# MODUL 1

**MEMBUAT DFD (DATA FLOW DIAGRAM) DAN UML (UNIFIED MODELING LANGUAGE)**

# DENGAN MENGGUNAKAN MICROSOFT VISIO PROFESSIONAL

1. **Tujuan**
   1. Mengetahui area kerja Microsoft Visio Professional.
   2. Mampu membuat DFD dan berbagai macam diagram UML yang disediakan oleh Visio Professional.

# Teori Dasar

* 1. **Tentang Ms. Visio Professional**

Microsoft Visio (atau sering disebut Visio) adalah sebuah [program](http://id.wikipedia.org/wiki/Program_komputer) aplikasi [komputer](http://id.wikipedia.org/wiki/Komputer) yang sering digunakan untuk membuat [diagram,](http://id.wikipedia.org/wiki/Diagram) [diagram alir](http://id.wikipedia.org/wiki/Diagram_alir) (flowchart), [brainstorm](http://id.wikipedia.org/w/index.php?title=Brainstorm&action=edit&redlink=1), dan skema [jaringan](http://id.wikipedia.org/wiki/Jaringan_komputer) yang dirilis oleh [Microsoft Corporation.](http://id.wikipedia.org/wiki/Microsoft_Corporation) Aplikasi ini menggunakan [grafik vektor](http://id.wikipedia.org/wiki/Grafik_vektor) untuk membuat diagram-diagramnya.

Bagian-bagian dari tampilan program Visio dapat dibagi menjadi 5 bagian, yaitu :

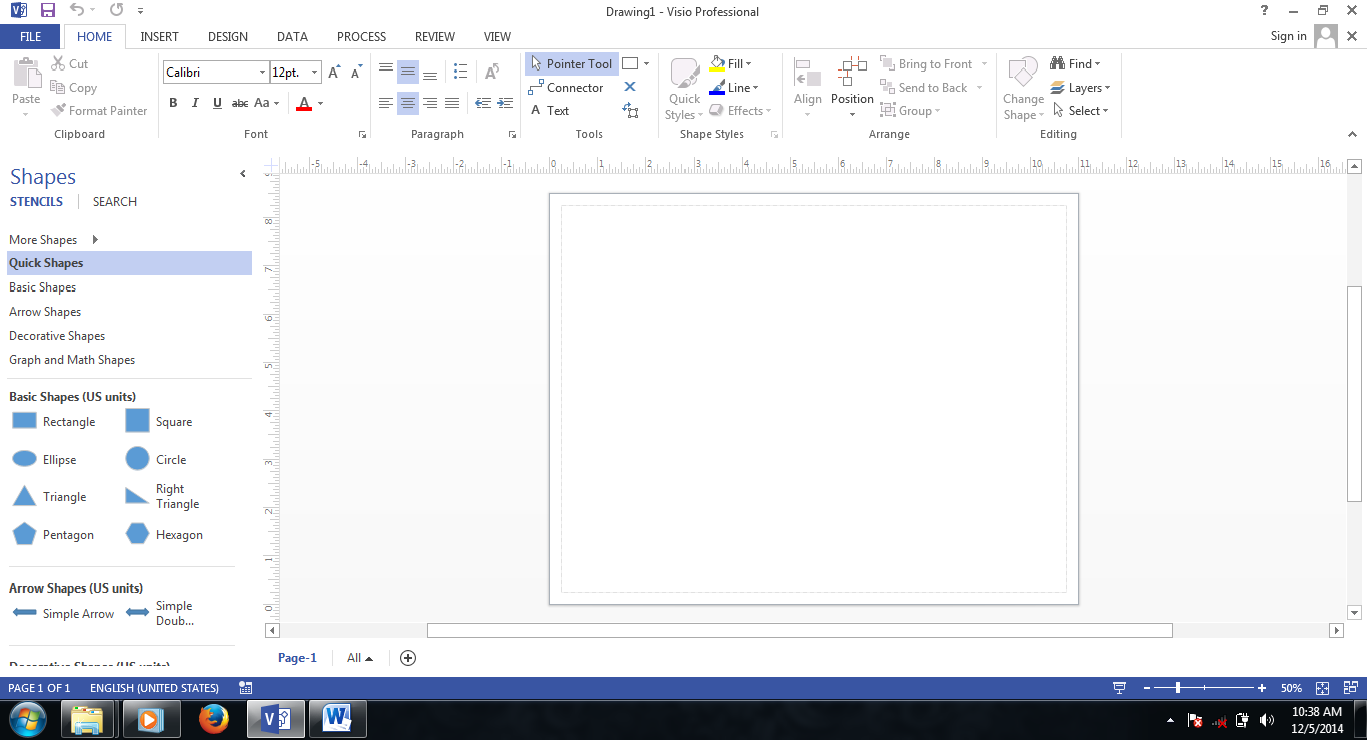
1. Bagian menu utama, berisikan menu-menu yang secara umum sama fungsinya yaitu memiliki sub-sub menu yang digunakan dalam program.
2. Bagian toolbar, berisi tool-tool yang digunakan untuk berbagai keperluan dalam menggambar

