

PRAKTIKUM 9

Materi:

Object-Oriented Data Modeling

Tujuan Praktikum :

Mahasiswa mampu membuat model basis data berbasis objek sederhana

Referensi :

Braham Silberschatz, Henry F. Korth, and S. Sudarshan. 2001. Database Systems Concepts (6th ed.). McGraw-Hill Higher Education.

ed.). McGraw-Hill Higher Education.
Jeffrey A. Hoffer, Ramesh Venkataraman, and Heikki Topi. 2012. Modern Database Management (11th ed.) Prentice Hall Press, Upper Saddle River, NJ, USA.

Penyajian:

Pemodelan data berorientasi objek dibuat berdasarkan pendekatan melihat segala sesuatu sebagai sebuah objek. Seperti halnya pada E-R model yang menggunakan pendekatan entitas, pada pendekatan berorientasi objek, objek dapat meng-enkapsulasi data dan perilaku dari objek. Pemodelan berorientasi objek tidak hanya bisa digunakan untuk memodelkan data tetapi juga memodelkan cara kerja sistem. Pendekatan berorientasi objek bagus digunakan untuk membuat sebuah sistem yang kompleks.

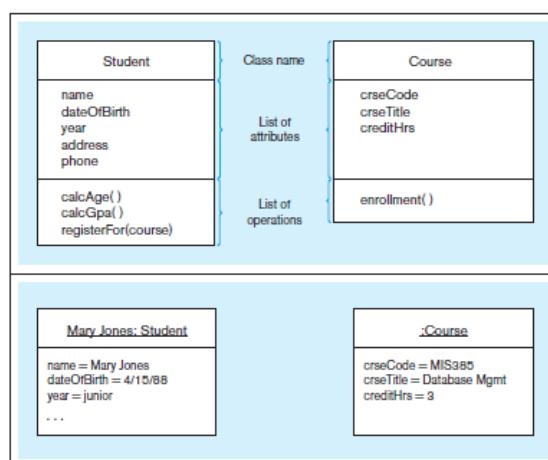
Class Diagram

Class diagram merupakan salah satu cara merepresentasikan model data berorientasi objek. Sebuah kelas merupakan konsep, abstraksi yang masuk akal. Sebuah kelas bisa merepresentasikan sesuatu yang terlihat seperti (orang, tempat, objek) atau sebuah konsep dan event (departemen, performa, pernikahan, pendaftaran) atau sebuah konsep dari proses desain (*user interface, controller, scheduler*). Kelas bisa dikatakan sebagai sebuah *blueprint* yang mendeskripsikan struktur sebuah objek.

Class diagram terdiri atas atribut dan *method*:

Atribut : mendefinisikan nilai yang dimiliki oleh *instance* kelas.

Method Mendefinisikan operasi atau apa saja yang bisa dilakukan oleh kelas tersebut.



Relationship

Hubungan antara satu kelas dengan kelas lain bisa dikategorikan sebagai berikut:

- *Association*



Association. Mahasiswa berkomunikasi dengan dosen

Asosiasi adalah jenis hubungan di mana satu objek berinteraksi dengan yang lain. Asosiasi merupakan hubungan struktural antarkelas.

- *Aggregation*



Aggregation. Departemen terdiri dari dosen

Agregasi adalah varian dari hubungan asosiasi. Hubungan asosiasi yang mewakili sebagian-seluruh atau sebagian dari hubungan. Agregasi mewakili "bagian dari" hubungan. Contoh: Perpustakaan dan mahasiswa, disini perpustakaan dapat tetap ada meskipun tanpa mahasiswa.

