**PERCOBAAN IV**

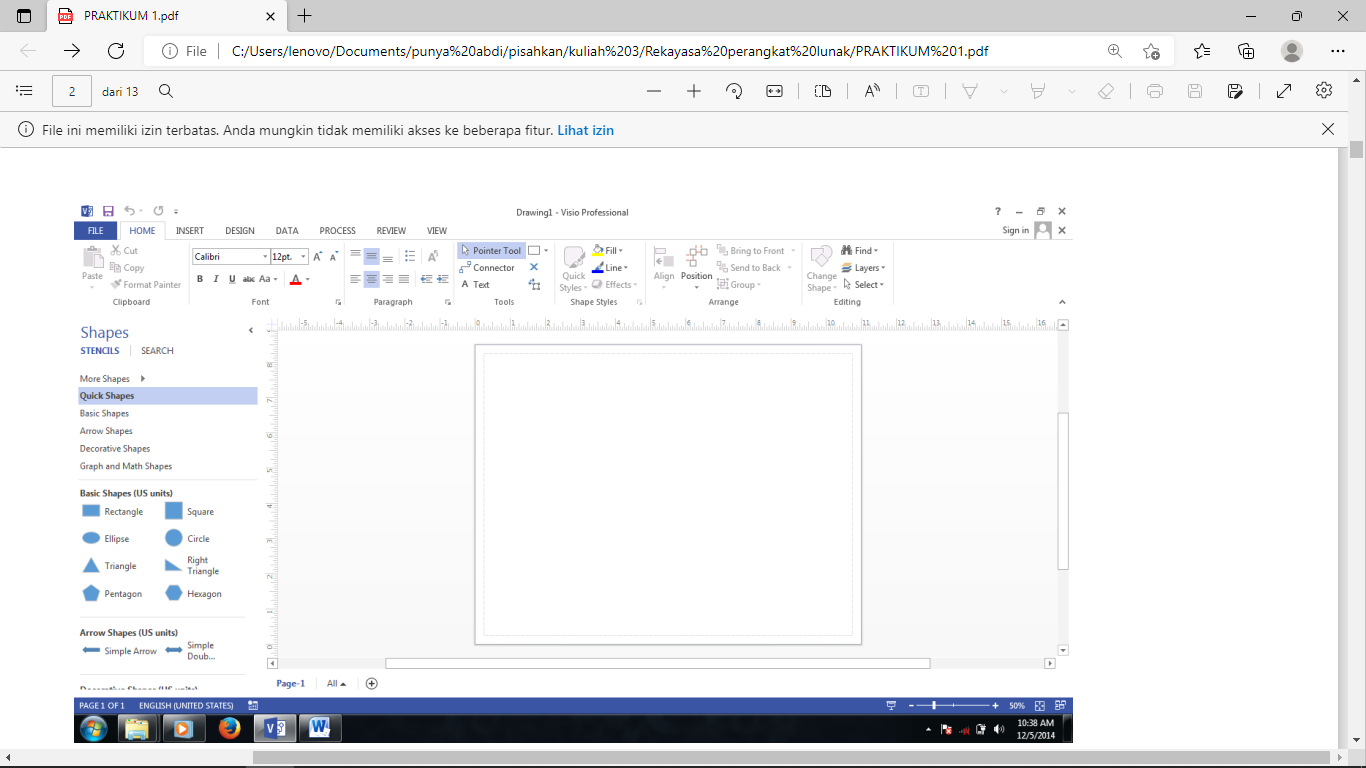
*ANDROID WEB API*

1. **TUJUAN**
2. Mengetahui area kerja *Microsoft Visio Professional*.
3. Mampu membuat *DFD* dan berbagai macam diagram *UML* yang disediakan oleh *Visio Professional*.
4. **ALAT DAN BAHAN**
5. Laptop/PC
6. *Microsoft Visio Professional* atau sejenisnya
7. Modul pratikum
8. **TEORI DASAR**
9. **Tentang *Ms. Visio Professional***

*Microsoft Visio* (atau sering disebut *Visio*) adalah sebuah program aplikasi komputer yang sering digunakan untuk membuat diagram, diagram alir (*flowchart*), *brainstorm*, dan skema jaringan yang dirilis oleh *Microsoft Corporation*. Aplikasi ini menggunakan grafik vektor untuk membuat diagram-diagramnya.

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

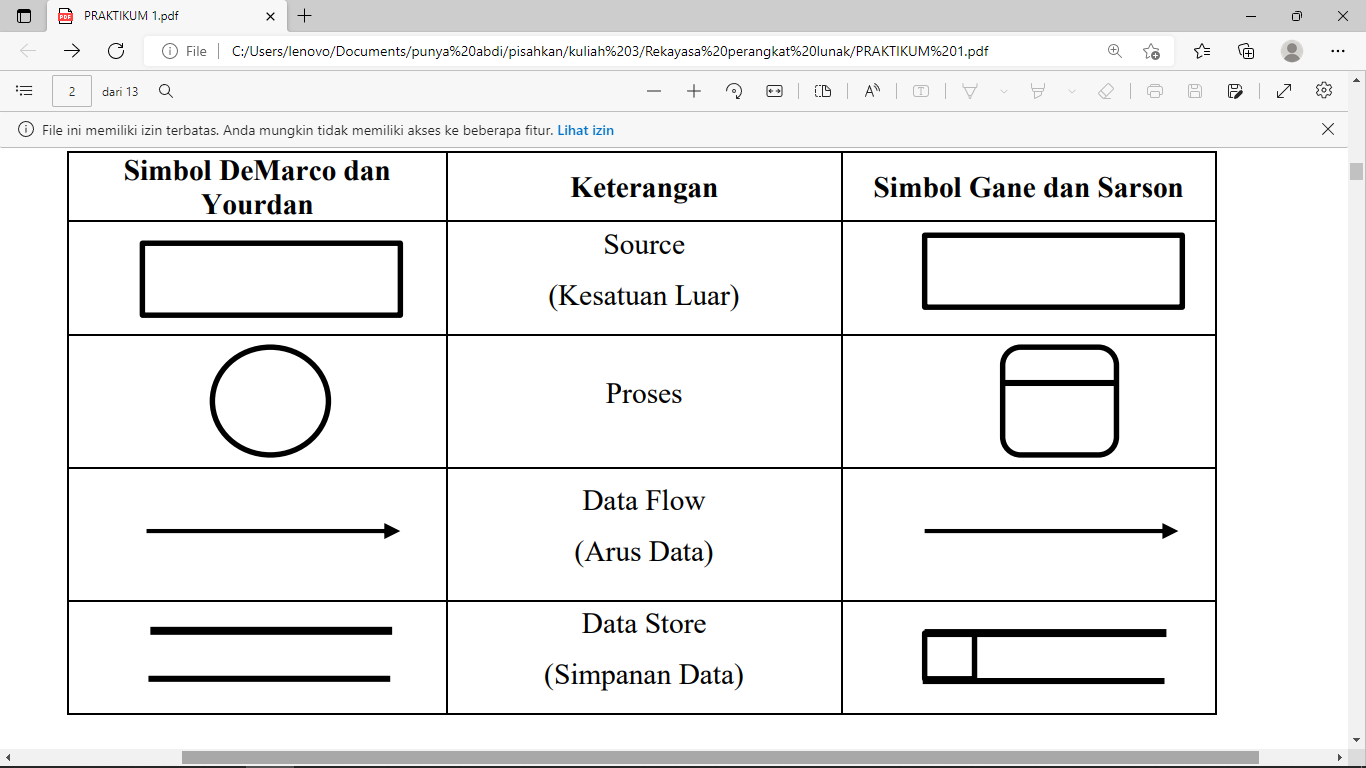
1. Bagian menu utama, berisikan menu-menu yang secara umum sama fungsinya yaitu memiliki *sub-sub* menu yang digunakan dalam program.
2. Bagian *toolbar*, berisi *tool-tool* yang digunakan untuk berbagai keperluan dalam menggambar. seperti membuat tulisan, membuat garis, memutar gambar, dan lainnya.
3. Bagian *stencil*, bagian ini memiliki berbagai gambar *template* yang dapat langsung kita gunakan atau kita edit kembali ukurannya maupun keterangan gambar tersebut.
4. Bagian halaman desain, bagian ini digunakan sebagai bidang kerja kita dalam membuat gambar.
5. 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

1. ***Data flow diagram* (*DFD*)**

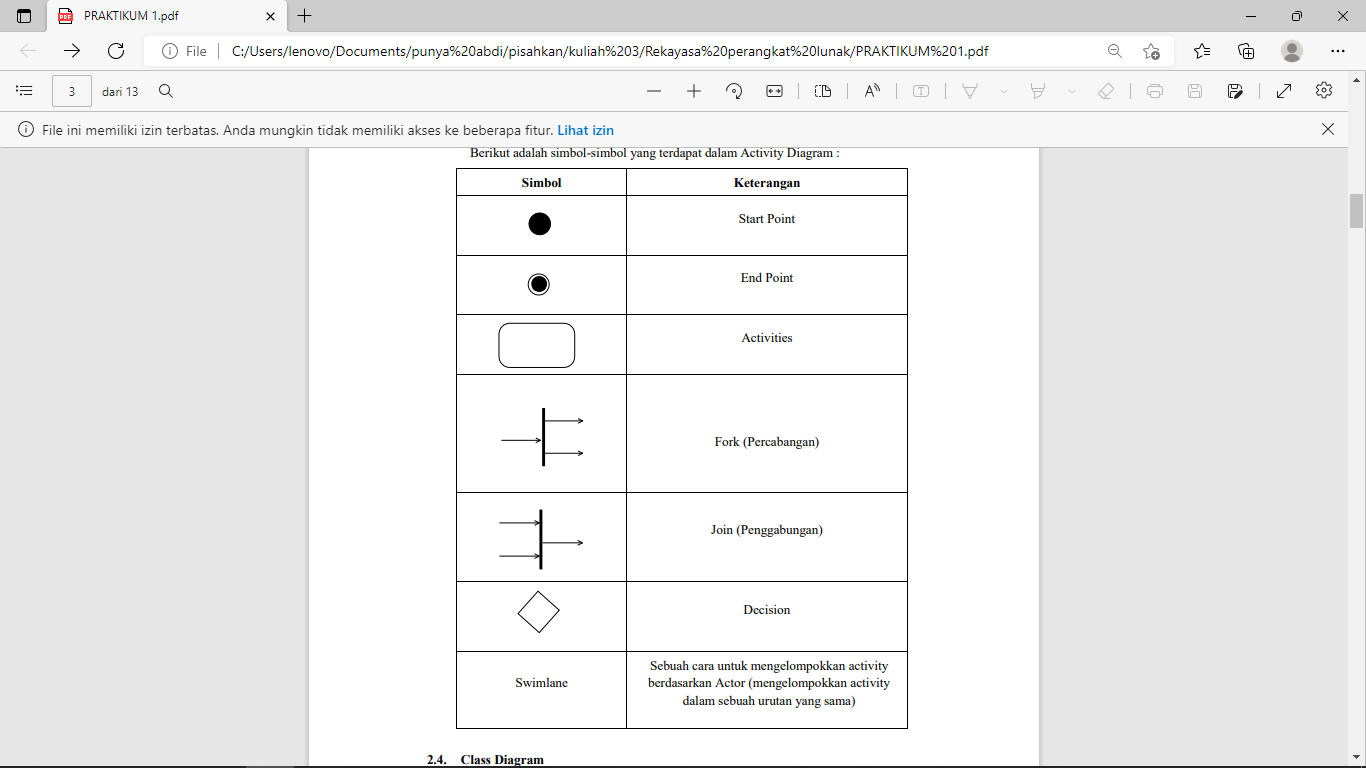
DFD menggambarkan arus data dari suatu sistem informasi, baik sistem lama maupun sistem baru secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut berada. Simbol yang digunakan dalam pembuatan *DFD* terbagi empat bagian, yaitu *Entity* (kesatuan luar), *Data flow* (arus data), *Process*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



