

MODUL III

CLASS DIAGRAM

1. Tujuan Praktikum

- Praktikan mengenal tentang Class Diagram
- Praktikan dapat membuat Class Diagram
- Praktikan mengimplementasikan Class Diagram pada kasus yang ada

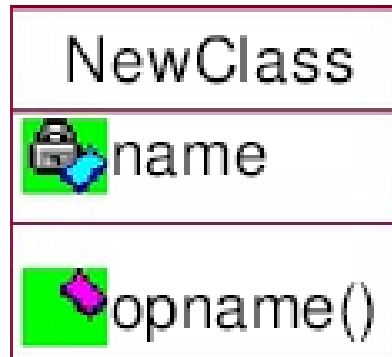
2. Dasar Teori

2.1. Pengertian Class Diagram

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 (metode/fungsi). Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Class memiliki tiga area pokok :

- Nama (dan *stereotype*)
- Atribut
- Metode Atribut dan metoda dapat memiliki salah satu sifat berikut :
- Private, tidak dapat dipanggil dari luar class yang bersangkutan

- Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
- Public, dapat dipanggil oleh siapa saja



Class dapat merupakan implementasi dari sebuah interface, yaitu class 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. Sesuai dengan perkembangan class model, class dapat dikelompokkan menjadi package. Kita juga dapat membuat diagram yang terdiri atas package.

2.2. Hubungan Antar Class

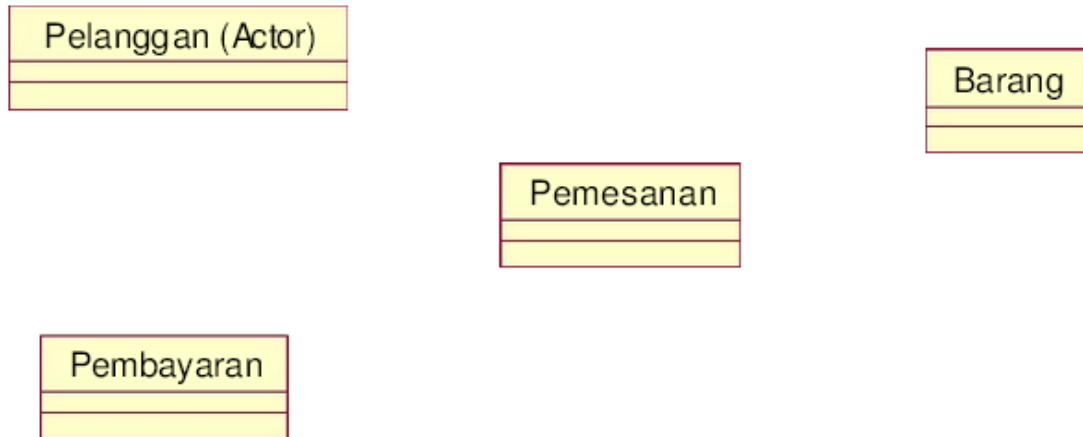
- Asosiasi, yaitu hubungan statis antar class. Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui eksistensi class lain. Panah navigability menunjukkan arah query antar class. (Direpresentasikan dalam bentuk garis penghubung)
- Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”). Digambarkan dengan garis dan belah ketupat di ujung.

- Pewarisan, yaitu hubungan hirarkis antar class. Class dapat diturunkandari class lain dan mewarisi semua atribut dan metoda class asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari classyang diwarisinya.
- Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-passingdari satu class kepada class lain. Hubungan dinamis dapat digambarkan dengan menggunakan sequence diagram yang akan dijelaskan kemudian.
- Ketergantungan – bentuk lemah yang menunjukkan relasi antara klien dan supplier dimana klien tidak memiliki pengetahuan semantik dari supplier. Sebuah ketergantungan akan mengatakan “Aku butuh servis darimu, tapi aku tidak tahu kalau kamu ada.” (Direpresentasikan dalam garis putus-putus yang diberi panah)

