

BAB II

KAJIAN LITERATUR

2.1 Konsep Sistem Informasi

Semua organisasi memiliki suatu sistem informasi. Sistem informasi ini merupakan satu kesatuan terdiri dari berbagai sumber logis dan fisik. Dari organisasi ke organisasi, sumber ini disusun atau terstruktur dalam sejumlah cara yang tak terhingga. Namun, karena sistem organisasi dan informasi merupakan sumber dinamis, suatu struktur yang kita konstruksi satu hari tidak perlu merefleksikan susunan aktual dari sumber-sumber ini ke hari berikutnya. Sehingga kita perlu suatu konsep yang secara logis menggambarkan struktur dari suatu sistem informasi, merefleksikan semua fisiknya, adalah tepat untuk setiap ukuran sistem informasi dalam setiap tipe organisasi, dan kembali konstan secara relatif.

Konsep sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*) yaitu :

1. Blok masukan (*input block*) *Input* mewakili data yang masuk ke dalam sistem informasi berupa dokumen dasar.
2. Blok model (*model block*) Blok ini terdiri dari kombinasi prosedur, logika dan metode matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data.
3. Blok keluaran (*output block*) Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas.
4. Blok teknologi (*technology block*) Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran.
5. Blok basis data (*basis data block*) Merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer.
6. Blok kendali (*control block*) Banyak faktor yang dapat merusak sistem informasi, misalnya bencana alam, kecurangan-kecurangan, dan sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atau bila terlanjur terjadi kesalahan dapat langsung diatasi. [3].

2.1.1 Sistem

Sistem dapat diartikan sebagai kumpulan dari elemen-elemen atau unsur-unsur berupa data, jaringan dari prosedur-prosedur yang saling berhubungan, sumber daya manusia, teknologi baik *hardware* maupun *software* yang saling berinteraksi sebagai satu kesatuan untuk mencapai tujuan atau sasaran tertentu yang sama [4].

Sistem dapat diartikan sebagai kumpulan objek atau unsur yang saling memiliki hubungan dan saling mempengaruhi satu sama lain serta memiliki keterkaitan pada rencana yang sama dalam mencapai suatu tujuan tertentu. Untuk membangun dan mengembangkan suatu sistem yang baik, maka perlu membedakan antara unsur-unsur dari sistem yang membentuknya. Berikut adalah uraian unsur-unsur sistem :

1. Komponen Sistem (*Component*) Suatu sistem terdiri dari beberapa komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.
2. Batasan Sistem (*Boundary*) Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya.
3. Lingkungan Luar Sistem (*Environment*) Lingkungan luar dari suatu sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem.
4. Penghubung Sistem (*Interface*) Penghubung sistem merupakan media penghubung antara suatu sistem dengan subsistem lainnya.
5. Masukan sistem (*Input*) Masukan adalah energi yang dimasukkan ke dalam sistem.
6. Keluaran sistem (*Output*) Keluaran adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.
7. Pengolah sistem (*Process*) Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.
8. Sasaran sistem (*Objective*) Suatu sistem pasti mempunyai tujuan atau sasaran. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem [5].

2.1.2 Informasi

Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang.

Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian nyata yang digunakan untuk pengambilan keputusan [6].

Data yang masuk akan diolah sehingga keluaran data tersebut akan berupa informasi, akan tetapi data yang diolah tersebut bisa saja tidak langsung menjadi informasi, tetapi disimpan dulu dalam tempat penyimpanan yang disebut Basis Data. Informasi juga tidak selalu diolah dari data yang baru dimasukkan, tetapi dapat pula dihasilkan dari data yang sudah ada di dalam basis data atau informasi didapat dari hasil gabungan antara data yang telah disimpan dengan data yang baru masuk [7].

Adapun pengertian informasi lebih singkatnya bahwa informasi merupakan sebuah data yang sudah diproses atau diolah menjadi sebuah *file* dalam bentuk paragraf maupun gambar-gambar. Informasi juga dikatakan sebagai hasil dari pengolahan data ke dalam bentuk yang lebih berguna lagi untuk penerimanya yang di dalamnya menggambarkan suatu kejadian-kejadian nyata dan mampu digunakan untuk alat bantu pengambilan suatu keputusan [8].

