

## **BAB II**

### **LANDASAN TEORI**

Pada bab ini berisi penjelasan tentang teori yang menjadi landasan dalam penyusunan tugas akhir ini. Dan secara garis besar akan dijelaskan mengenai pengertian dan konsep-konsep dasar yang akan digunakan dalam perancangan sistem yang akan dibuat dalam tugas akhir ini

#### **1. Konsep Sistem Informasi**

##### **a. Konsep dasar sistem dan Informasi**

Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. (Abdul Kadir, 2003)

Demikian pula seperti yang didefinisikan James A.Hall (2007:6) mengenai pengertian sistem, yaitu sebagai berikut :*“Sistem adalah kelompok dari dua atau lebih komponen atau subsistem yang saling berhubungan yang berfungsi dengan tujuan yang sama”*.

Informasi adalah data yang telah diolah menjadi bentuk yang berarti bagi penerimanya dan bermanfaat dalam mengambil keputusan saat ini atau mendatang. dasar dari informasi adalah data, kesalahan dalam mengambil atau memasukan data, dan kesalahan dalam mengolah data akan menyebabkan kesalahan dalam memberikan informasi.jadi data yang didapatkan dan diinputkan harus valid(benar) hingga bentuk pengolahannya , agar bisa menghasilkan informasi yang dapat dipercaya.

Informasi diperoleh dari sistem informasi (*Informasi System*) atau disebut juga *processing system* atau *informasi processing system* atau *informasi generating system*.

##### **b. Konsep Dasar Sistem Informasi**

Sistem Informasi adalah kombinasi antara prosedur kerja informasi orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi ( Alter, 1992 )

Sistem informasi adalah kumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna ( Bodnar dan Hopwood, 1993)

Sistem informasi adalah suatu sistem didalam organisasi , yang mempertemukan kebutuhan pengolahan transaksi harian, yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan.

Sedangkan Menurut Laudon dalam bukunya' *Management Information system new approaches to organization & Technology*' Mengatakan bahwa Sistem Informasi merupakan komponen – komponen yang saling berhubungan dan bekerjasama untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi tersebut untuk mendukung proses pengambilan keputusan, koordinasi dan pengendalian. *Hall mengatakan bahwa* : Sistem Informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi untuk tujuan yang spesifik( Hall, 2001 )

## **2. Analisa dan Perancangan Sistem Berorientasi Obyek dengan UML**

### **a. UML (*Unified Modeling Language*)**

Menurut *Sun Microsystems Inc*, “ *The Unified Modeling ( UML ) is a graphical language for visualizing, specifying construction, and documenting the artifacts of a software instensive system* “ UML adalah bahasa nyata ( grafis) untuk menggambarkan, menetapkan, membangun, dan mendokumentasikan sesuatu ( Benda ) pada sebuah sistem perangkat lunak secara insentif

UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan model diharapkan pengembangan software dapat memenuhi semua kebutuhan pengguna dengan lengkap dan tepat, termasuk factor – factor seperti lingkup ( *scalability* ), kemampuan ( *robustness* ), keamanan ( *security* ), dan sebagainya

UML didefinisikan notasi dan sintaksis/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram software.

Pengembangan UML dimulai pada Oktober 1994 oleh Booch dan Rumbaugh dan setelah itu Jacobson bergabung pada musim gugur 1995. Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh Object Management Group (OMG). Tahun 1997 UML Versi 1.1 muncul dan kini dengan 1.5 yang dirilis bulan Maret 2003, UML telah menjadi standar bahasa pemodelan untuk aplikasi berorientasi obyek.

#### **b. Analisa Sistem berorientasi Obyek**

Analisis berorientasi obyek adalah tahap menentukan kebutuhan perangkat lunak, yang mendaftar apa pun yang harus dipenuhi oleh sistem software, bukan mengenai bagaimana sistem software melakukannya hasil dari tahap analisis ini adalah dokumen *software requirement specification* (SRS)

Yang dilakukan dalam kegiatan analisis adalah mempelajari domain permasalahan kemudian menghasilkan spesifikasi dari tingkat laku eksternal yang diamati akan mempengaruhi dan dukungan domain permasalahan. Pernyataan yang lengkap, konsisten, dan layak adalah yang harus dihasilkan dari sebuah analisis untuk menjabarkan karakteristik operasional baik fungsi maupun logikanya.