seperti membuat tulisan, membuat garis, memutar gambar, dan lainnya.

1. Bagian stencil, bagian ini memiliki berbagai gambar template yang dapat langsung kita gunakan

atau kita edit kembali ukurannya maupun keterangan gambar tersebut.

1. Bagian halaman desain, bagian ini digunakan sebagai bidang kerja kita dalam membuat gambar.
2. Jumlah halaman, pada bagian ini kita dapat menambahkan halaman baru jika halaman yang digunakan tidak dapat menampung jumlah gambar yang kita buat.



Area kerja Ms. Visio Professional 2013

# Data Flow Diagram (DFD)

DFD menggambarkan arus data dari suatu sistem informasi, baik sistem lama maupun sistem baru secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut berada. Simbol yang digunakan dalam pembuatan DFD terbagi empat bagian, yaitu Entity (kesatuan luar), Data flow (arus data), Process (proses), dan Data store (simpanan data) yang terdapat dalam 2 versi yang dapat dilihat pada tabel di bawah ini :

|  |  |  |
| --- | --- | --- |
| **Simbol DeMarco dan Yourdan** | **Keterangan** | **Simbol Gane dan Sarson** |
|  | Source (Kesatuan Luar) |  |
|  | Proses |  |
|  | Data Flow (Arus Data) |  |
|  | Data Store (Simpanan Data) |  |

# Activity Diagram

Acitivity Diagram (AD) diperlukan untuk menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses.

Berikut adalah simbol-simbol yang terdapat dalam Activity Diagram :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Simbol** | | | | **Keterangan** |
|  | | | | Start Point |
|  | | | | End Point |
|  | | | | Activities |
|  | | | | Fork (Percabangan) |
|  | | | | Join (Penggabungan) |
|  | |  | |
|  |  |
|  | |  |  |
|  |  |  | |
|  | | | |
|  | | | | Decision |
| Swimlane | | | | Sebuah cara untuk mengelompokkan activity berdasarkan Actor (mengelompokkan activity dalam sebuah urutan yang sama) |

# Class Diagram

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

Berikut adalah simbol-simbol yang terdapat dalam Class Diagram :

|  |  |
| --- | --- |
| **Simbol** | **Deskripsi** |
| *Package*  package | *Package* merupakan sebuah bungkusan dari satu atau lebih kelas |
| Kelas  **nama\_kelas**  +atribut  +operasi() | Kelas pada struktur sistem |
| Antarmuka / *interface*    **nama\_interface** | Sama dengan konsep *interface* dalam pemrograman berorientasi objek |
| Asosiasi / *association* | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity* |
| Asosiasi berarah / *directed association* | Relasi antar kelas dengan makna  kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity* |
| Generalisasi | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus) |
| Kebergantungan / *dependency* | Relasi antar kelas dengan makna kebergantungan antar kelas |
| Agregasi / *aggregation* | Relasi antar kelas dengan makna semua-bagian (*whole-part*) |

# Sequence Diagram



Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display/form) berupa message yang digambarkan terhadap waktu.