1. ***Activity diagram***

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

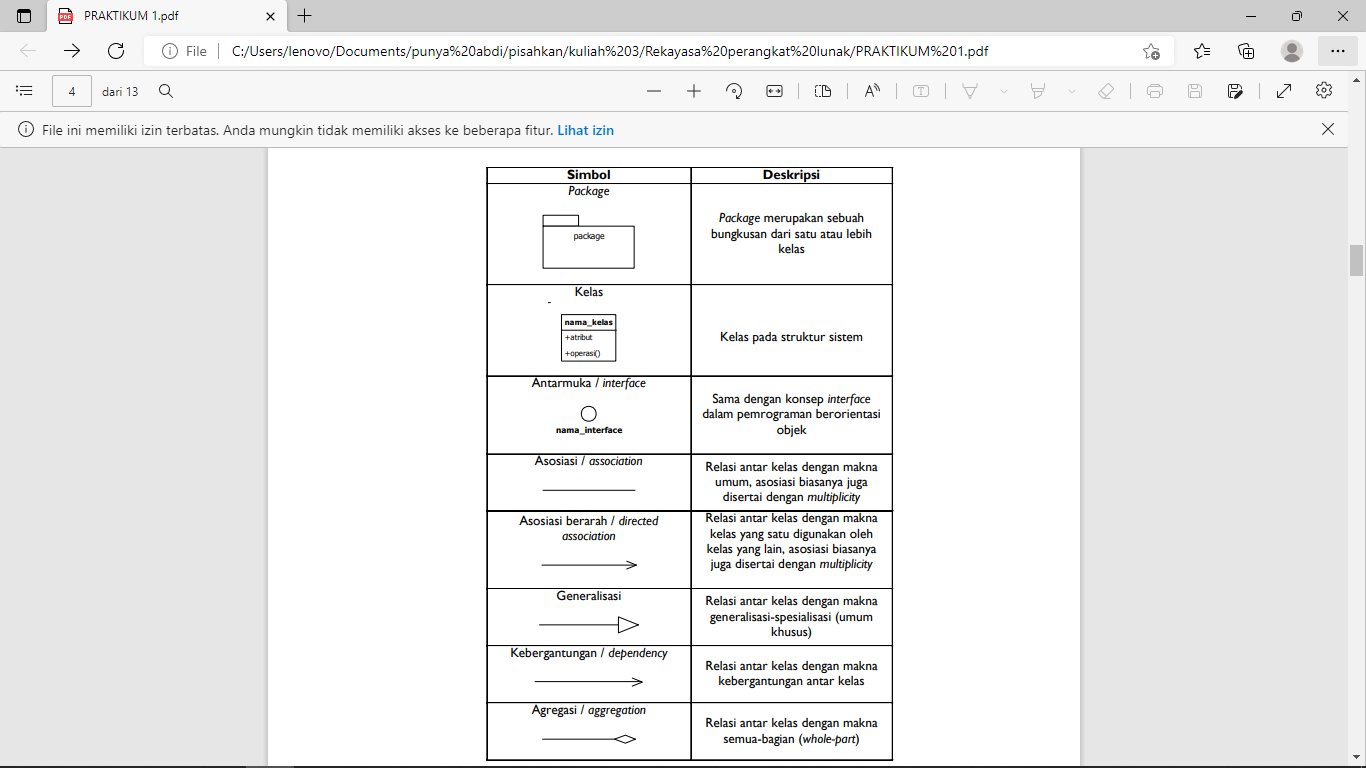
Berikut adalah simbol-simbol yang terdapat dalam *Activity Diagram* :



1. ***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 diinstansiasi 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).

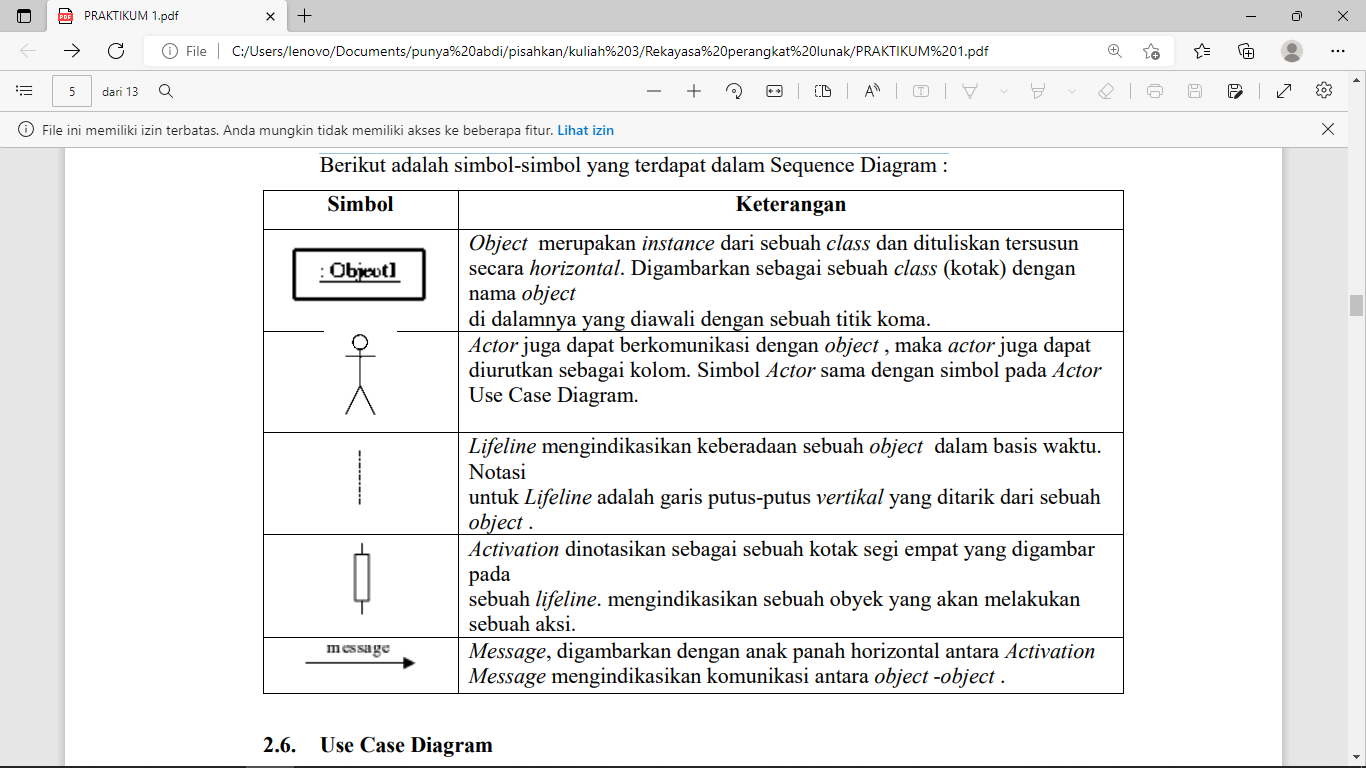
Berikut adalah simbol-simbol yang terdapat dalam *Class Diagram* :



1. ***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.

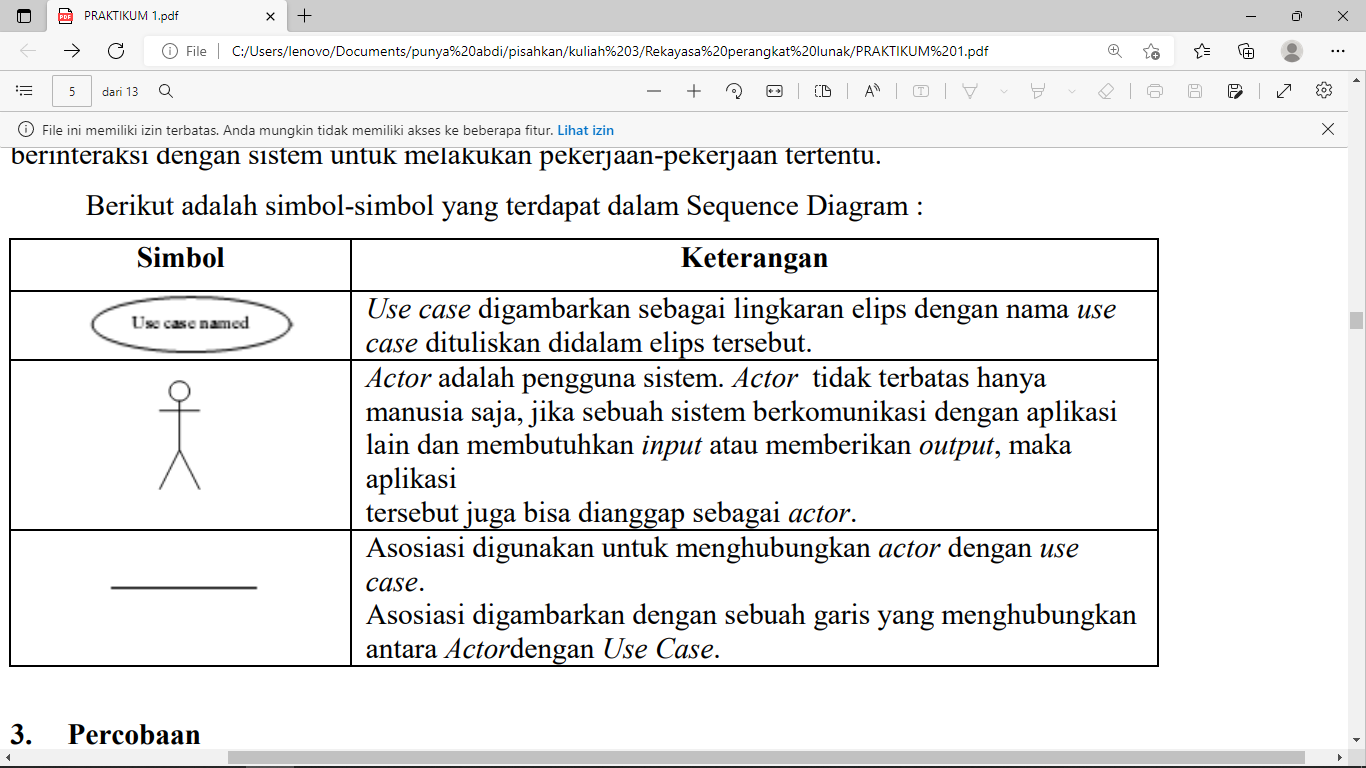
Berikut adalah simbol-simbol yang terdapat dalam *Sequence Diagram* :