Seorang analisis sistem harus dapat menyatakan apa yang harus dihasilkan dari permasalahan tertentu pada domain, dan harus menentukan batasan permasalahannya.

##### **1) Activity Diagram**

Activity Diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing – masing alir berawal, keputusan yang mungkin terjadi dan bagaimana mereka berakhir. Activity Diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Activity Diagram merupakan diagram khusus dimana sebagian besar keadaan adalah *action* dan

sebagian besar transisi dipacu oleh selesainya keadaan sebelumnya (*internal processing*)

Di dalam diagram aktivitas terdapat :

- a) *Start poin*, menggambarkan awal dari aktivitas



**Gambar II.1**  
**Simbol Start Point**

- b) *End poin*, menggambarkan akhir dari aktivitas



**Gambar II.2**  
**Simbol End Point**

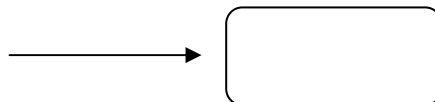
- c) *Activities*, menggambarkan proses bisnis



**Gambar II.3**  
**Simbol Activity**

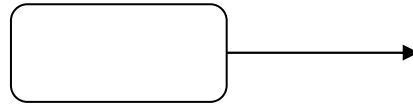
Jenis-jenis activities :

- (1) *Black Hole Activities*, ada masukan dan tidak ada keluaran.



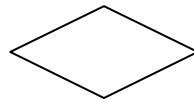
**Gambar II.4**  
**Simbol Black Hole Activities**

- (2) *Miracle Activities*, tidak ada masukan dan ada keluaran dan dipakai pada waktu start point.



**Gambar II.5**  
**Simbol Miracle Activities**

- (3) *Decision point*, tidak ada keterangan (pertanyaan) pada tengah belah ketupat seperti pada *flowchart* dan harus mempunyai *guards* (kunci).



**Gambar II.6**  
**Simbol Decision point**

- (4) *Guards* (kunci), sebuah kondisi benar sewaktu melewati sebuah transisi, harus konsisten dan tidak *overlap*.

[ .... ]

**Gambar II.7**  
**Simbol Guards**

Contoh :  $x < 0$ ,  $x = > 0$  konsisten

$x < = 0$  dan  $x > = 0$  tidak konsisten

- (5) *Swimlane*, Sebuah cara untuk mengelompokkan activity berdasarkan actor (mengelompokkan activity dalam sebuah urutan yang sama).

A rectangular box with a light gray background and a thin black border. Inside the box, the text 'NewSwimlane NewSwimla...' is written in a dark gray, sans-serif font. The text is centered horizontally and appears to be a truncated label for a swimlane diagram.

**Gambar II.8**  
**Simbol Swimlane**

## **2) Analisa Dokumen Masukan**

Analisa keluaran adalah bagian dari pengumpulan informasi tentang sistem yang sedang berjalan. Tujuan analisa masukan adalah memahami prosedur sistem yang sedang berjalan.

## **3) Analisa Dokumen Keluaran**

Analisa masukan adalah analisa mengenai dokumen – dokumen keluaran yang dihasilkan dari sebuah sistem.

## **4) Use case Diagram**

Use case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem dan bukan “bagaimana”. sebuah use case mempersentasikan sebuah integrasi antara actor dengan sistem. Usecase merupakan sebuah pekerjaan tertentu misalnya login kesistem, menciptakan sebuah daftar belanja dan sebagainya.

Secara umum *use case diagram* terdiri dari :

### **a) Actor**

*Actor* adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem. Untuk mengidentifikasi *actor* harus ditentukan pembagian kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. *Actor* dilukiskan dengan peran yang mereka mainkan dalam *use case*, seperti staf penjualan, pelanggan, dll.



**Gambar II.9**  
**Simbol Actor**

**b) Use case**

*Use case* menggambarkan perilaku, termasuk didalamnya interaksi antara *actor* dengan sistem. *Use case* dibuat berdasarkan keperluan *actor*, merupakan “apa” yang dikerjakan sistem bukan “bagaimana” sistem mengerjakannya. Setiap *use case* harus diberi nama yang menyatakan apa hal yang dicapai dari hasil interaksinya.dengan *actor*. Nama *use case* boleh terdiri dari beberapa kata dan tidak boleh ada dua *use case* yang memiliki nama yang sama.