2.1.3 Sistem Informasi

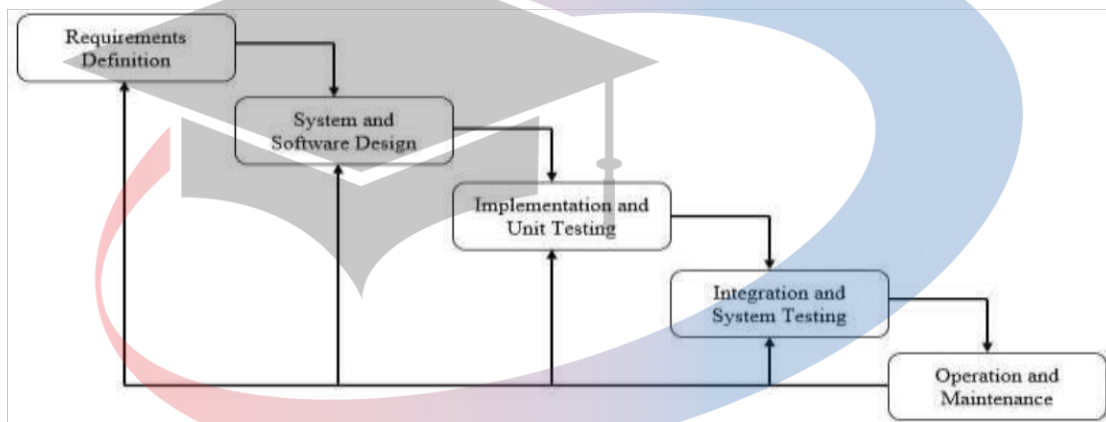
Sistem informasi adalah alat untuk mempermudah proses penggunaan data atau pemrosesan data pada suatu organisasi, institusi ataupun badan usaha, secara garis besar sistem informasi merupakan kombinasi dari perangkat keras, perangkat lunak dan jaringan telekomunikasi yang berguna untuk mempersingkat suatu pemrosesan data dalam berbagai kepentingan seperti mengumpulkan, mengubah dan mendistribusikan informasi pada suatu organisasi, institusi atau badan usaha [9].

Sistem Informasi juga merupakan kumpulan dari prosedur kegiatan yang memproses sebuah data sedemikian rupa untuk menyelesaikan permasalahan tertentu dengan cara melakukan pengolahan data dengan bantuan komputer sehingga dapat menghasilkan suatu informasi yang bermanfaat bagi orang lain agar dapat digunakan dalam mengambil suatu keputusan [10].

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan [11].

2.2 Metode *Waterfall*

Metode air terjun atau yang sering disebut metode *Waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), nama model ini sebenarnya adalah “*Linear Sequential Model*” di mana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna [12]. *Waterfall Model* mempunyai 5 tahap yaitu *Requirements Definition*, *System and Software Design*, *Implementation and Unit Testing*, *Integration and System testing*, dan *Operation and Maintenance* [13]. Model *Waterfall* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Model Waterfall (Ian Sommerville)

Model *Waterfall* seperti yang digambarkan pada gambar 2.1 terdapat beberapa proses yang dilalui. Dalam model *Waterfall Sommerville* terdapat kemungkinan untuk kembali ke tahap sebelumnya apabila terjadi kesalahan atau perbaikan.

1. *Requirement Definition*

Tahap awal di mana adanya analisis untuk menentukan kebutuhan, batasan, dan tujuan dari perangkat lunak sesuai yang diinginkan. Hal tersebut kemudian didefinisikan secara rinci dan terbentuk sebagai spesifikasi sistem.

2. *System and Software Design*

System design merupakan proses perancangan perangkat keras maupun perangkat lunak yang dilibatkan untuk menunjang sistem yang akan dibangun. Sementara itu *software design* merupakan proses perancangan yang melibatkan identifikasi dan menggambarkan dasar sistem serta hubungan satu sama lain.

3. *Implementation and Unit Testing*

Pada tahap ini, *software design* yang telah dilakukan sebelumnya kemudian diimplementasikan dalam bentuk unit program. Setelah unit program dibuat, kemudian dilakukan *testing* pada unit program tersebut untuk memastikan implementasi berjalan dengan baik.

4. *Integration and System Testing*

Setelah semua unit program berhasil diimplementasikan dan lolos *testing* maka dilanjutkan dengan mengintegrasikan setiap unit untuk membentuk sistem yang diinginkan. Sistem yang sudah dibentuk kemudian dites kembali untuk memastikan unit program dapat berjalan satu sama lain dalam sistem dan sistem yang dibuat sudah memenuhi kebutuhan.

