan dan kebijakan serta pengawasan jalannya usaha ditentukan olehnya.

2. Kasir

Bagian ini melaksanakan kegiatan yang berhubungan dengan urusan keuangan perusahaan serta pengadaan inventaris perusahaan dan juga melakukan pengadaan barang untuk keperluan produksi yang pelaksanaannya bekerjasama dengan bagian penjualan.

3. Bagian Penjualan

Bagian ini melakukan kunjungan ke outlet-outlet dan mengantar pesanan customer, dan bagian inilah yang akan melaporkan kepada bagian administrasi tentang semua hasil transaksi kepada bagian administrasi.

4. Admin inventory

Bagian ini bertugas melakukan pengelolaan yang didalamnya terdapat pengolahan data master barang , data transaksi barang , serta pengelolaan laporan data barang.