3. Langkah Praktikum

3.1. Membuat Class Diagram dan Menghapus Class Diagram

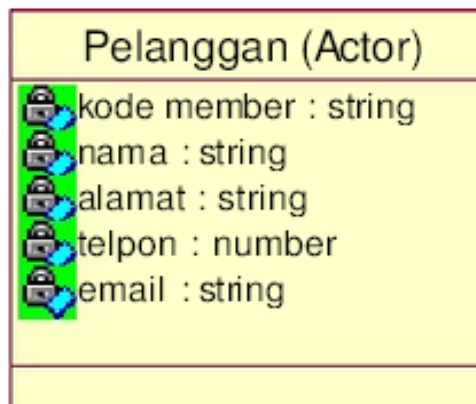
1. **Klik kanan** pada browser Logical View pilih **New**  **Diagram** dan beri nama **BookClass**
2. **Klik ganda** pada diagram baru yang telah dibuat untuk membukanya.
3. **Klik** toolbar Class pada menu toolbar lalu taruh **klik** pada jendela diagram.
4. Buatlah **empat (4) buah** Class baru dan beri nama dengan cara **klik** pada namanya :
 - Pelanggan (Actor)
 - Barang
 - Pemesanan
 - Pembayaran



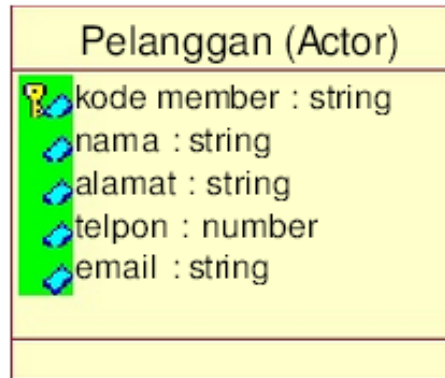
5. Untuk menghapus Class Diagram **klik kanan** pada Class Diagram yang akan dihapus lalu pilih **Delete**.

3.2. Menambahkan atribut pada Class Diagram dan Menghapusnya

1. Ubah pada class **Pelanggan (Actor)** untuk menambahkan atribut dengan cara **klik kanan** pada class tersebut pilih **New Attribute**
2. Ubah nama atribut seperti dibawah ini:



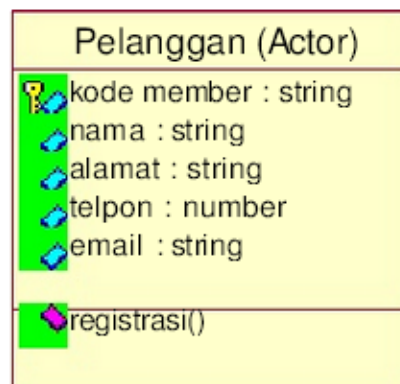
3. Kemudian ubah **visibility** pada atribut dengan cara **klik** pada ikon atribut dan pilih **jenisnya**. Dan hasilnya seperti dibawah ini:



4. Untuk menghapus Atribute pada Class Diagram **blok atribut** yang akan dihapus, kemudian **klik kanan** pilih **Delete** lalu **klik** di sembarang tempat diluar class yang bersangkutan, maka akan muncul dialog untuk menghapus atau tidak, kemudian pilih **Yes**.

3.3. Menambahkan operasi pada Class Diagram dan Menghapusnya

1. Kemudian tambahkan operasi pada class **Pelanggan(Actor)** dengan cara klik kanan pada class tersebut pilih **New Operation**.
2. Lalu ubah nama operasi seperti dibawah ini:

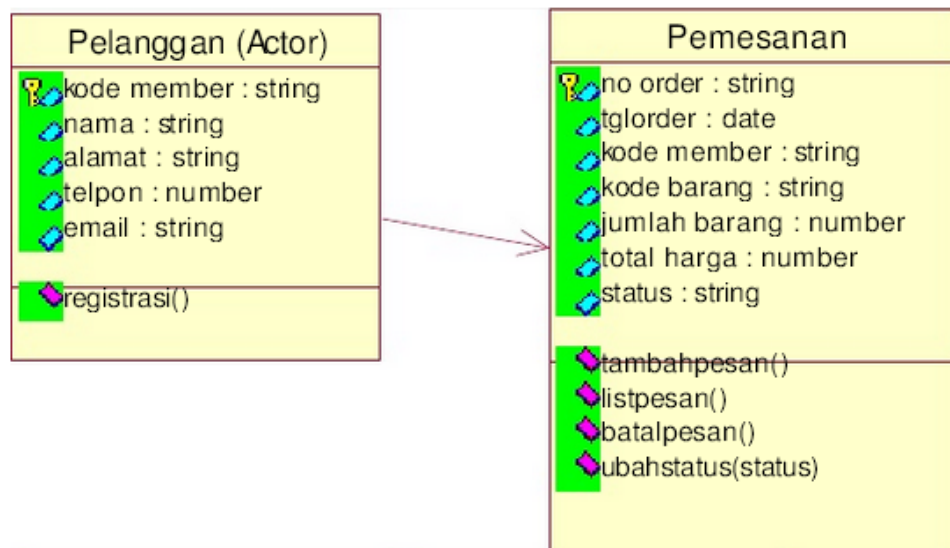


- Untuk menghapus operasi lakukan langkah yang sama ketika akan menghapus atribut.
- Kemudian buat class-class yang lain seperti dibawah ini:



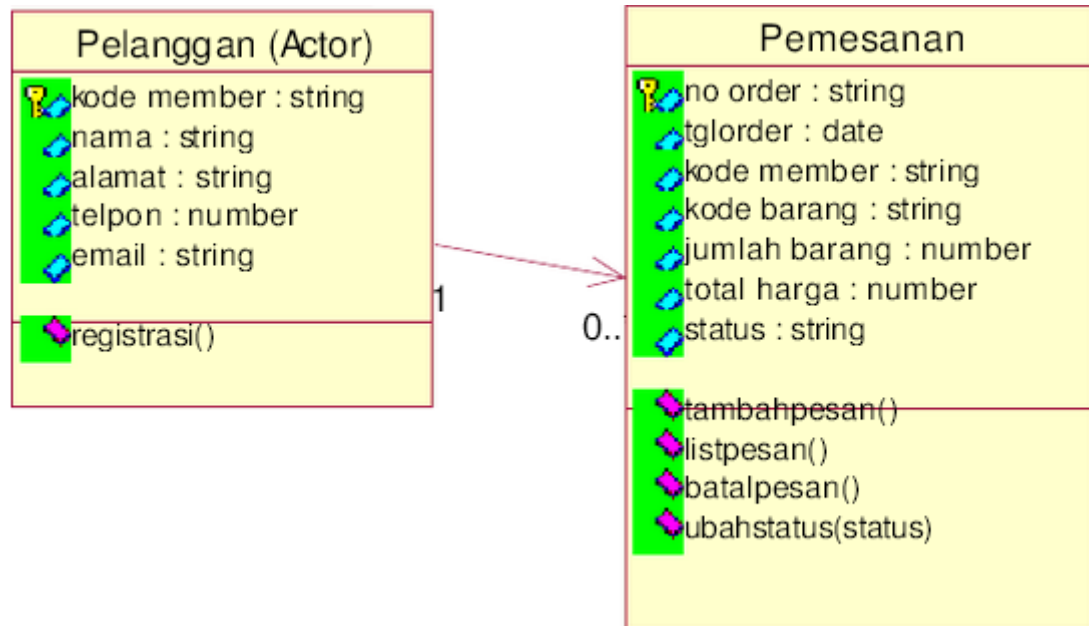
3.4. Membuat relasi antar Class

- Kemudian buatlah relasi antar class dengan cara pilih toolbar **Undirectional Associate**.
- Klik pada salah satu class dan drag ke class yang lain.