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah- langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

Berikut adalah simbol-simbol yang terdapat dalam Sequence Diagram :

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
|  | *Object* merupakan *instance* dari sebuah *class* dan dituliskan tersusun secara *horizontal*. Digambarkan sebagai sebuah *class* (kotak) dengan nama *object*  di dalamnya yang diawali dengan sebuah titik koma. |
|  | *Actor* juga dapat berkomunikasi dengan *object* , maka *actor* juga dapat diurutkan sebagai kolom. Simbol *Actor* sama dengan simbol pada *Actor* Use Case Diagram. |
|  | *Lifeline* mengindikasikan keberadaan sebuah *object* dalam basis waktu.  Notasi  untuk *Lifeline* adalah garis putus-putus *vertikal* yang ditarik dari sebuah  *object* . |
|  | *Activation* dinotasikan sebagai sebuah kotak segi empat yang digambar pada  sebuah *lifeline*. mengindikasikan sebuah obyek yang akan melakukan sebuah aksi. |
|  | *Message*, digambarkan dengan anak panah horizontal antara *Activation Message* mengindikasikan komunikasi antara *object* -*object* . |

# Use Case Diagram

Diagram Use Case menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. Yang menjadi persoalan itu apa yang dilakukan bukan bagaimana melakukannya. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Berikut adalah simbol-simbol yang terdapat dalam Sequence Diagram :

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
|  | *Use case* digambarkan sebagai lingkaran elips dengan nama *use*  *case* dituliskan didalam elips tersebut. |
|  | *Actor* adalah pengguna sistem. *Actor* tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan *input* atau memberikan *output*, maka aplikasi  tersebut juga bisa dianggap sebagai *actor*. |
|  | Asosiasi digunakan untuk menghubungkan *actor* dengan *use case*.  Asosiasi digambarkan dengan sebuah garis yang menghubungkan  antara *Actor*dengan *Use Case*. |

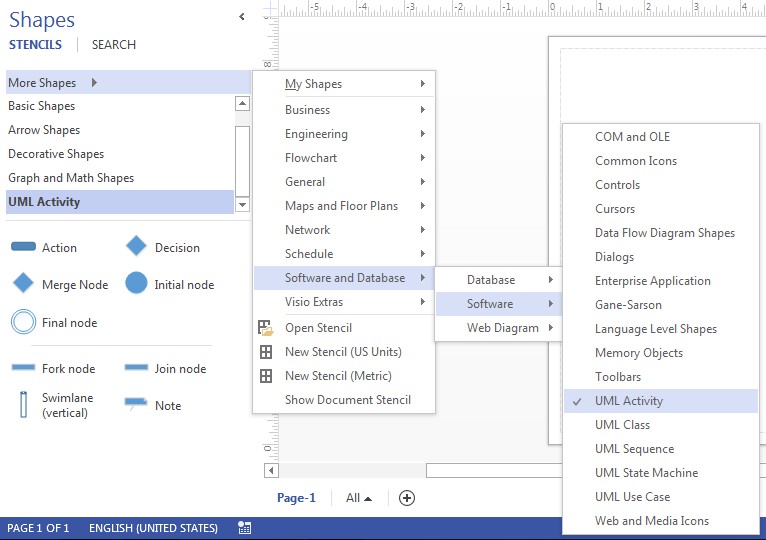
# Percobaan

Buatlah Activity Diagram, Class Diagram, Contex Diagram, Data Flow Diagram, Sequence Diagram dan Use Case Diagram dari sistem penjualan pada suatu tempat/toko.

# Langkah Kerja

1. **Activity Diagram**
   1. Bukalah aplikasi Ms. Visio->Pilih Blank Drawing->Pilih US Units->Create.
   2. Pada bagian Shapes, pilih More Shapes->Software and Database->Software-

>UML Activity.