**Gambar II.10**  
**Simbol Use Case**

**5) Deskripsi Usecase Diagram**

Deskripsi Use case adalah Suatu perilaku sistem bagaimana sistem beraksi dan bereaksi dan perilaku ini merupakan aktifitas sistem yang bisa dilihat dari luar dan bisa diuji.

**a) Perancangan Sistem Berorientasi Obyek**

Tahap perancangan atau pendesaignan merupakan tahap untuk menghubungkan antara spesifikasi kebutuhan dan implementasi. Rancangan merupakan rekayasa representasi yang berarti tertahap

sesuatu yang hendak dibangun. Hasil perancangan harus dapat ditelusuri sampai ke spesifikasi kebutuhan dan dapat diukur kualitasnya berdasar kriteria – kriteria rancangan yang baik. Rancangan menekankan pada solusi logis mengenai cara sistem memenuhi kebutuhan.

Perancangan sistem bertujuan untuk menghasilkan model atau representasi entitas yang akan dibangun. Proses pembangunan model mengombinasikan antara intuisi dan pertimbangan yang berdasarkan pada :

- 1) Pengalaman dalam membangun entitas – entitas serupa
- 2) Sekumpulan prinsip dan / atau heuristik yang menuntun jalan dalam evolusi model
- 3) Sekumpul kriteria yang mempertimbang kualitas
- 4) Proses iterasi yang menuntun ke representasi rancangan sistem

#### **1) Entity Relationship Diagram ( ERD )**

Merupakan notasi grafis dalam pemodelan dalam konseptual yang mendeskripsikan hubungan antara penyimpanan. ERD digunakan untuk memodelkan struktur data dan hubungan antardata, ERD pertama kali dideskripsikan oleh *Peter Chen* dalam bukunya “ *The Entity Relationship Model – Toward a Unified of Data*” . Dalam bukunya Peter Chen mencoba merumuskan dasar – dasar model dan setelah itu dikembangkannya dan dimodifikasi. Setelah itu ERD dibuat sebagai bagian dari software CASE yang juga merupakan modifikasi khusus.

#### **2) Local Record Structure( LRS )**

LRS adalah digambarkan oleh kotak persegi panjang dan dengan nama yang unik. *File record* pada LRS ditempatkan dalam kotak. LRS terdiri dari *link-link* diantar tipe *record* lainnya, banyaknya *link* dari LRS yang diberi nama oleh field-field yang kelihatan pada kedua *link* tipe *record*.



### 3) Tabel / Relasi

Tabel adalah koleksi objek yang terdiri dari sekumpulan elemen yang diorganisasi secara *kontinue*, artinya memori yang dialokasi antara satu elemen dengan elemen yang lainnya mempunyai *address* yang berurutan.

Pada tabel, pengertian perlu dipahami adalah:

- a) Keseluruhan tabel (sebagai koleksi) adalah kontainer yang menampung seluruh elemen
- b) Indek tabel, yang menunjukan address dari sebuah elemen
- c) *Element* tabel, yang dapat dipacu melalui indeknya, bertipe tertentu yang sudah terdefinisi
- d) Seluruh *element* tabel bertipe "sama". Dengan catatan : beberapa bahasa pemograman memungkinkan pendefinisian tabel dengan elemen generik, tapi pada saat diinstansiasi, harus diinstansiasi dengan tipe sama.

### 4) Spesifikasi Basis Data

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lain dan tersimpan diluar komputer serta digunakan perangkat lunak (*software*) tertentu untuk memanipulasinya.

Sedangkan sistem berbasis data adalah suatu sistem penyusunan dan pengelolaan *record-record* dengan menggunakan komputer dengan tujuan untuk menyimpan atau merekam serta melihat data operasional lengkap pada sebuah organisasi, sehingga mampu menyediakan informasi yang diperlukan untuk kepentingan proses pengambilan keputusan.