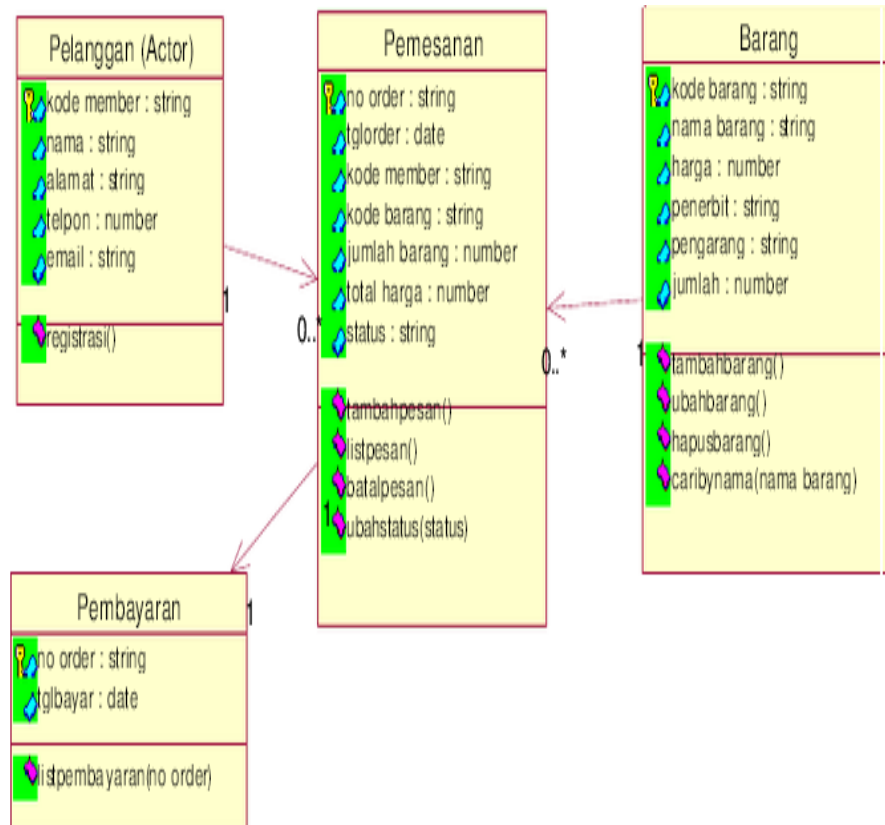


3. Kemudian ubah tipe relasi antar class dengan cara **klik** relasinya, **klik kanan** pada relasi yang dekat dengan class **Pelanggan**

4. Pilih **Multiplicity** $\Rightarrow$ **1** Dan **klik** $\Rightarrow$ **kanan** relasi yang dekat dengan class **Pemesanan** pilih **Multiplicity** $\Rightarrow$ **Zero or More**.



5. Dan buatlah relasi antar class-class yang lain seperti dibawah ini:

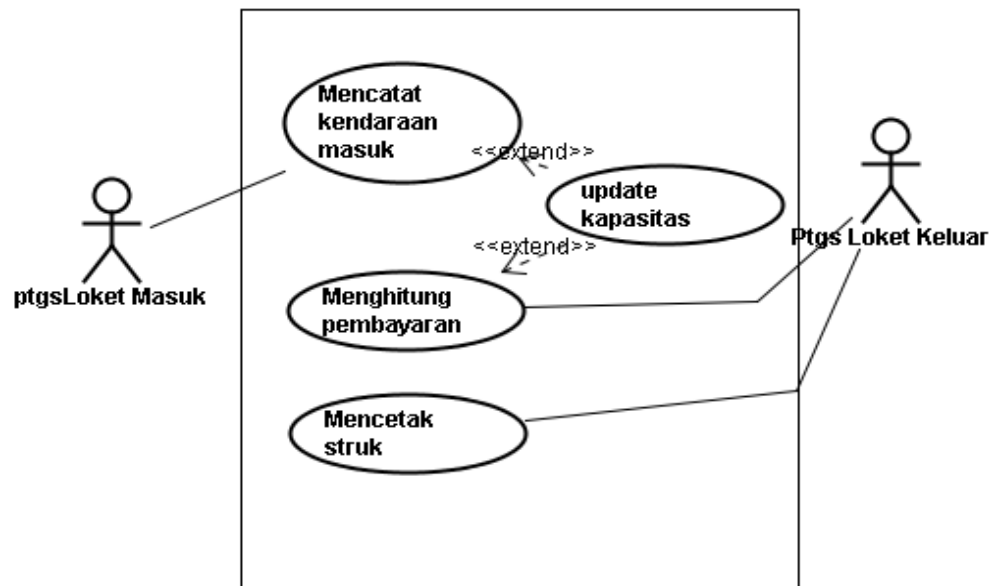


6. Jangan lupa simpan hasil praktikum ini ke dalam file yang lama.



Latihan

1. Buatlah diagram kelas untuk sistem parkir berdasarkan diagram use case berikut ini :



3. Sistem penjualan obat di Apotek dapat dioperasikan oleh kasir dan petugas gudang yang berhak. Persyaratan atau kebutuhan sistem tersebut diantaranya adalah harus dapat :

1. Memvalidasi pemakai sistem yang berhak
2. Mencatat data obat dan stok yang dijual
3. Mencatat data obat yang terjual baik melalui resep atau tidak
4. Mencatat identitas pasien untuk penjualan obat dengan resep
5. Memperbaharui data stok obat
6. Menampilkan informasi total pembayaran

7. Mencatat pembayaran dari pembeli
8. Mencetak struk pembayaran
9. Mencatat penerimaan obat dari supplier oleh petugas gudang
10. Mencetak laporan penjualan dan laporan persediaan obat

LEMBAR KERJA PRAKTIKUM

Paraf Dosen/Asisten

Tanggal :