* 1. Pilihlah simbol-simbol yang terdapat pada UML Activity, lalu gambarkan activity diagram dari sistem penjualan sesuai dengan gambar di bawah ini :



Tampilan Data Barang

Menambah Barang

Pilih Barang

Masukan Data

Barang

Hapus data Barang

Menyimpan Data

Barang

System

User

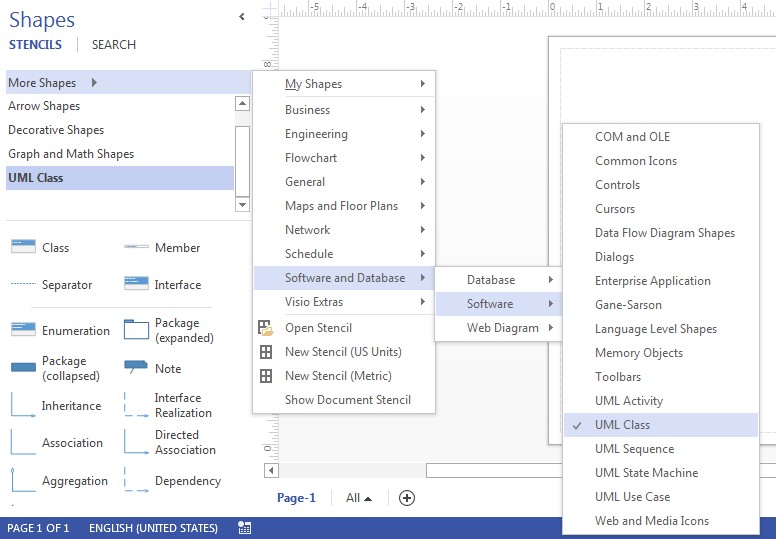
Sistem Penjualan

* 1. Analisalah gambar diagram tersebut.

# Class Diagram

* 1. Tambahkan halaman baru pada file yang sudah terbuka dengan cara pilih menu Insert->New Page.
  2. Pada bagian Shapes, pilih More Shapes->Software and Database->Software-

>UML Class.



* 1. Pilihlah simbol-simbol yang terdapat pada UML Class, lalu gambarkan class diagram dari sistem penjualan sesuai dengan gambar di bawah ini :

+SetNo\_Beli () : void

+SetTgl\_Beli () : void

+SetNama\_Pemasok () : void

+SetKode\_Barang () : void

+SetNama\_Barang () : void

+SetHarga\_Beli () : void

+SetJumlah\_Beli () : void

+SetSub\_Total () : void

+GetNo\_Beli () : string

+GetTgl\_Beli () : string

+GetNama\_Pemasok () : string

+GetKode\_Barang () : string

+GetNama\_Barang () : string

+GetHarga\_Beli () : string

+GetJumlah\_Beli () : string

+GetSub\_Total () : string

+SetKode\_Barang () : void

+SetNama\_Barang () : void

+SetHarga () : void

+SetStok () : void

+GetKode\_Barang () : string

+GetNama\_Barang () : string

+GetHarga () : string

+GetStok () : string

+SetKode\_Pemasok () : void

+SetNama\_Pemasok () : void

+SetAlamat () : void

+GetKode\_Pemasok () : string

+GetNama\_Pemasok () : string

+GetAlamat () : string

-Kode\_Barang : varchar

-Nama\_Barang :varchar

-Harga : int

-Stok : int

-No\_Beli : varchar

-Tgl\_Beli : date

-Nama\_Pemasok : varchar

-Kode\_Barang : varchar

-Nama\_Barang : varchar

-Harga\_Beli : int

-Jumlah\_Beli : int

-Sub\_Total : int

-Kode\_Pemasok : varchar

-Nama\_Pemasok : varchar

-Alamat : varchar

Barang

Pembelian

Pemasok

* 1. Analisalah diagram tersebut.