### 5) Rancangan Dokumen Keluran

Keluaran merupakan hasil dari proses data - data didalam sistem informasi yang diperoleh pada terjadi transaksi - transaksi yang dilakukan oleh organisasi merupakan bahan mentah untuk menghasilkan informasi. Tujuan dari rancangan keluaran yaitu untuk menentukan

kebutuhan Output dari sistem yang baru .Bagaimana output dimedia lunak , media keras dilayar terminal.

#### **6) Rancangan Dokumen Masukan**

Masukaan merupakan awal dimulai proses informasi. Data - data yang diperoleh pada tetrtjadi transaksi – transaksi yang dilakukan oleh organisasi merupakan bahan mentah untuk menghasilkan informasi. ( Jogiyanto,1990 )

#### **7) Rancangan Layar Program**

Rancangan tampilan merupakan bentuk tampilan sistem layar komputer sebagai antar muka dengan pemakai yang akan dihasilkan dari sistem yang dirancang

#### **8) Sequence Diagram**

*Sequence diagram* adalah menjelaskan interaksi obyek yang disusun dalam suatu urutan waktu/kejadian. Urutan waktu yang dimaksud adalah urutan yang dilakukan seorang aktor dalam menjalankan sistem. Diagram ini secara khusus berasosiasi dengan *use case diagram*, memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam *use case*. Diagram ini sebaiknya digunakan diawal tahap perancangan atau analisis karena kesederhanaannya dan mudah dimengerti.

*Sequence diagram* menunjukkan bagaimana data operasi dilakukan pesan apa yang dikirim dan kapan *sequence diagram* didasarkan atas kelas diagram yang sudah dibuat hanya saja kelas diagram yang sudah dibuat belum menyertakan kelas *boundary* dan kelas *control*, maka sebelum membuat *sequence diagram* perlu dibuat *class boundary* dan kelas *control* terlebih dahulu.

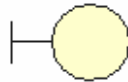
Beberapa simbol yang umum digunakan pada *sequence diagram*, yaitu:

- a) *Entity Obyek*, suatu obyek yang berisi informasi kegiatan yang terkait yang tetap dan disimpan ke dalam suatu database.



**Gambar II.11**  
**Simbol Entity Object**

b) *Interface/Boundary Object*, sebuah obyek yang menjadi penghubung antara user dengan sistem. Contohnya *window*, *dialogue box* atau *screen*(tampilan layar)



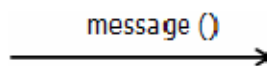
**Gambar II.12**  
**Simbol Boundary Object**

c) *Control Object*, suatu obyek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada *entitas*. Contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai obyek. *Control object* mengkoordinir pesan(*message*) antara *boundary* dengan *entitas*.



**Gambar II.13**  
**Simbol Control Object**

d) *Simple Message*, simbol pengiriman pesan dari sebuah obyek ke obyek lain.



**Gambar II.14**  
**Simbol Message**

- e) *Recursive*, sebuah obyek yang mempunyai sebuah operation kepada dirinya sendiri.



**Gambar II.15**

**Simbol Recursive**

- f) *Lifeline*, garis titik-titik yang terhubung dengan obyek, sepanjang *lifeline* terdapat *activation*.



**Gambar II.16**

**Simbol Lifeline**

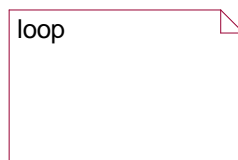
- g) *Activation*, *activation* mewakili sebuah eksekusi operasi dari obyek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.



**Gambar II.17**

**Simbol Activation**

- h) *Loop*, menggambarkan dari suatu kejadian yang dilakukan secara berulang-ulang.



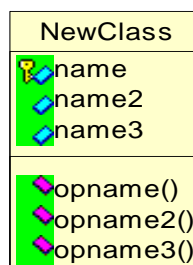
**Gambar II.18**

**Simbol Loop**

## 9) Class Diagram

*Class diagram* sangat membantu dalam visualisasi kelas dari suatu sistem. Hal ini disebabkan karena *class* adalah deskripsi obyek-obyek dengan atribut (*property*), perilaku (*operation*) dan relasi yang sama. Disamping itu *class diagram* bisa memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari *class-class* yang ada dan relasinya satu dengan lainnya. Itulah sebabnya *class diagram* menjadi diagram paling populer di UML.