5. *Operation and Maintenance*

Tahap ini merupakan tahap di mana sistem sudah dipasang kemudian melakukan perbaikan ketika terdapat kesalahan atau *error* yang tidak ditemukan sebelumnya saat pembangunan sistem berlangsung. Perbaikan juga dilakukan jika terdapat kebutuhan baru yang perlu ada pada sistem [14].



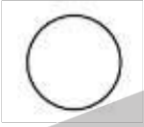


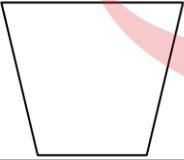



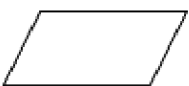

2.3 Teknik Pengembangan Sistem


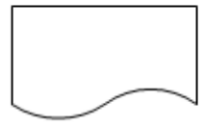
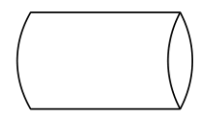
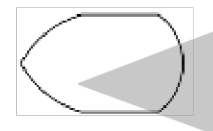
Pengembangan sistem dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau perbaikan pada sistem yang telah ada, dengan harapan bahwa sistem yang baru tersebut dapat mengatasi permasalahan yang timbul pada sistem yang lama. Sedangkan definisi lain menyebutkan pengembangan sistem adalah proses memodifikasi atau mengubah sebagian atau seluruh sistem informasi [15].

2.3.1 Flowchart

Flowchart merupakan sebuah diagram alir yang mempresentasikan rancangan dari langkah-langkah berbentuk perintah yang dibuat pada sistem. Penggunaan *flowchart* ini nantinya berbentuk dokumentasi yang berfungsi untuk menjelaskan gambaran dari sebuah sistem yang akan dibuat. *Flowchart* juga berguna untuk memberikan sebuah solusi terhadap masalah yang sedang dihadapi ketika membuat sistem. Gambaran dari *flowchart* sendiri yaitu berbentuk simbol-simbol dan garis penghubung yang setiap simbol serta garisnya mewakili masing-masing proses [28].

Tabel 2.1 Simbol Flowchart

Simbol	Nama	Keterangan
	<i>Flow Direction Symbol/Connecting Line</i>	Menyatakan arus suatu proses
	<i>Communication Link</i>	Merupakan fungsi untuk transmisi data dari satu lokasi ke lokasi lainnya.
	<i>Connector</i>	Menyatakan sambungan dari proses yang satu ke proses berikutnya di halaman yang sama
	<i>Offline Connector</i>	Menyatakan sambungan dari proses yang satu ke proses berikutnya di halaman yang berbeda
	<i>Processing</i>	Digunakan untuk menunjukkan pengolahan yang akan dilakukan dalam komputer
	<i>Manual Operation</i>	Digunakan untuk menunjukkan pengolahan yang tidak dilakukan oleh komputer
	<i>Decision</i>	Digunakan untuk memilih proses yang akan dilakukan berdasarkan kondisi tertentu
	<i>Predefined Process</i>	Digunakan untuk mempersiapkan penyimpanan yang sedang atau akan digunakan dengan memberikan harga awal
	<i>Terminator</i>	Digunakan untuk memulai atau mengakhiri program
	<i>Input / Output Data</i>	Menunjukkan proses <i>input / output</i> data, parameter, dan informasi.
	<i>Offline Storage</i>	Berfungsi untuk menunjukkan bahwa data akan disimpan di media tertentu

	<i>Manual Input Symbol</i>	Digunakan untuk menginputkan data secara manual dengan <i>keyboard</i>
	<i>Document</i>	Digunakan untuk menyatakan masuk dan keluar yang berasal dari dokumen
	<i>Direct data</i>	Simbol yg menyatakan <i>input</i> berasal dari <i>drum magnetic</i> atau <i>output</i> disimpan ke <i>drum magnetic</i> .
	<i>Display</i>	Digunakan untuk menyatakan keluar melalui layar monitor

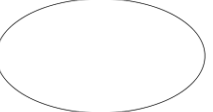
2.3.2 UML (Unified Modeling Language)

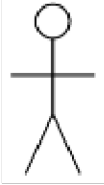


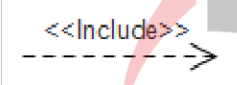
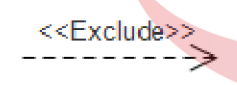
UML (*Unified Modeling Language*) merupakan salah satu standar visualisasi, perancangan dan permodelan yang banyak digunakan untuk mendefinisikan serta menggambarkan arsitektur dalam sebuah pemrograman berorientasi objek [16]. Jenis UML yang digunakan pada penelitian ini yakni :

1. *Use case diagram*

use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu [17].