+SetKode\_Pelanggan () : void

+SetNama\_Pelanggan () : void

+SetAlamat\_Pelanggan () : void

+GetKode\_Pelanggan () : string

+GetNama\_Pelanggan () : string

+GetAlamat\_Pelanggan () : string

-Kode\_Pelanggan : varchar

-Nama\_Pelanggan : varchar

-Alamat\_Pelanggan : varchar

Pelanggan

-SetNo\_Nota () : void

-SetTgl\_Nota () : void

-SetKode\_Barang () : void

-SetNama\_Barang () : void

-SetHarga\_Barang () : void

-SetJumlah\_Jual () : void

-SetTotal () : void

-SetUser () : void

-GetNo\_Nota () : string

-GetTgl\_Nota () : string

-GetKode\_Barang () : string

-GetNama\_Barang () : string

-GetHarga\_Barang () : string

-GetJumlah\_Jual () : string

-GetTotal () : string

-GetUser () : string

-No\_Nota : varchar

-Tgl\_Nota : date

-Kode\_Barang : varchar

-Nama\_Barang : varchar

-Harga\_Barang : int

-Jumlah\_Jual : int

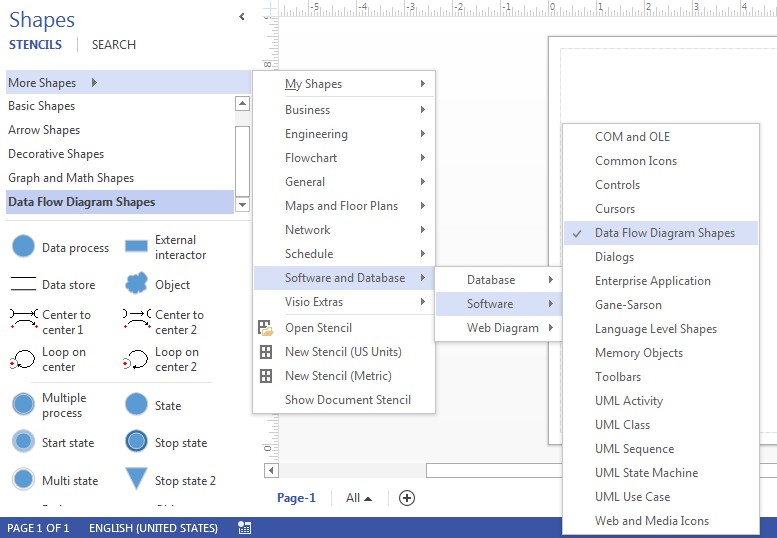
-Total : int

-User : varchar

Penjualan

# Contex Diagram

* 1. Tambahkan halaman baru pada file yang sudah terbuka dengan cara pilih menu Insert->New Page.
  2. Pada bagian Shapes, pilih More Shapes->Software and Database->Software->Data Flow Diagram Shapes.



* 1. Pilihlah simbol-simbol yang terdapat pada Data Flow Diagram Shapes, lalu gambarkan Contex Diagram dari sistem penjualan sesuai dengan gambar di bawah ini :

Manager

Laporan

Data Barang Order

Informasi Stok Barang

SISTEM PENJUALAN

0

Informasi Stok Barang Data Barang Yang Dibeli

Kasir/Operator

Data Barang Data Pemasok

Petugas Gudang

Data Order Barang

Return Pembelian

Struk Belanja

Data Barang dibeli

Return Pembelian Data Barang Order

Data Pemasok

Pemasok

Pelanggan

* 1. Analisalah gambar diagram tersebut.

# Data Flow Diagram (DFD) Level 0

* 1. Tambahkan halaman baru pada file yang sudah terbuka dengan cara pilih menu Insert->New Page.
  2. Pilihlah simbol-simbol yang terdapat pada Data Flow Diagram Shapes, lalu gambarkan DFD dari sistem penjualan sesuai dengan gambar di bawah ini :