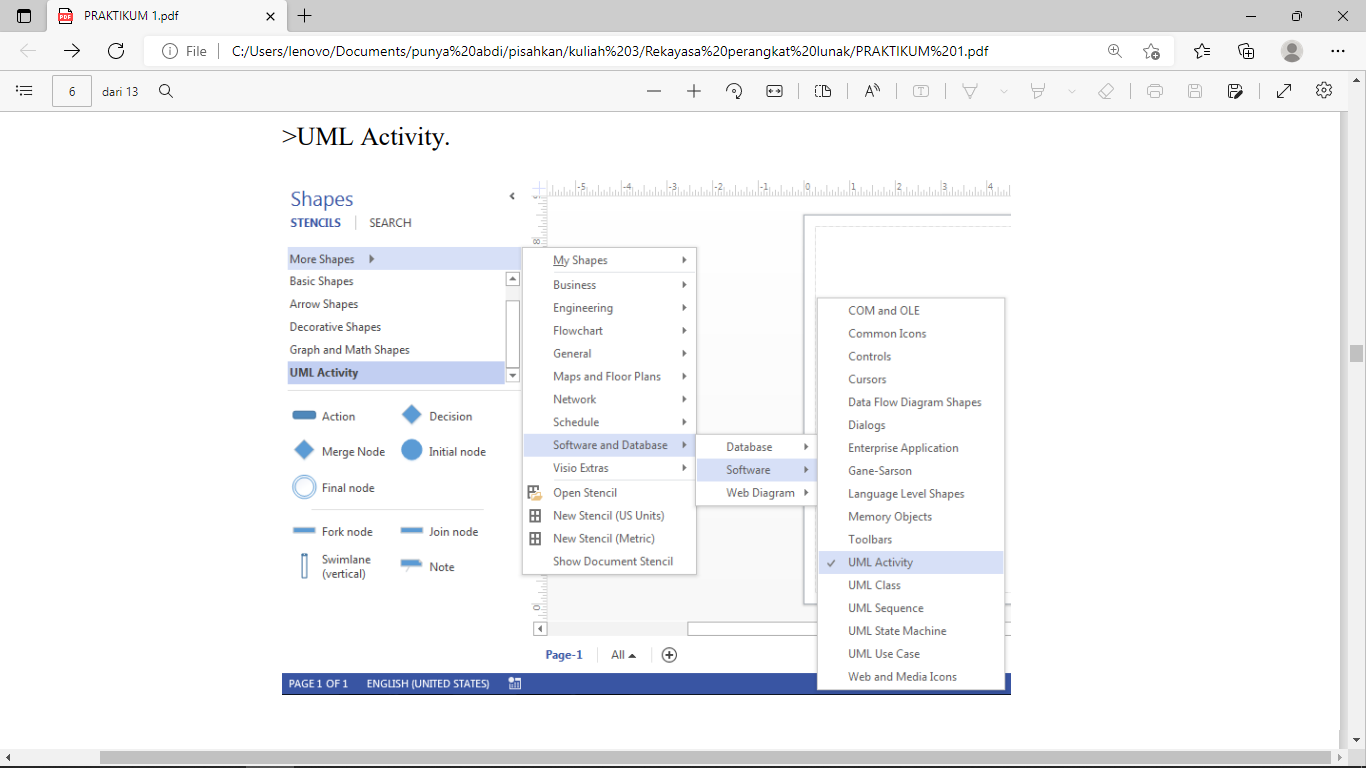
1. ***Use case diagram***

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.

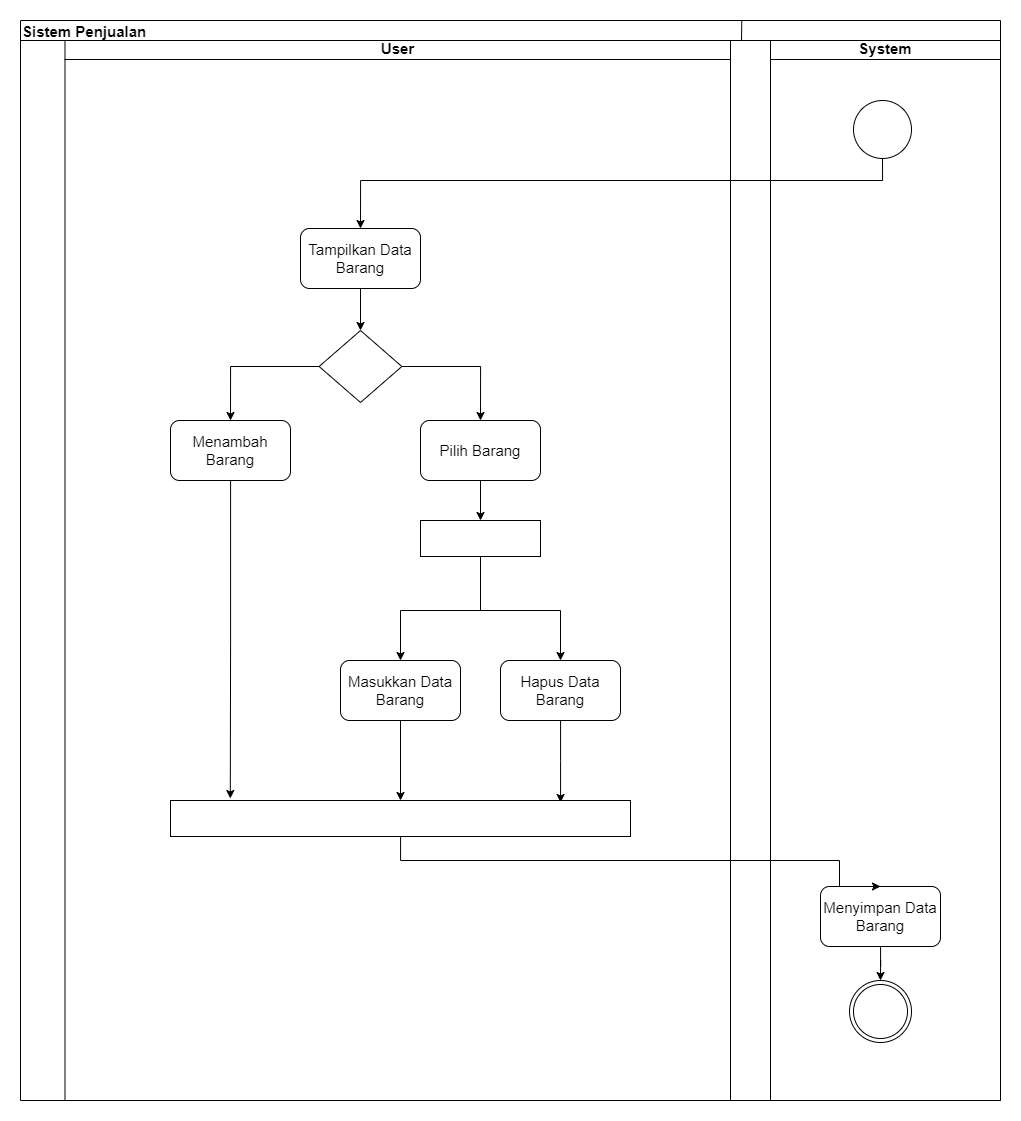
Berikut adalah simbol-simbol yang terdapat dalam *Use Case* *Diagram* :



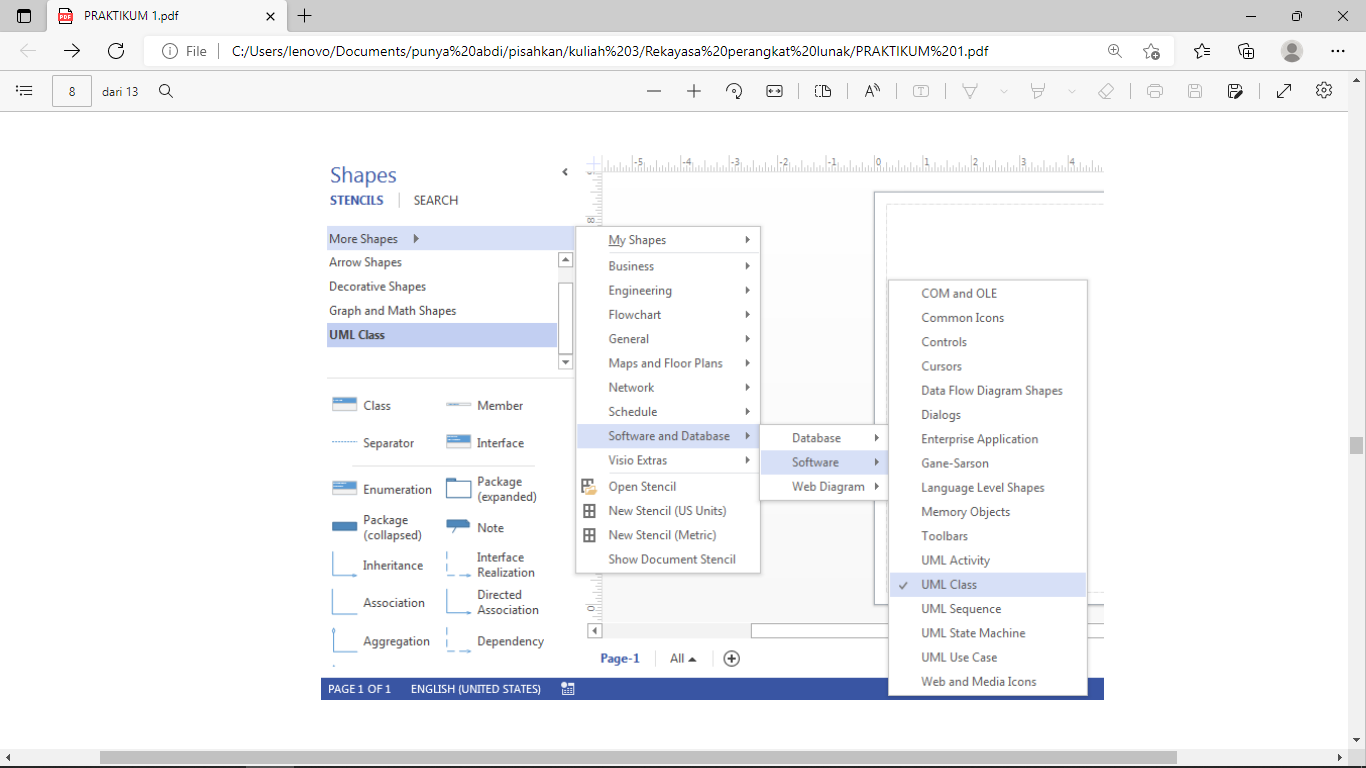
1. **LANGKAH KERJA**
2. ***Activity diagram***
3. Bukalah aplikasi *Ms. Visio* -> Pilih *Blank Drawing* -> Pilih *US Units* -> *Create.*
4. Pada bagian *Shapes*, pilih *More Shapes* -> *Software and Database* -> *Software* -> *UML Activity.*