Tabel 2.2 Simbol Use Case



Simbol	Nama	Keterangan
	<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau <i>actor</i> biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .






	<i>Actor</i>	Orang proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i> .
	<i>Association</i>	Komunikasi antara <i>actor</i> dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i> .
	<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> di mana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
	<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsional atau sebagai syarat dijalankan <i>use case</i> ini.
	<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.

2. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi [17].

Tabel 2.3 Simbol Class Diagram





Simbol	Nama	Keterangan
	<i>Generalization</i>	Hubungan di mana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.

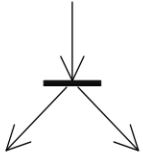

	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i>
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

3. Activity diagram

Activity diagram merupakan analisis dengan kemampuan untuk memodelkan proses dalam suatu sistem informasi. *Activity diagram* dapat digunakan untuk alur kerja model, *use case* individual, atau logika keputusan yang terkandung dalam metode individual. *Activity diagram* juga menyediakan pendekatan untuk proses pemodelan *parallel* [18].

Tabel 2.4 Simbol Activity Diagram

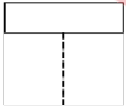


Simbol	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas
	<i>End Point</i> , akhir aktivitas
	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i>

	<p><i>Fork</i>/percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu</p>
	<p><i>Swimlane</i>, Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.</p>

4. Sequence diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. *Sequence diagram* menunjukkan urutan *event* kejadian dalam suatu waktu. Komponen *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segi empat bernama [18].

Tabel 2.5 Simbol Sequence Diagram

Simbol	Nama	Keterangan
	<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi - informasi tentang aktivitas yang terjadi
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi - informasi tentang aktivitas yang terjadi

2.3.3 Konsep Dasar UML

UML terdiri dari beberapa elemen yang membentuk diagram dengan aturan tertentu. Diagram ini bertujuan untuk menggambarkan sistem dari berbagai sudut pandang. Relasi *Unified Modelling Language* menggambarkan hubungan antar object di dalam suatu sistem. Beberapa relasi dalam UML yaitu :

1. *Dependency*, yaitu suatu relasi di mana suatu objek tergantung dengan pada objek yang lain. Perubahan pada objek tersebut berpengaruh terhadap objek lain. Relasi digambarkan sebagai garis terputus-putus yang memiliki arah menunjuk pada suatu objek.
2. *Generalization*, yaitu relasi antar objek yang umum (*parent*) dengan sesuatu yang lebih spesifik (*child*) di mana objek yang lebih khusus dapat digantikan dengan objek yang lebih umum. Generalisasi digambarkan sebagai garis dengan arah menunjuk ke objek yang lebih umum (*parent*).
3. *Association*, yaitu relasi hubungan struktural yang menggambarkan sekumpulan *link*, di mana *link* adalah hubungan antar objek. *Association* digambarkan sebagai garis, kadang disertai dengan keterangan dan sering mengandung *multiciply*.
4. *Realization*, yaitu hubungan di mana suatu *classifiers* mengspesifikasikan sebuah kontrak sedangkan *classifiers* lain menyelesaikannya. Kita akan menemukan hubungan *realization* pada 2 buah tempat yaitu diantara *interface* dan *class*/komponen yang tercakup di dalamnya. Juga diantaranya *use case* dan kolaborasi yang mengerjakannya. Secara grafik hubungan *realization* dinyatakan sebagai gabungan antara *generalization* dan *dependency* [19].

2.3.4 Langkah-Langkah Penggunaan UML

Berikut ini adalah langkah-langkah menggunakan UML:

1. Buatlah daftar *business process* dari level tertinggi untuk mendefinisikan aktivitas dan proses yang mungkin muncul.
2. Petakan *use case* untuk tiap *business process* untuk mendefinisikan dengan tepat fungsionalitas yang harus disediakan oleh sistem. Kemudian perhalus *use case* diagram dan lengkapi dengan *requirement*, *constraints* dan catatan-catatan lain.
3. Buatlah *deployment* diagram secara kasar untuk mendefinisikan arsitektur fisik sistem.
4. Definisikan *requirement* lain (non-fungsional, *security* dan sebagainya) yang juga harus disediakan oleh sistem.
5. Berdasarkan *use case* diagram, mulailah membuat *activity* diagram.