Diagram kelas memperlihatkan aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Diagram ini berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur yang dibuat. Diagram ini merupakan fondasi untuk *component diagram* dan *deployment diagram*. Dalam notasi UML *class* digambarkan dengan kotak. Nama class menggunakan huruf besar diawal kalimatnya dan diletakkan diatas kotak.



**Gambar II.19**

### **Simbol Class Diagram**

Komponen-komponen *class diagram* :

#### a) Kelas, Atribut, Operasi

Kelas didefinisikan sebagai kumpulan atau himpunan obyek-obyek yang dengan atribut dan *operation* yang sama. Obyek (object) adalah orang, benda, tempat, kejadian atau konsep-konsep yang ada didunia nyata yang penting bagi suatu aplikasi perangkat lunak/perangkat keras. Setiap obyek memiliki keadaan sesaat (*state*),

perilaku (*behaviour*). *State* sebuah obyek adalah kondisi obyek tersebut yang dinyatakan dalam *attribute/properties*. Sedangkan perilaku suatu obyek mendefinisikan bagaimana sebuah objek bertindak atau berinteraksi. Perilaku sebuah obyek dinyatakan dalam *operation*.

Atribut adalah data yang dimiliki suatu obyek dalam suatu kelas, misalnya kelas barang, yang memiliki atribut *KdBrg*, sebuah kelas (tingkah laku sebuah obyek) atau fungsi yang dapat diaplikasikan ke suatu obyek dalam kelas. Misalnya suatu obyek barang pasti memiliki fungsi-fungsi seperti cari, simpan, hapus dan sebagainya. Operasi yang sama dapat diterapkan pada kelas yang berbeda, misalnya fungsi simpan dapat diterapkan pada kelas barang maupun pada kelas pelanggan.

*b) Association*

*Association* menunjukkan hubungan antara masing-masing kelas. Setiap *association* mempunyai dua *association end*. Masing-masing *end* dihubungkan ke satu kelas dari kelas-kelas dalam *association*. Sebuah *end* dapat dibuat lebih jelas dengan memberikan nama dengan sebuah label. Label ini disebut dengan *role name* (*association end* sering disebut *role*).

Sebuah *association end* juga mempunyai atau memiliki “*multiplicity*”, *multiplicity* ini menunjukkan berapa banyak obyek yang berpartisipasi dalam suatu relasi. Secara umum, *multiplicity* menunjukkan batasan terendah dan tertinggi untuk obyek-obyek yang berpartisipasi. *Multiplicity* yang paling umum digunakan adalah 1, \*, dan 0..1.

---

**Gambar II.20**  
**Simbol Association**

*Multiplicity* menunjukkan beberapa banyak obyek yang berpartisipasi dalam suatu relasi. Secara umum, *multiplicity* menunjukkan batasan terendah dan tertinggi untuk obyek-obyek yang berpartisipasi.

*Multiplicity* yang paling umum digunakan adalah 1, \*, dan 0..1. Langkah-langkah *transformasi* dari *conceptual* data model ke tabel relasi adalah sebagai berikut :

- (1) Jika hubungan yang terjadi antar *class* adalah 1 to 1 (one to one) maka atribut dari *relationship* set diambil dan dimasukkan ke entitas yang lebih membutuhkan.
- (2) Jika hubungan yang terjadi antar *class* adalah 1 to 0..1 (*one to zero one*) maka atribut dari *relationship* set digabung ke entitas yang memiliki *multiplicity* 0..1.
- (3) Jika hubungan yang terjadi antar *class* adalah 1 to \* (one to many). maka atribut dari *relationship* set digabung dengan set entitas yang memiliki *multiplicity* banyak (many).