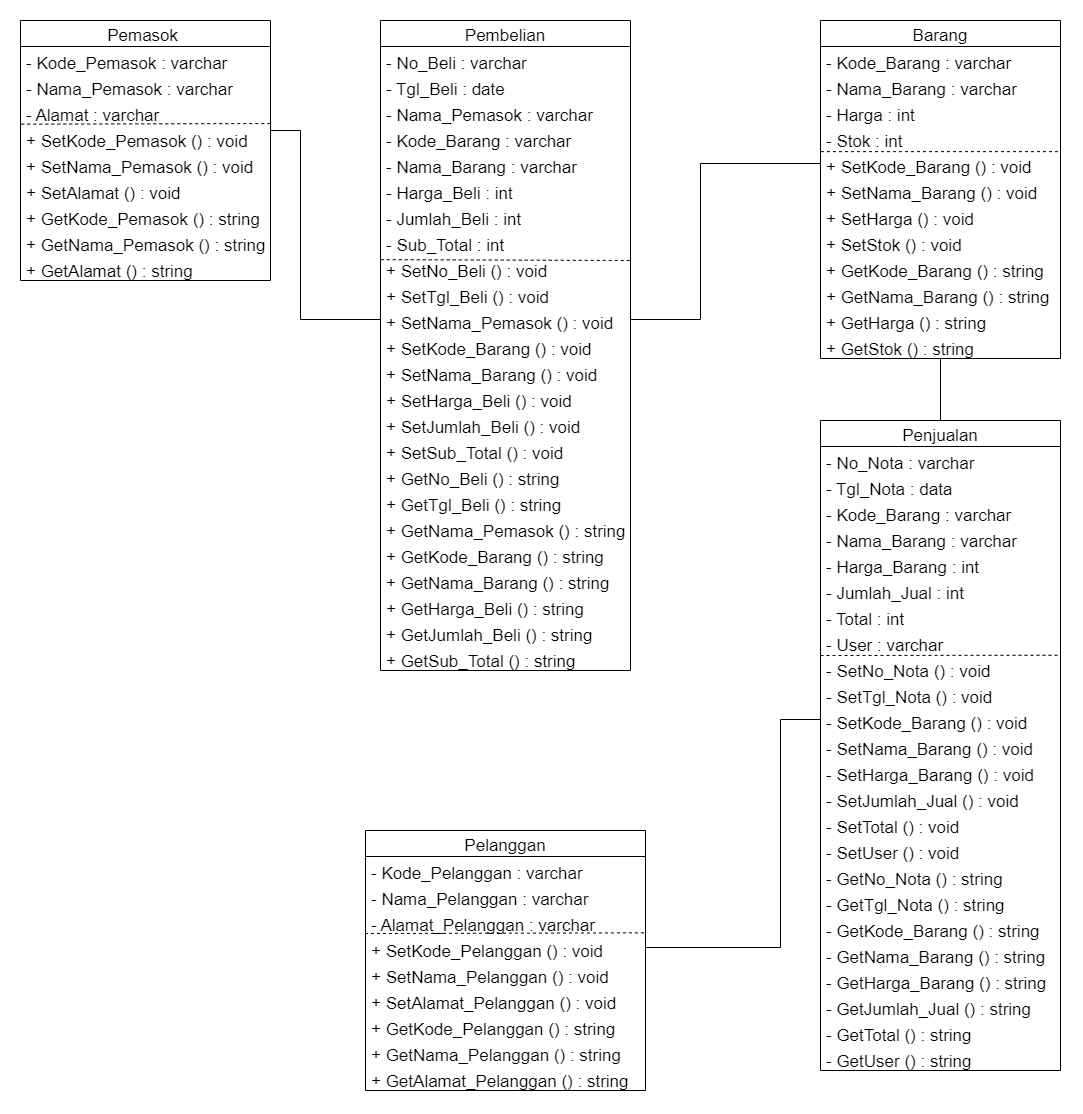
1. Pilihlah simbol-simbol yang terdapat pada *UML Activity*, lalu gambarkan *activity diagram* dari sistem penjualan sesuai dengan gambar di bawah ini :



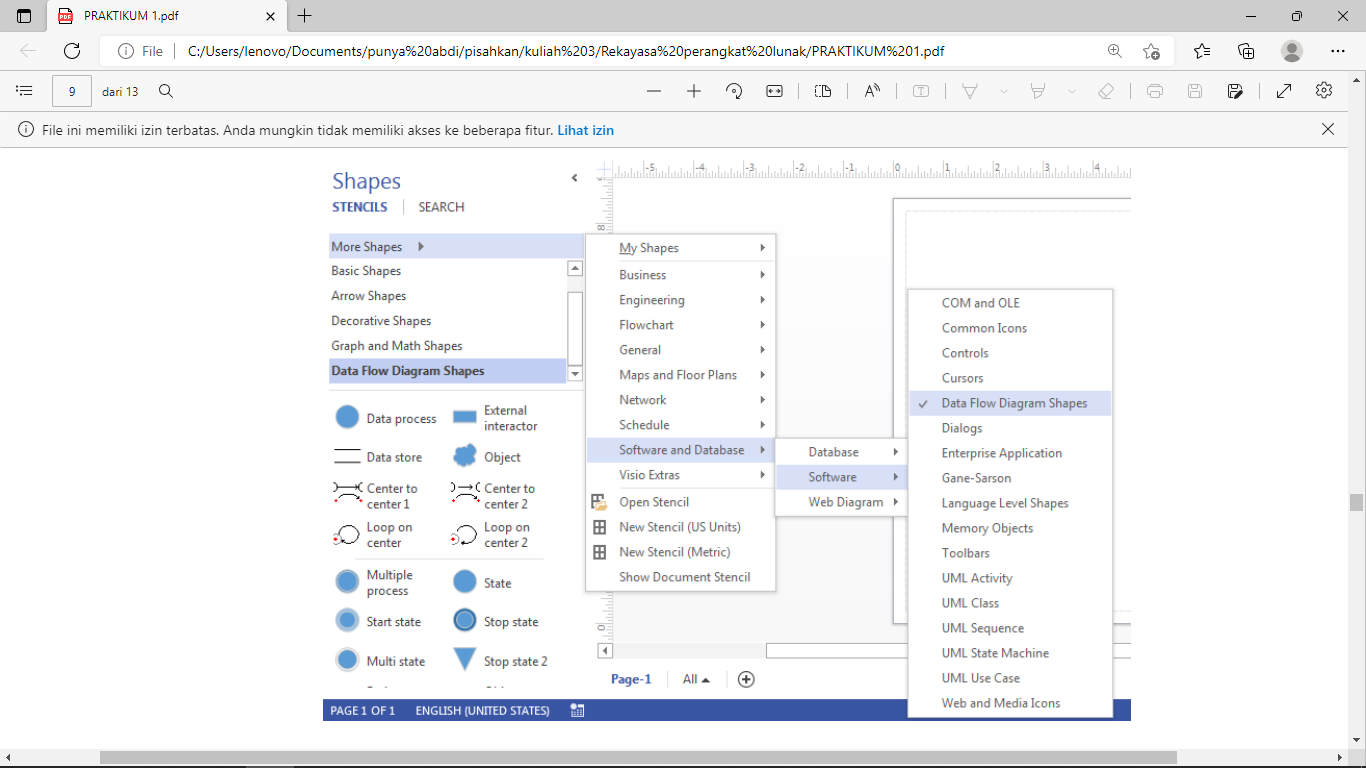
1. ***Class program***
2. Tambahkan halaman baru pada file yang sudah terbuka dengan cara pilih *menu Insert* -> *New Page.*
3. Pada bagian *Shapes*, pilih *More Shapes* -> *Software and Database* -> *Software -* > *UML Class.*



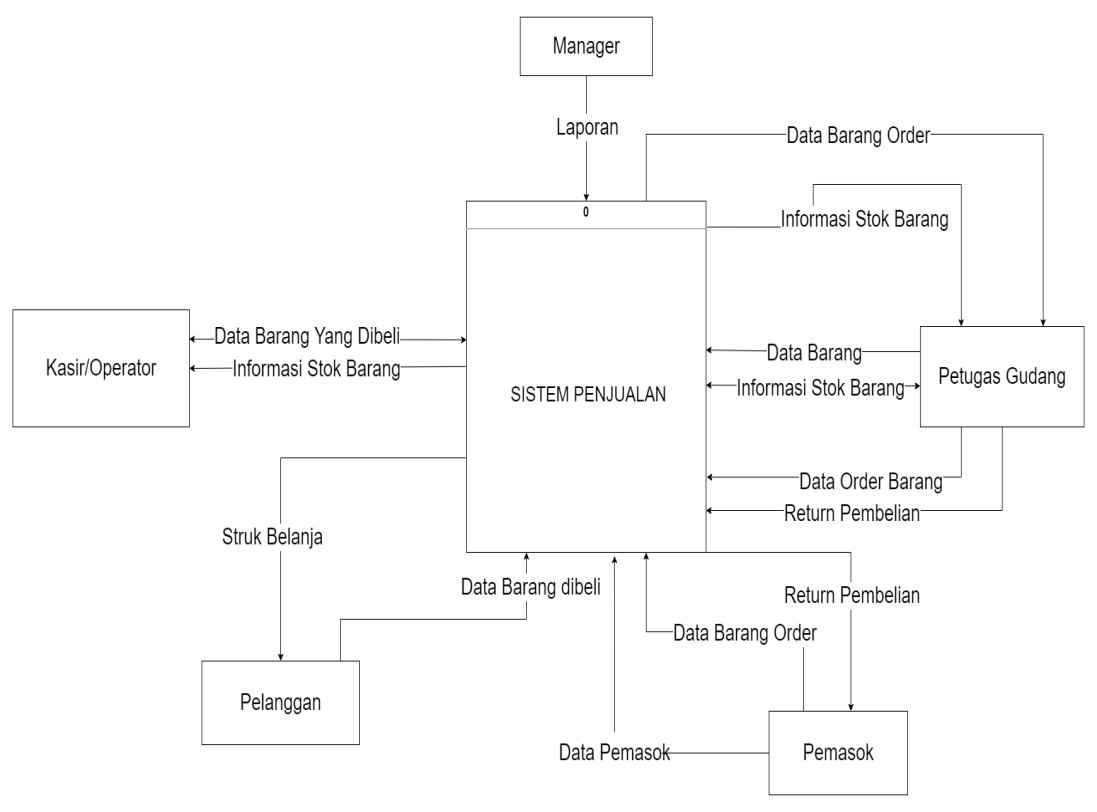
1. Pilihlah simbol-simbol yang terdapat pada *UML* *Class*, lalu gambarkan *class diagram* dari sistem penjualan sesuai dengan gambar di bawah ini :