6. Definisikan objek-objek level atas (*package* atau *domain*) dan buatlah *sequence* dan *collaboration* diagram untuk tiap alir pekerjaan. Jika sebuah *use case* memiliki kemungkinan alir normal dan *error*, buatlah satu diagram untuk masing-masing alir.
7. Buatlah rancangan *user interface* model yang menyediakan antarmuka bagi pengguna untuk menjalankan skenario *use case*.
8. Berdasarkan model-model yang sudah ada, buatlah *class diagram*. Setiap *package* atau *domain* dipecah menjadi hirarki *class* lengkap dengan atribut dan metodenya. Akan lebih baik jika untuk setiap *class* dibuat *unit test* untuk menguji fungsionalitas *class* dan interaksi dengan *class* lain.
9. Setelah *class* diagram dibuat, kita dapat melihat kemungkinan pengelompokkan *class* menjadi komponen-komponen. Karena itu buatlah *component diagram* pada tahap ini. Juga, definisikan tes integrasi untuk setiap komponen meyakinkan ia berinteraksi dengan baik.
10. Perhalus *deployment* diagram yang sudah dibuat. Detailkan kemampuan dan *requirement software*, sistem operasi, jaringan, dan sebagainya. Petakan komponen ke dalam *node*.
11. Mulailah membangun sistem. Ada dua pendekatan yang dapat digunakan pada tahapan UML ini :
 - a. Pendekatan *use case*, dengan meng-*assign* setiap *use case* kepada tim pengembang tertentu untuk mengembangkan *unit code* yang lengkap dengan tes.
 - b. Pendekatan komponen, yaitu meng-*assign* setiap komponen kepada tim pengembang tertentu.
12. Lakukan uji modul dan uji integrasi serta perbaiki model beserta *codenya*. Model harus selalu sesuai dengan *code* yang aktual.
13. *Software* siap dirilis [19].

2.3.5 Tujuan Pemanfaatan UML

Berikut tujuan utama dalam desain UML adalah sebagai berikut :

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.

2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
7. Memiliki integrasi praktik terbaik [20].

2.3.6 Area penggunaan UML

UML digunakan paling efektif pada domain sebagai berikut :

1. Sistem Informasi Perusahaan
2. Sistem Perbankan dan Perekonomian
3. Bidang Telekomunikasi
4. Bidang Transportasi
5. Bidang Penerbangan
6. Bidang Perdagangan
7. Bidang Pelayanan Elektronik
8. Bidang Pelayanan Berbasis Web Terdistribusi [21].

2.3.7 Bagian-bagian UML

Berikut adalah bagian-bagian dari UML :

1. *View*

Digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi jumlah diagram. Beberapa jenis *View* dalam UML antara lain :

a. *Use case View*

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan *external actors*. *Actor* yang berinteraksi dengan sistem dapat berupa *user*

atau sistem lainnya. *View* ini digambarkan dalam *use case diagram* dan kadang-kadang dengan *activity diagram*.

b. *Logical View*

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class*, *object*, dan *relationship*) dan kolaborasi dinamis yang terjadi ketika *object* mengirim pesan ke *object* lain dalam suatu fungsi tertentu. *View* ini digambarkan dalam *class diagram* untuk struktur statis dan dalam *state*, *sequence*, *collaboration*, dan *activity diagram* untuk model dinamis. *View* ini digunakan untuk merancang (*designer*), dan pengembang (*developer*).

c. *Component View*

Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari *code module* diperhatikan dengan struktur dan ketergantungan juga alokasi pada komponen dan informasi *administrative* lainnya.

d. *Concurrency View*

Membagi sistem ke dalam proses dan prosesor. *View* ini digambarkan dalam diagram (*state*, *sequence*, *collaboration*, dan *activity diagram*) dan diagram implementasi (*component* dan *deployment diagram*) serta untuk pengembang (*developer*), pengintegrasian (*integrator*), dan penguji (*tester*).

e. *Deployment View*

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan lainnya. *View* ini digambarkan dalam *deployment diagram* dan digunakan untuk pengembang (*developer*), pengintegrasian (*integrator*), dan penguji (*tester*) [22].