- *Composition*



Composition. Universitas terdiri dari departemen

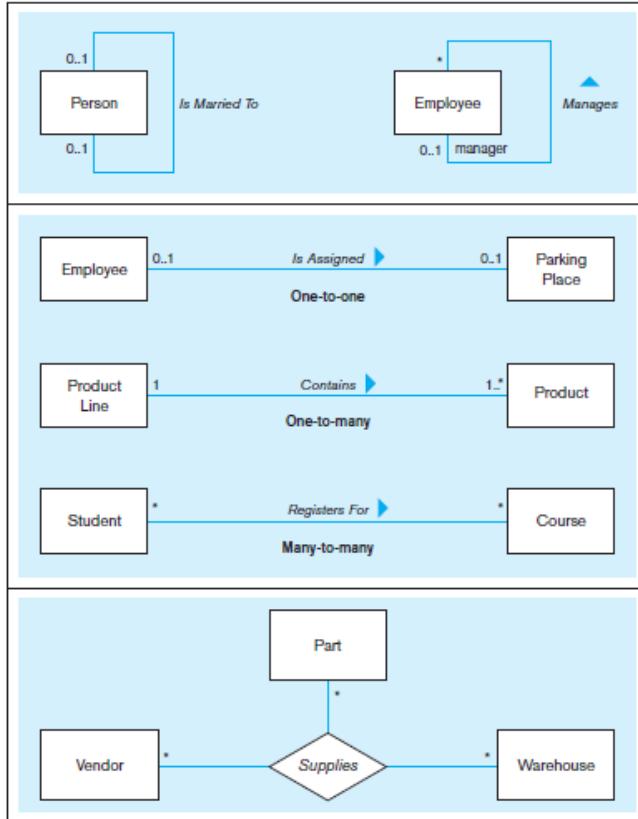
Jenis agregasi khusus di mana bila satu bagian maka kelas lain juga ikut hilang. Objek dari Class2 bergantung dengan Class1. Contoh: mesin merupakan bagian dari mobil.

- *Dependency*



Dependency. pegawai bergantung pada gaji

Hubungan antara elemen model dependen dan independen. Ada di antara dua kelas jika perubahan definisi dari satu dapat menyebabkan perubahan ke yang lain (tetapi tidak sebaliknya).



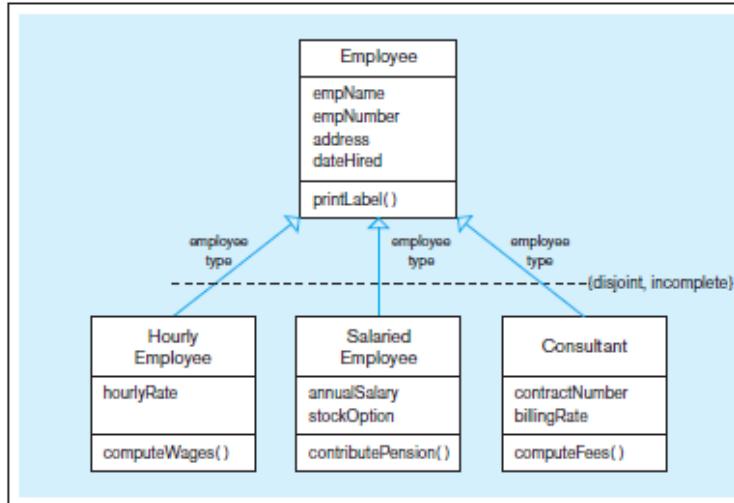
Multiplicity

Menunjukkan jumlah objek yang bisa berhubungan dengan objek lainnya.

0	Tidak ada objek
0..1	Tidak ada objek atau satu objek
1	Hanya tepat satu
1..1	Hanya tepat satu
0..*	Tidak ada atau lebih
*	Tidak ada atau lebih
1..*	Minimal satu atau lebih

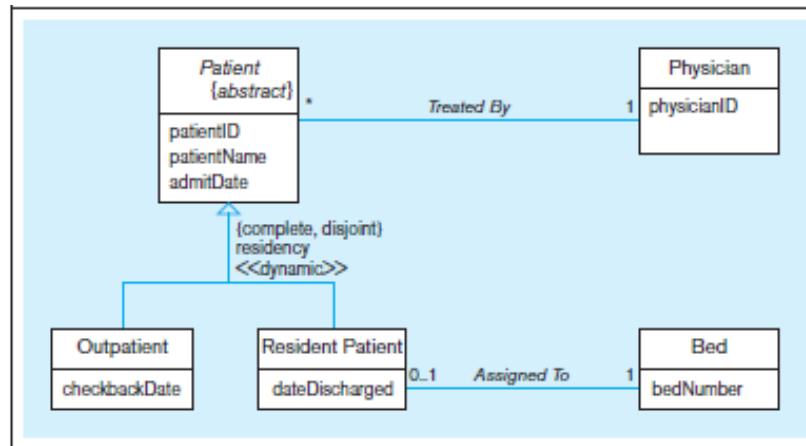
Generalisasi, Pewarisan, dan Overriding

Generalisasi menunjukkan bahwa salah satu dari dua kelas terkait (*subclass*) dianggap sebagai bentuk khusus dari yang lain (*superclass*) dan *superclass* dianggap sebagai Generalisasi dari *subclass*. Kelas *subclass* mewarisi atribut dan operasi dari kelas induk (*superclass*).