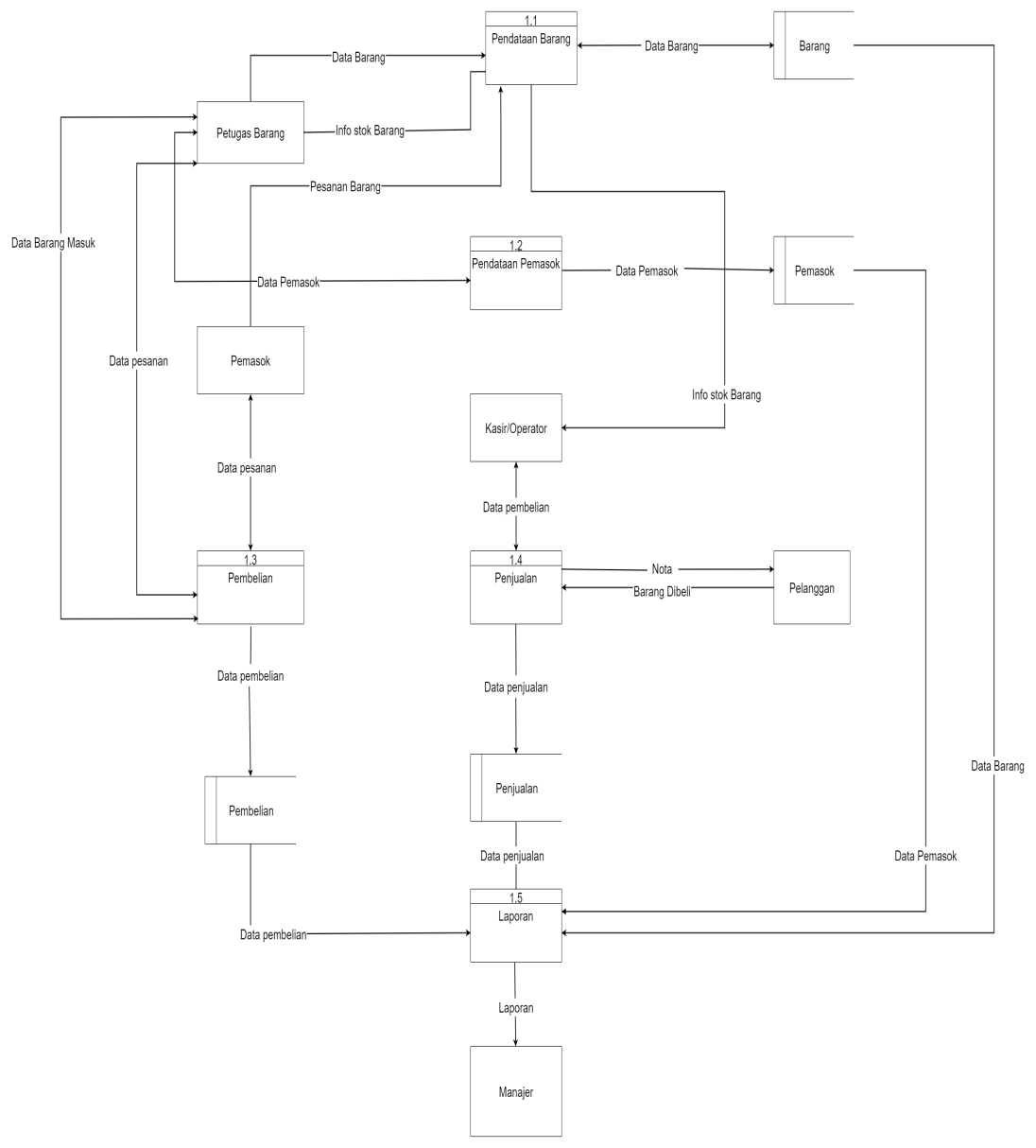
1. ***Data flow diagram* (*DFD*) *level* 0**
2. Tambahkan halaman baru pada file yang sudah terbuka dengan cara pilih menu Insert -> *New Page.*
3. Pada bagian *Shapes*, pilih *More Shapes* -> *Software and Database* -> *Software* -> *Data Flow Diagram Shapes.*



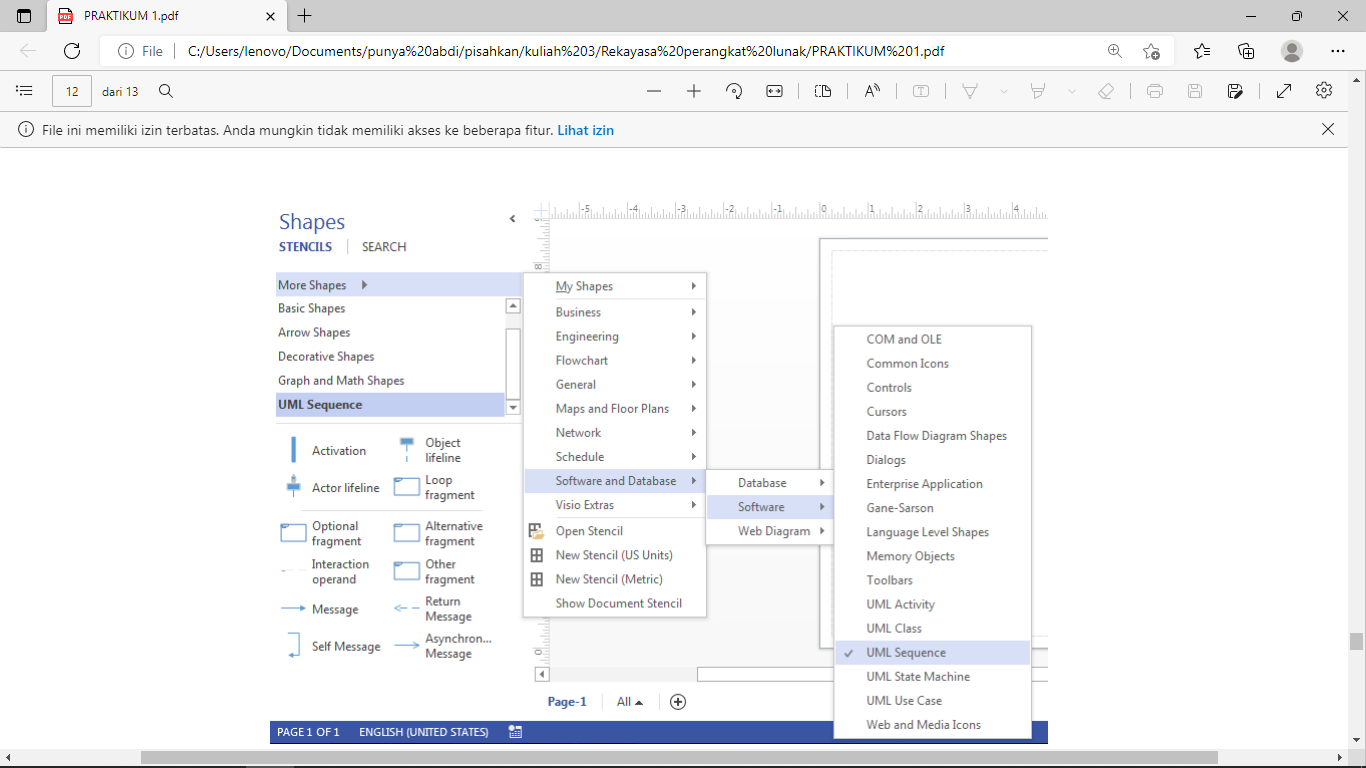
1. Pilihlah simbol-simbol yang terdapat pada *Data Flow Diagram Shapes*, lalu gambarkan *Contex Diagram* dari sistem penjualan sesuai dengan gambar di bawah ini :



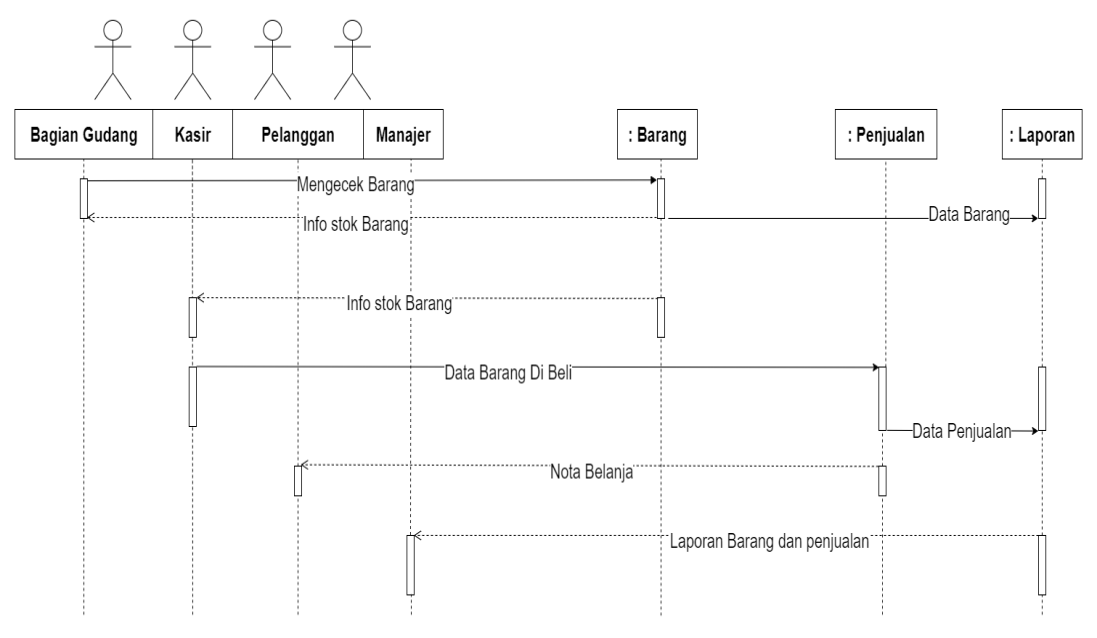
1. ***Data flow diagram* (*DFD*) *level* 1**
2. Tambahkan halaman baru pada file yang sudah terbuka dengan cara pilih menu *Insert* -> *New Page.*
3. Pilihlah simbol-simbol yang terdapat pada *Data Flow Diagram Shapes*, lalu gambarkan *DFD* dari sistem penjualan sesuai dengan gambar di bawah ini