1.1

Data Barang

Pendataan

Barang

Info Stok Barang

Pesanan Barang

1.2

Data Pemasok

Data Barang Masuk

Pendataan

Pemasok

Pemasok

Data Pesanan

Kasir/Operator

Data Pesanan

Data Pembelian

1.3

Pembelian

Petugas

Gudang

Data Pembelian

Pelanggan

1.4

Penjualan

Nota

Barang Dibeli

Penjualan

Data Penjualan

1.5

Data Pembelian

Laporan

P

Data Penjualan

embelian

Laporan

Data Barang Barang

Data Pemasok

Pemasok

Info Stok Barang

Data Barang

Data Pemasok

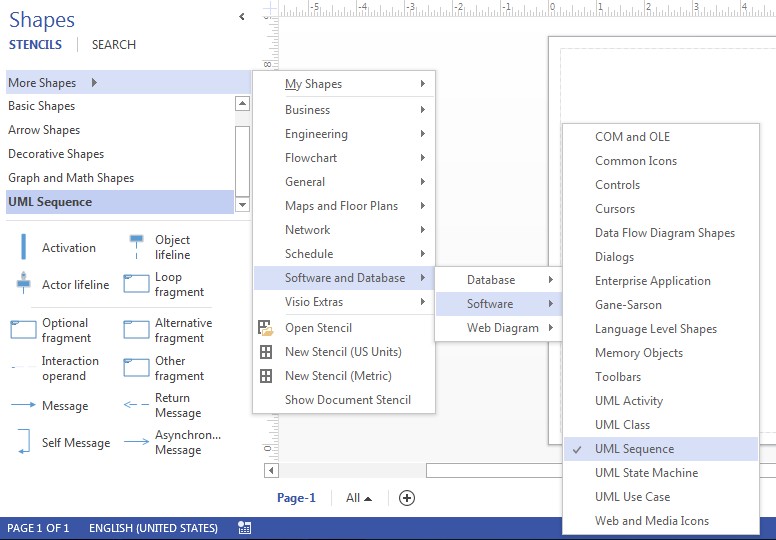
Manajer

* 1. Analisalah gambar diagram tersebut.

# Sequence Diagram

* 1. Tambahkan halaman baru pada file yang sudah terbuka dengan cara pilih menu Insert->New Page.
  2. Pada bagian Shapes, pilih More Shapes->Software and Database->Software-

>UML Sequence.



* 1. Pilihlah simbol-simbol yang terdapat pada UML Sequence, lalu gambarkan Sequence Diagram dari sistem penjualan sesuai dengan gambar di bawah ini :



**Bagian**

**Gudang**

**Kasir**

**Pelanggan Manajer**

**: Barang**

**: Penjualan**

**: Laporan**

Mengecek Barang

Info Stok Barang Data Barang

Info Stok Barang

Data Barang Di Beli

Data Penjualan

Nota Belanja

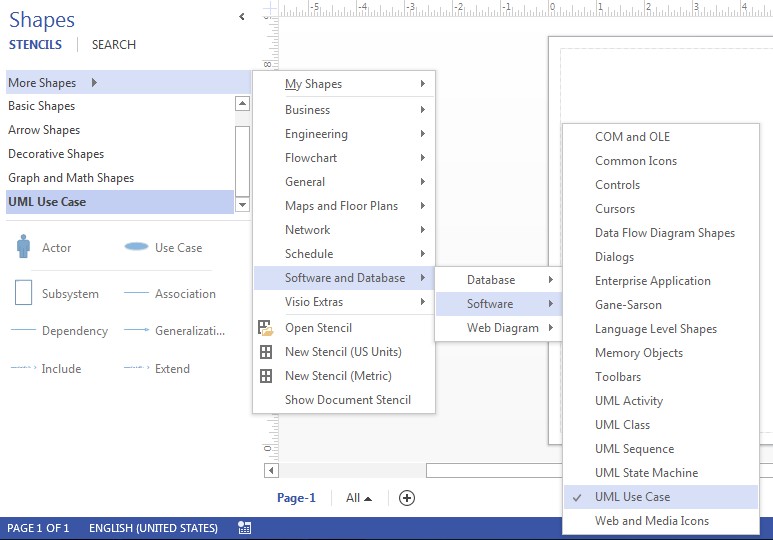
Laporan Barang dan Penjualan

* 1. Analisalah gambar diagram tersebut.

# Use Case Diagram

* 1. Tambahkan halaman baru pada file yang sudah terbuka dengan cara pilih menu Insert->New Page.
  2. Pada bagian Shapes, pilih More Shapes->Software and Database->Software-

>UML Use Case.



* 1. Pilihlah simbol-simbol yang terdapat pada UML Use Case, lalu gambarkan Use Case Diagram dari sistem penjualan sesuai dengan gambar di bawah ini :



Memenuhi persediaan barang

Bagian Gudang

Mengecek stok barang

Transaksi Penjualan

<<include>>

Kasir

Pelanggan

Pembayaran

<<extend>>

Pengambilan Nota

Manajer

Laporan Data Penjualan

* 1. Analisalah gambar diagram tersebut.

# Latihan

Buatlah Activity Diagram, Class Diagram, Contex Diagram, Data Flow Diagram, Sequence Diagram dan Use Case Diagram dari sistem pada suatu perpustakaan online.