2.4 Penjualan

Definisi penjualan merupakan sebuah sumber pendapatan pada perusahaan yang berasal dari penjualan. Penjualan sendiri termasuk dalam puncak kegiatan di seluruh kegiatan perusahaan. Definisi lain dari penjualan yaitu merupakan sebuah proses pertukaran antara barang atau jasa yang dihasilkan, juga meliputi antara penjual dan pembeli. Maka dari itu dapat diartikan bahwa penjualan merupakan usaha yang dilakukan oleh manusia guna

menyampaikan barang bagi mereka yang membutuhkan yang ditukar dengan uang berdasarkan harga yang telah ditentukan oleh kedua belah pihak [23].

Sedangkan sistem informasi penjualan merupakan serangkaian prosedur dan metode yang dirancang untuk menghasilkan, menganalisis, menyebarkan dan memperoleh informasi guna mendukung pengambilan keputusan mengenai penjualan. Sistem informasi penjualan mencakup kumpulan prosedur yang melaksanakan, mencatat, mengkalkulasi, membuat dokumen dan informasi penjualan untuk keperluan manajemen dan bagian lain yang berkepentingan, dari mulainya *order* penjualan hingga transaksi dijalankan [24].

Berdasarkan pengertian di atas, dapat disimpulkan bahwa komponen-komponen dari sistem informasi penjualan secara umum terdiri dari :

1. Pencatatan transaksi penjualan
2. Pengecekan stok barang
3. Kalkulasi jumlah dan harga
4. Pembuatan dan pencetakan nota penjualan
5. Pembuatan dokumen atau informasi penjualan untuk keperluan manajemen

Dalam sistem yang dibangun, komponen pencatatan transaksi penjualan, pengecekan stok barang, kalkulasi jumlah dan harga beserta pembuatan dan pencetakan laporan penjualan masuk dalam subsistem *point of sales*. Sedangkan dokumen atau informasi penjualan untuk keperluan manajemen akan menjadi *output* dari sistem informasi yang dibangun [25].

2.5 Pembelian

Pembelian merupakan suatu kegiatan dalam perusahaan sebagai bentuk pengadaan barang yang diperlukan oleh perusahaan. Pembelian memiliki beberapa fungsi diantaranya untuk menangani rutinitas pekerjaan yang berulang-ulang dari bagian pembelian dan penerimaan, mendukung pengambilan keputusan dari orang-orang yang mengatur bagian pembelian, penerimaan, dan membantu dalam penyajian laporan internal dan laporan eksternal [26].

Sistem informasi pembelian memiliki kegunaan untuk mengidentifikasi kebutuhan-kebutuhan informasi mengenai pembelian yang diperuntukkan untuk pengambilan sebuah keputusan. Dengan adanya sistem informasi pembelian tentunya dapat mempermudah dalam

proses transaksi pembelian barang. Di mana sistem akan mengatur stok persediaan barang dan mengatur kapan harus melakukan pemesanan ulang untuk pembelian barang.

Sistem informasi pembelian berperan penting dalam membantu proses pembelian barang. Beberapa hal yang bisa dilakukan jika pembelian memanfaatkan kegunaan dari sistem informasi adalah dapat membantu melakukan pencatatan seperti barang masuk dan barang keluar, penerimaan barang, transaksi pembayaran, dan melakukan penyimpanan data barang ke dalam basis data. Semua transaksi pembelian akan teroganisir dengan adanya sistem.

2.6 Persediaan

Persediaan adalah suatu aktiva yang meliputi barang-barang milik perusahaan dengan maksud untuk dijual dalam suatu periode yang normal, atau sediaan barang-barang yang masih dalam pengerjaan atau proses produksi, ataupun sediaan bahan baku yang menunggu penggunaannya dalam suatu proses produksi [27].

Pemanfaatan sistem informasi dalam mengatur persediaan barang berguna untuk melakukan pencatatan terhadap transaksi dan melakukan pengolahan terhadap barang seperti pengadaan stok barang, penerimaan barang dan barang keluar yang kemudian akan diserahkan ke bagian gudang sehingga menjadi persediaan barang.

Sistem informasi persediaan barang dapat mendukung kelancaran kegiatan kerja pada perusahaan, di mana sistem informasi persediaan ini memiliki peran untuk mengontrol persediaan barang seperti melakukan pencatatan terhadap data transaksi penerimaan barang dan transaksi penggunaan barang. Sistem informasi persediaan juga dapat digunakan untuk pembuatan keputusan dalam mengatur dan mengontrol persediaan gudang untuk meningkatkan produktivitas perusahaan.