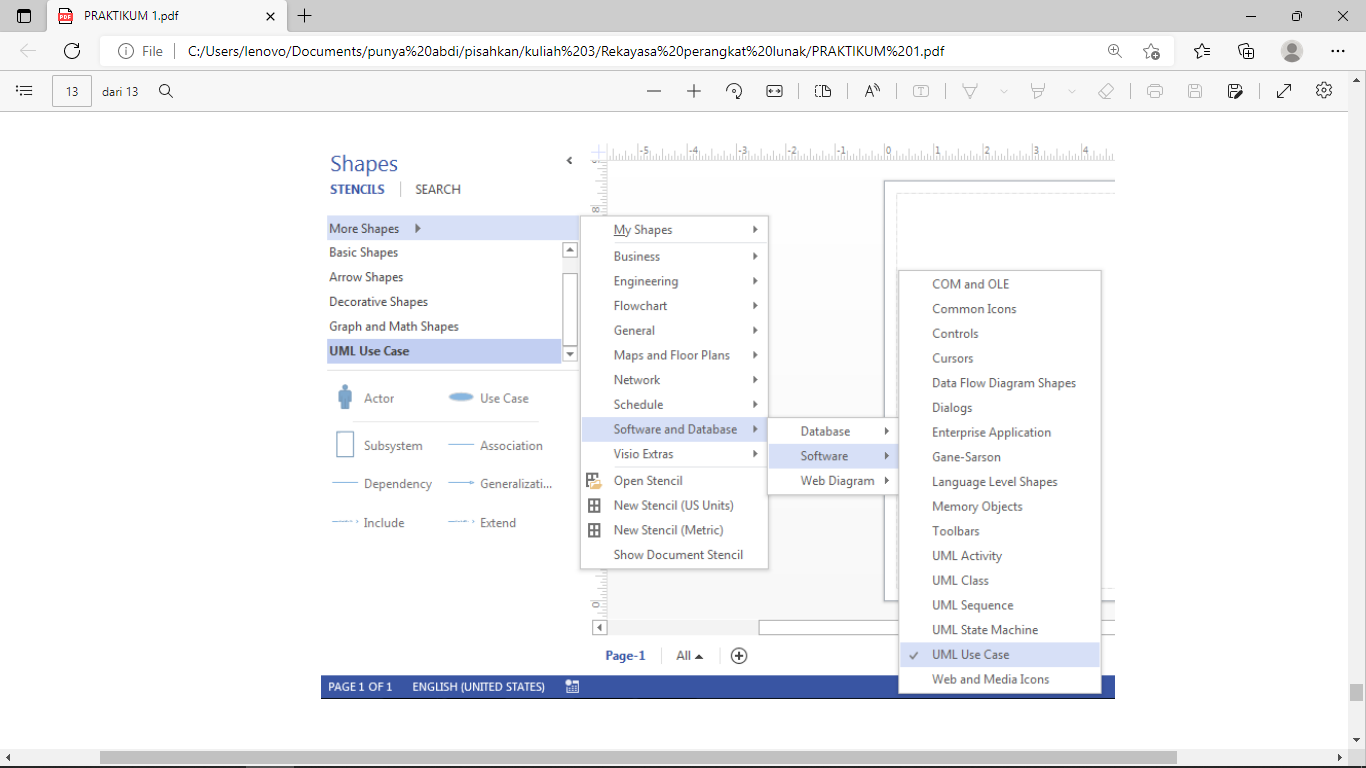
1. ***Sequence Diagram***
2. Tambahkan halaman baru pada *file* yang sudah terbuka dengan cara pilih *menu Insert* -> *New Page*.
3. Pada bagian *Shapes*, pilih More *Shapes*->*Software and Database*->*Software*- >*UML Sequence*



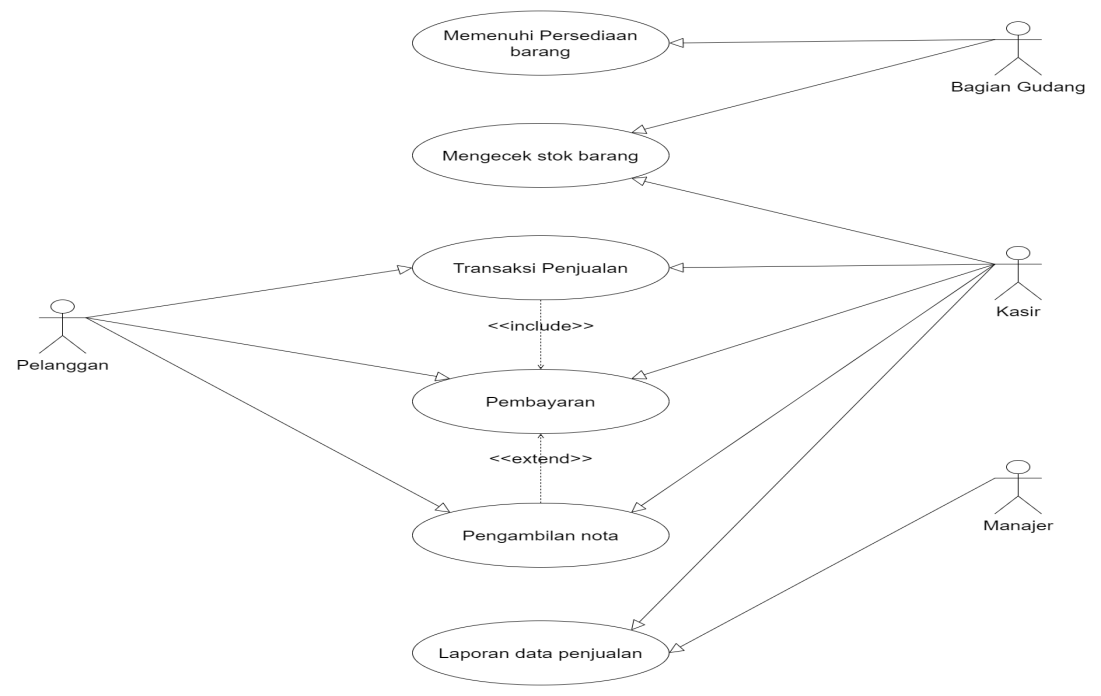
1. Pilihlah simbol-simbol yang terdapat pada *UML Sequence*, lalu gambarkan *Sequence Diagram* dari sistem penjualan sesuai dengan gambar di bawah ini :



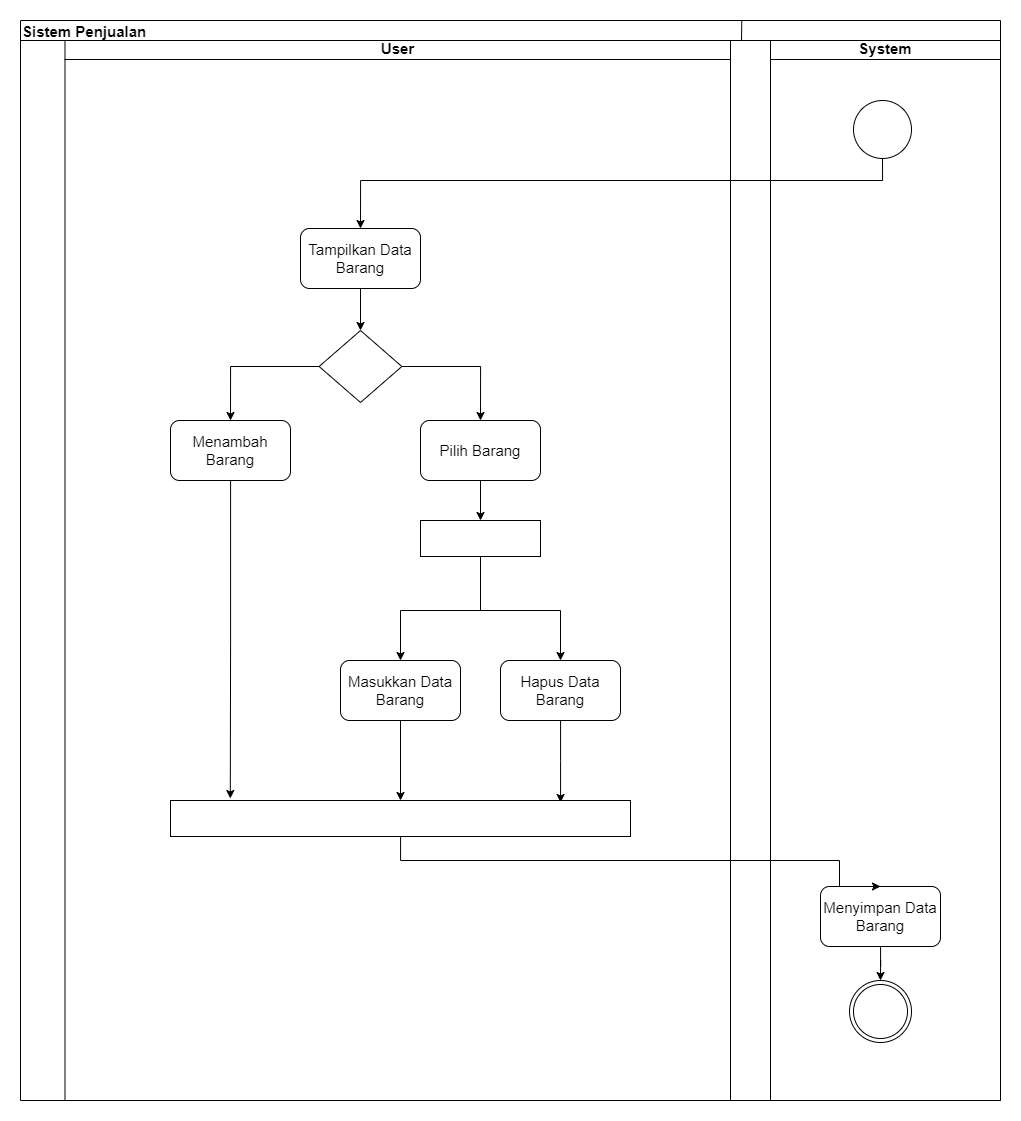
1. ***Use Case Diagram***
2. Tambahkan halaman baru pada *file* yang sudah terbuka dengan cara pilih *menu Insert* -> *New Page.*
3. Pada bagian Shapes, pilih *More Shapes* ->*Software and Database* -> *Software* -> *UML Use Case.*



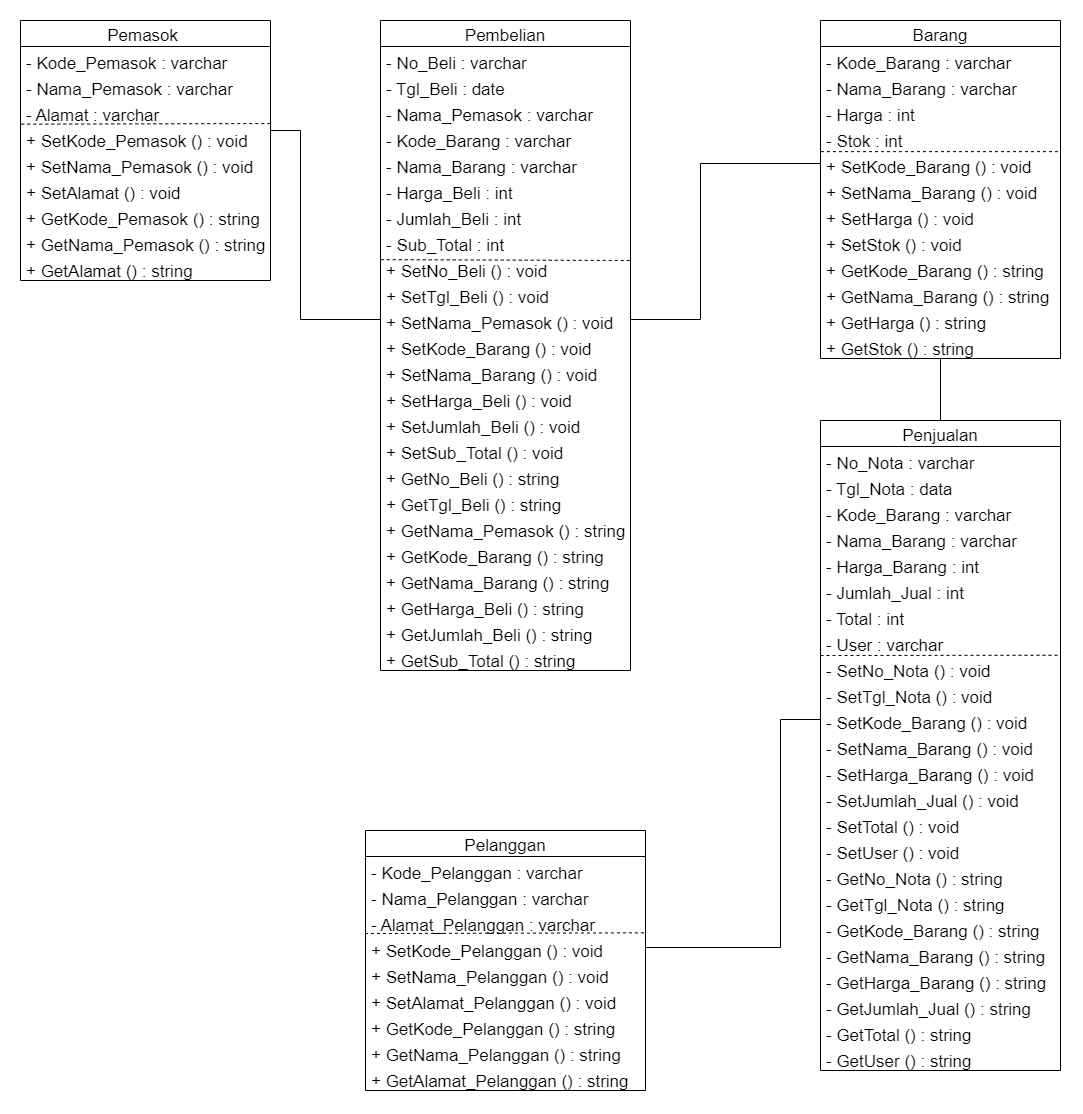
1. Pilihlah simbol-simbol yang terdapat pada *UML Use Case*, lalu gambarkan *Use Case Diagram* dari sistem penjualan sesuai dengan gambar di bawah ini :