$$\frac{1}{\quad\quad\quad} \quad\quad\quad 1 \dots *$$

**Gambar II.21**  
**Simbol Multiplicity**

### 3. Teori Pendukung

#### a. Sistem Informasi Penjualan Tunai

##### 1) Pengertian Penjualan

Penjualan adalah suatu usaha yang terpadu untuk mengembangkan rencana- rencana strategis yang diarahkan pada usaha pemuasan kebutuhan dan keinginan pembelian, guna mendapatkan penjualan yang menghasilkan laba( Marwan , 1991). Penjualan merupakan sumber hidup suatu perusahaan karena dari penjualan dapat diperoleh laba serta suatu usaha memikat konsumen yang diusahakan

untuk mengetahui daya tarik mereka sehingga dapat mengetahui hasil produk yang dihasilkan. Menurut Winardi ( 1982 ), Penjualan adalah suatu transfer hak atas benda – benda. Dari penjelasan tersebut dalam memindahkan atau mentransfer barang dan jasa diperlukan orang – orang yang bekerja dibidang penjualan seperti pelaksanaan dagang , agen wakil pelayanan dan wakil pemesaran.

## **2) Faktor – faktor yang mempengaruhi kegiatan Penjualan**

Dalam Praktek, kegiatan Penjualan itu dipengaruhi oleh beberapa faktor sebagai berikut : ( Swastha dan Irawan : 1990 )

### **a) Kondisi dan Kemampuan Penjual**

Transaksi jual-beli atau pemindahan hak milik secara komersial atas barang dan jasa itu pada prinsipnya melibatkan dua pihak, yaitu penjual sebagai pihak pertama dan pembeli sebagai pihak kedua. Disini penjual harus dapat menyakinkan kepada pembelinya agar dapat berhasil mencapai sasaran penjualan yang diharapkan. Untuk maksud tersebut penjual harus memahami beberapa masalah penting yang sangat berkaitan, yakni:

- a. Jenis dan karakteristik barang yang di tawarkan.
- b. Harga produk.
- c. Syarat penjualan seperti: pembayaran, penghantaran, pelayanan sesudah penjualan, garansi dan sebagainya.

### **b) Kondisi Pasar.**

Pasar, sebagai kelompok pembeli atau pihak yang menjadi sasaran dalam penjualan, dapat pula mempengaruhi kegiatan penjualannya. Adapun faktor-faktor kondisi pasar yang perlu di perhatikan adalah:

- a. Jenis pasarnya
- b. Kelompok pembeli atau segmen pasarnya
- c. Daya belinya
- d. Frekuensi pembelian
- e. Keinginan dan kebutuhan



### **c) Modal**

Akan lebih sulit bagi penjualan barangnya apabila barang yang dijual tersebut belum dikenal oleh calon pembeli, atau apabila lokasi pembeli jauh dari tempat penjual. Dalam keadaan seperti ini, penjual harus memperkenalkan dulu membawa barangnya ketempat pembeli. Untuk melaksanakan maksud tersebut diperlukan adanya sarana serta usaha, seperti: alat transport, tempat peragaan baik didalam perusahaan maupun di luar perusahaan, usaha promosi, dan sebagainya. Semua ini hanya dapat dilakukan apabila penjualan memiliki sejumlah modal yang diperlukan untuk itu.

## **2) Sistem penjualan Tunai**

### **a) Pengertian Penjualan Tunai**

Penjualan diartikan sebagai prosedur pengeluaran barang dan penerimaan uang oleh perusahaan. Penjualan adalah aktivitas utama dalam perusahaan dagang sebab penjualan merupakan sasaran terakhir dalam seluruh aktivitas untuk mendapatkan keuntungan. Penjualan juga dapat diartikan sebagai suatu kegiatan untuk menawarkan barang dan jasa kepada pelanggan.

Kegiatan penjualan terdiri dari transaksi penjualan barang dan jasa secara kredit dan tunai. Dalam transaksi penjualan kredit, jika order dari pelanggan telah terpenuhi perusahaan memiliki piutang kepada langganannya. Dalam transaksi penjualan tunai barang atau jasa baru diserahkan kepada pembelian jika telah diterima uang dari pembeli.

Penjualan tunai dapat diartikan suatu proses transaksi yang dilakukan oleh dua pihak tanpa ada perjanjian khusus yang mengikat didalam terjadinya proses pembelian suatu barang dan jasa.

Penjualan tunai dilakukan oleh perusahaan dengan cara mewajibkan pembeli melakukan pembayaran harga barang terlebih dahulu sebelum barang diserahkan kepada pembeli. Setelah diterima barang kemudian diserahkan kepada pembeli dan transaksi penjualan tunai kemudian dicatat oleh perusahaan