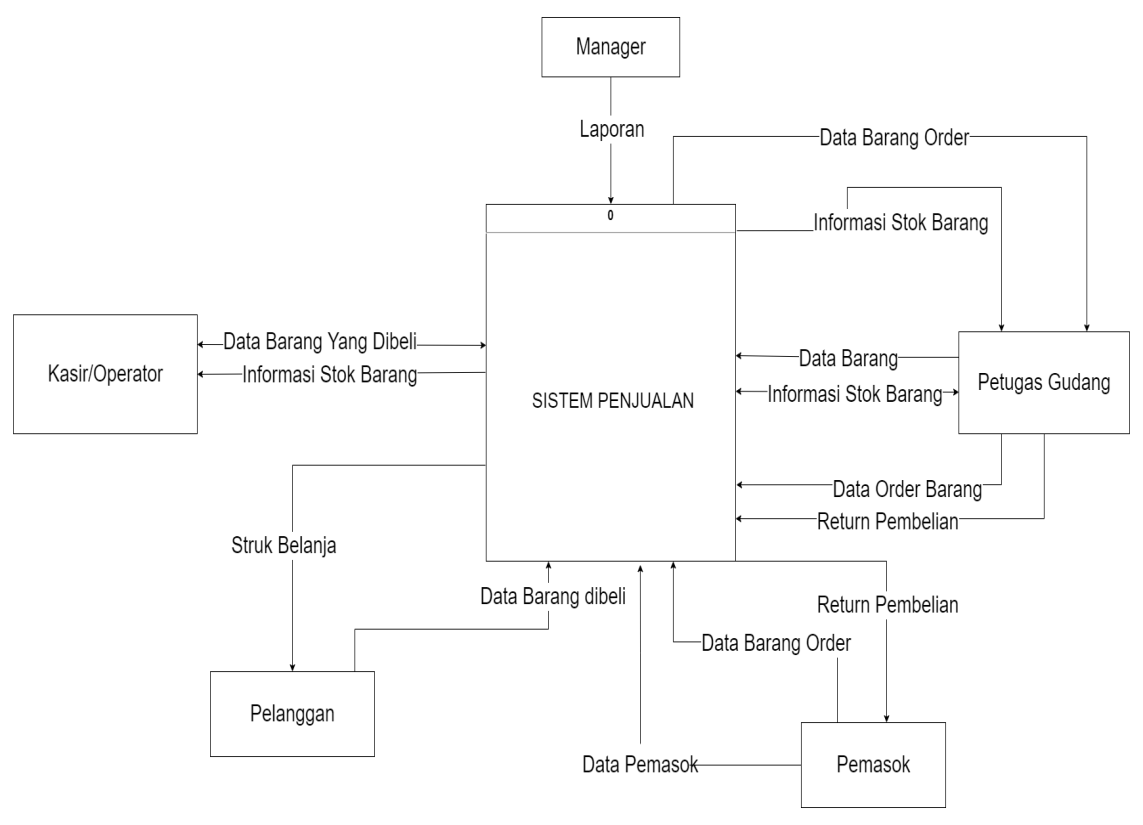
1. **HASIL PERCOBAAN**
2. ***Activity diagram***



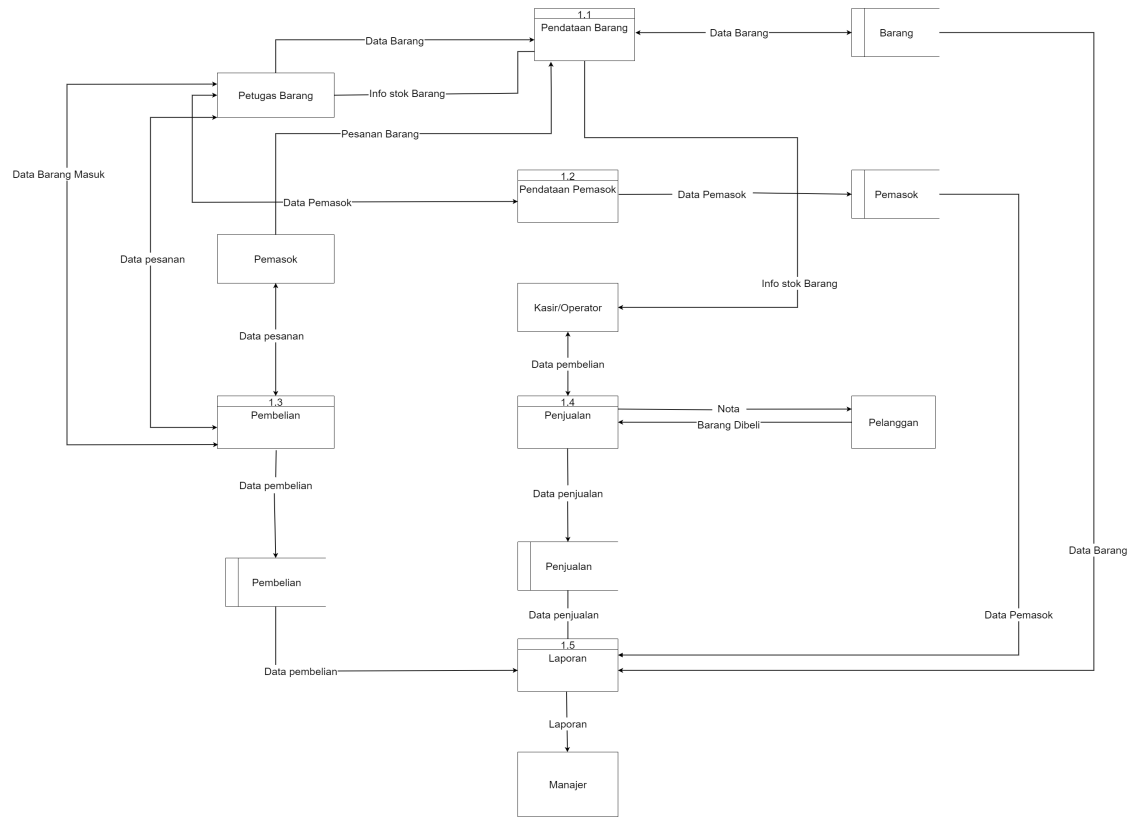
1. ***Class diagram***



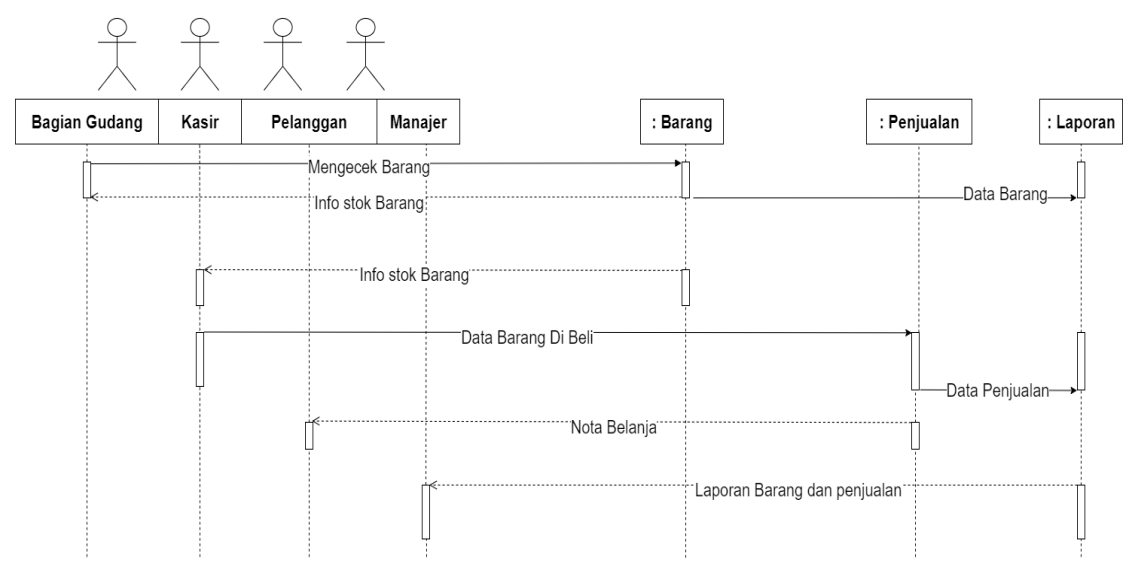
1. ***Data flow diagram* (*DFD*) *level* 0**



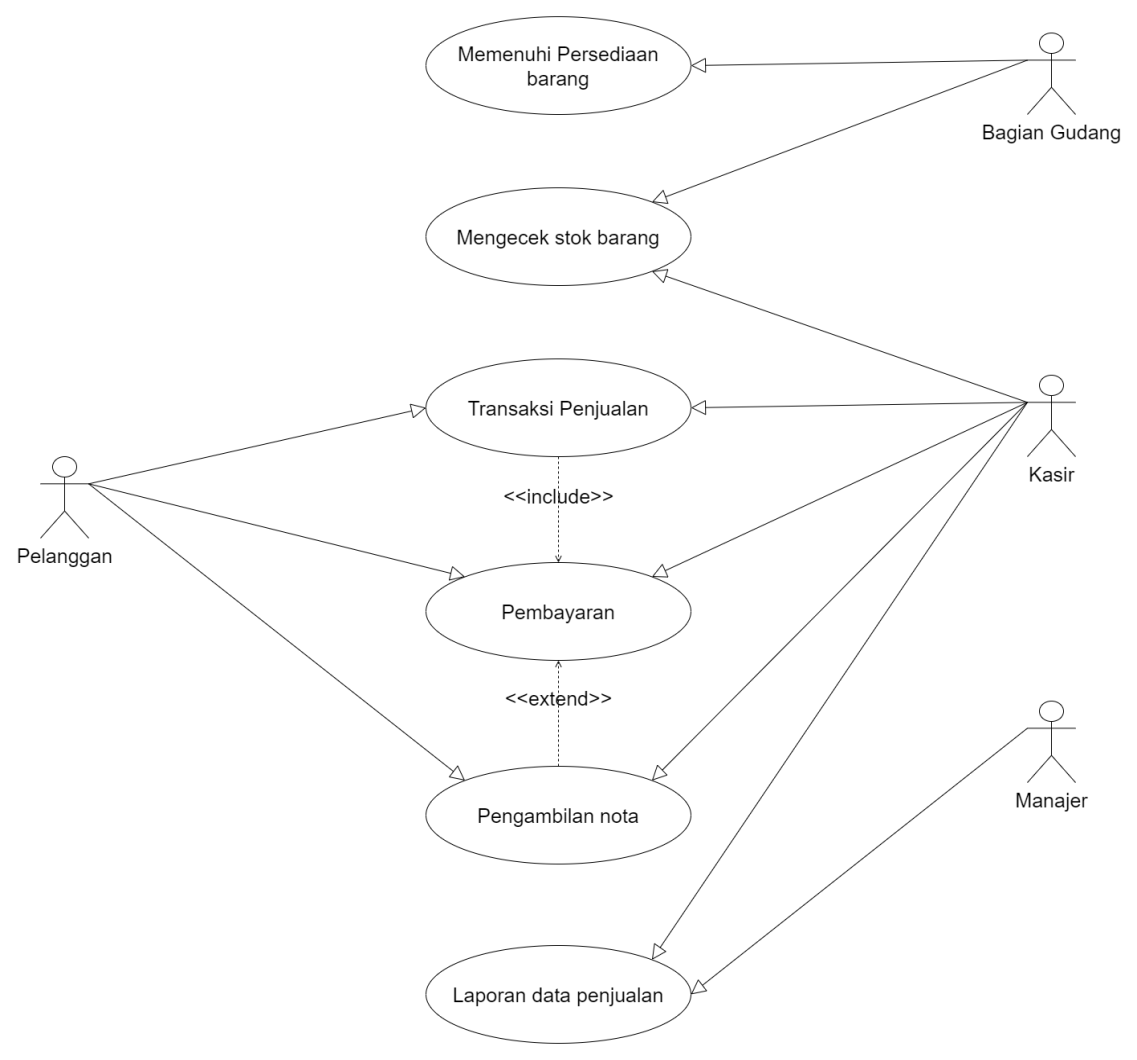
1. ***Data flow diagram* (*DFD*) *level* 1**



1. ***Sequence diagram***



1. ***Use case diagram***



1. **ANALISIS**

Setelah melakukan percobaan diatas, maka dapat dianalisa bahwa *Activity Diagram, Class Diagram, Data Flow Diagram(DFD),* dan *Use Case Diagram* memiliki fungsi masing-masing dan disebut sebagai “*UML*”,atau “*Unified Modeling Language”* yang merupakan suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek.

Pada “*Activity Diagram”* berfungsi untuk menampilkan urutan langkah- langkah setiap proses yang ingin dibuat. Dimana diagram tersebut menggambarkan tentang sistem penjualan. Pada diagram tersebut, dijelaskan bahwa ”*System*” akan memulai jalannya program yaitu “Sistem penjualan”. Kemudian “*User*” akan memperlihatkan “Tampilan Data Barang”, lalu masuk ke “*Decision”* atau “*User*” dimana akan dihadapkan oleh dua pilihan yaitu “Menambah Barang” atau “Pilih Barang”, jika “*User*” memilih “Menambah barang” maka datanya akan langsung diproses dan pada bagian “*System*” kemudian langsung memproses “*Menyimpan Data Barang*” dan program “Sistem Penjualan” berakhir. Namun, jika “*User*” memilih “Pilih Barang”, maka datanya akan diproses dan akan mendapat dua pilihan lagi yaitu “Masukan Data Barang” dan “Hapus Data Barang”. Tetapi apapun yang dipilih oleh “*User*”, pasti langsung diarahkan ke “*System*” yaitu “Menyimpan Data Barang” dan program “Sistem Penjualan” pun berakhir.

Selanjutnya pada bagian “*Class Diagram*” dimana diagram tersebut menggambarkan tentang sistem penjualan. Pada diagram tersebut terdapat “5” kelas yaitu “Pemasok”, “Pembelian”, “Barang”, “Penjualan” dan “Pelanggan”. Pada kelas “Pemasok” terdapat atribut yaitu diantaranya “Kode\_Pemasok”, “Nama\_Pemasok” dan “Alamat”. Lalu pada kelas “Pembelian” terdapat atribut yaitu diantaranya “No\_Beli”, “Tgl\_Beli”, “Nama\_Pemasok”, “Kode\_Barang”, “Nama\_Barang”, “Harga\_Beli”, “Jumlah\_Beli” dan “Sub\_Total”. Selanjutnya pada kelas “Barang” terdapat atribut yaitu “Kode\_Barang, “Nama\_Barang”, “Harga” dan “Stok”. Kemudian pada kelas “Penjualan” terdapat atribut yaitu “No\_Nota”, “Tgl\_Nota”, “Kode\_Nota”, “Nama\_Barang”, “Harga\_Barang”, “Jumlah\_Jual”, “Total” dan “*User*”. Sedangkan pada kelas “Pelanggan” terdapat atribut yaitu “Kode\_Pelanggan”, “Nama\_Pelanggan” dan “Alamat\_Pelanggan”. Didalam kelas tersebut terdapat beberapa tipe data yaitu “*varchar*”, “*void*”, “*int*”, ”*string*” dan “*date*”. Setiap kelas juga mempunyai “2” operasi dimasing-masing atribut yaitu “*Set*” dan “*Get*”. Dimana “*Set*” bertipe data “*void*” sedangkan “*Get*” bertipe data “*string*”.

Pada bagian “*Data Flow Diagram* (*DFD*) *Level* 0” terdapat “5” entitas yaitu ada “*Manager*”, “Kasir/Operator”, “Pelanggan”, “Pemasok” dan “Petugas Gudang”. Dimana pada diagram tersebut dibuat versi “Level 0”. Pada fungsi “Manager” itu untuk menerima laporan dari sistem penjualan. Lalu pada fungsi “Kasir/Operator” gunanya untuk menerima informasi stok barang dan data barang yang dibeli. Kemudian fungsi “Pelanggan” untuk menerima struk belanja dan memberi data yang dibeli. Selanjutnya pada fungsi “Pemasok” yaitu menerima *return* pembelian dan memberi data pemasok dan data barang diorder. Sedangkan pada fungsi “Petugas Gudang” gunanya untuk menerima data barang yang diorder, informasi stok barang, data pemasok, memberi data barang, data order barang sampai *return* pembelian.

Pada bagian ”*Data Flow Diagram* (*DFD*) *Level* 1” terdapat 5 entitas didalamnya yaitu ada “Petugas Gudang”, “Pemasok”, “Kasir/Operator”, “Pelanggan” dan “Manajer”. Fungsi “Petugas Gudang” yaitu untuk mengirimkan data barang dan info stok barang kepada “1.1 Pendataan Barang” dan juga menerima data barang yang masuk, data pesanan, dan data pemasok. Kemudian pada fungsi “Pemasok” gunanya untuk mengirimkan pesanan barang dari “1.1 Pendataan Barang” dan menerima data pesanan dari “1.3 Pembelian”. Lalu untuk fungsi “Kasir/Operator” yaitu mengirimkan data pembelian dari “1.4 Penjualan” dan menerima info stok barang dari “1.1 Pendataan Barang”. Selanjutnya pada fungsi “Pelanggan” gunanya untuk menerima nota dan memberi barang yang dibeli dari ”1.4 Penjualan”. Sedangkan pada fungsi “Manajer” yaitu menerima laporan dari “1.5 Laporan”.

Selanjutnya yaitu “*Sequence diagram*” dimana terdapat 4 aktor yaitu “Bagian Gudang”, “Kasir”, “Pelanggan” dan “Manajer” dan juga terdapat 3 objek yaitu “: Barang”, “: Penjualan” dan “: Laporan”. Untuk aktor “Bagian Gudang” berfungsi mengecek barang pada objek “: Barang” dan menerima info stok barang dan data barang dari objek “: Barang” akan dikirimkan langsung ke objek “: Laporan”. Pada aktor “Kasir” fungsinya untuk menerima info stok barang dari objek “: Barang” dan juga memberikan data barang yang di beli ke objek “: Penjualan”. Setelah itu, data penjualan tersebut akan diterima oleh objek “: Laporan”. Selanjutnya yaitu aktor “Pelanggan” berfungsi untuk menerima nota belanja dari objek “: Penjualan”. Dan aktor terakhir yaitu “Manajer” dimana kegunaanya untuk mendapatkan laporan barang dan penjualan dari objek “: Laporan”.

Pada bagian “*Use Case Diagram*” terdapat 4 jenis aktor yaitu “Pelanggan”, “Bagian Gudang”, “Kasir” dan “Manajer”. Aktor “Bagian Gudang” berfungsi Untuk memenuhi persediaan barang dan mengecek stok barang. Aktor “Kasir” berfungsi untuk mengecek stok barang, transaksi penjualan, pembayaran, pengambilan nota dan laporan data penjualan. Aktor “Pelanggan” berfungsi untuk melakukan transaksi pernjualan, pembayaran dan pengambilan nota. Dan aktor “Manajer” berfungsi untuk melihat laporan data penjualan.

Pada percobaan kali ini juga dapat dianalisiskan bahwa untuk membuat sebuah tampilan atau gambar dari UML, bisa menggunakan aplikasi Microsoft Visio, yang dimana aplikasi ini berfungsi untuk membuat diagram secara baik dan menarik. Bahkan segala fitur UML dapat diperoleh oleh aplikasi ini, sehingga aplikasi mampu mempermudah proses kerja manusia untuk membuat suatu diagram. Selain itu juga Microsoft Visio juga dapat memungkinkan untuk mendokumentasikan alur kerja atau proses dengan mudah. Dan perlu diketahui semua percobaan bagian UML diatas dibuat oleh aplikasi Microsoft Visio.

1. **KESIMPULAN**

Berdasarkan percobaan diatas maka dapat disimpulkan bahwa *UML* (*Unified Modeling Language*) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. Yang dimana pada diagram tersebut memiliki banyak jenis dan bentuk yang pasti berbeda-beda disetiap diagramnya. Jenis-jenis diagram tersebut antara lain *Data Flow Diagram* (*DFD*), *Activity Diagram*, *Class Diagram*, *Sequence Diagram*, dan *Use Case Diagram*. Dan ini merupakan bagian yang penting untuk dipahami agar nantinya bisa menjelaskan setiap UML(*Unified Modeling Language*) Diagram dengan baik dan jelas. Supaya setiap proses yang dijalankan mudah dipahami kedepannya.