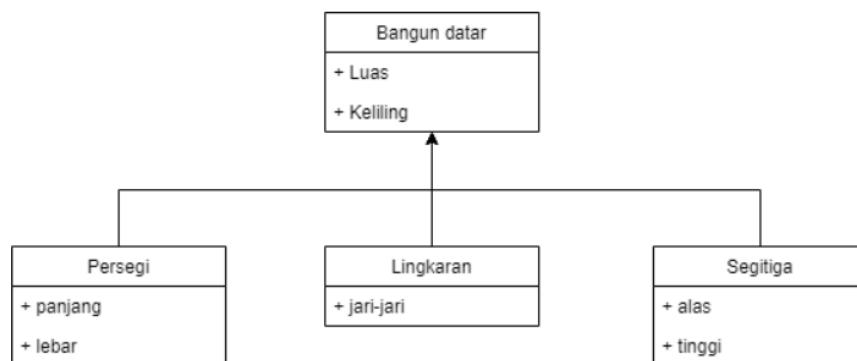


Terdapat dua tipe kelas yaitu :

- **Abstract Class:** Kelas abstrak adalah definisi template metode dan variabel kelas (kategori objek) yang berisi satu atau lebih metode yang diabstraksi. **Kelas yang tidak memiliki instance langsung tetapi turunannya mungkin memiliki instance langsung.**
Contoh:



Atau

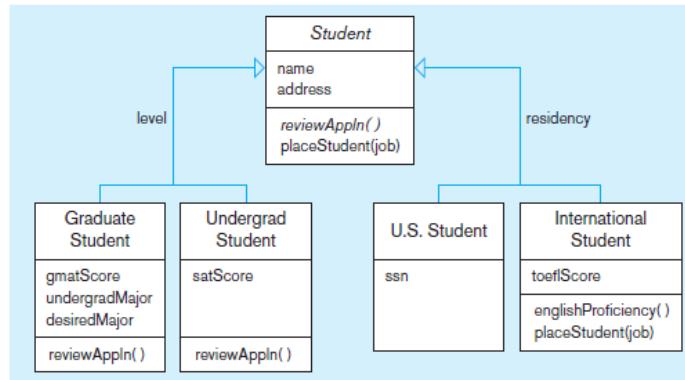


Kelas bangun datar merupakan *abstract class*

- **Concrete Class:** Kelas yang dapat memiliki *instance langsung*

Overriding merupakan proses penggantian metode yang diwarisi dari *superclass* dengan implementasi yang lebih spesifik metode itu dalam *subclass*.

Contoh :



Pada *method* `reviewAppln()` merupakan contoh *overriding*. Jika *method* `reviewAppln()` dipanggil maka akan menghasilkan hasil yang berbeda tergantung pada kelas yang digunakan.

LEMBAR KERJA PRAKTIKUM 9

Nama:	Tanggal Praktikum:
NRP :	Waktu Praktikum:
Nilai :	Nama Asisten :

Indonesia saat ini menurut pemeringkatan dari GHI (Global Hunger Index) menempati peringkat 73 dari 119 negara. Salah satu hal yang menyebabkan hal ini adalah karena masih banyak kasus kelaparan yang diderita oleh balita di wilayah Indonesia. Balita yang mengalami kelaparan ini begitu kekurangan gizi sehingga mempunyai tinggi badan yang di bawah rata-rata balita seusianya (*stunting*). Untuk mengatasi masalah ini dibutuhkan kerjasama dari seluruh lapisan masyarakat. Maka dari itu dibutuhkan sebuah sistem donasi untuk menyambungkan antara donatur dengan mereka yang membutuhkan (balita *stunting*).

Setiap pengguna atau *user* memiliki *username*, nama, *email*, id, dan *password*-nya masing-masing. Akan tetapi setiap *user* mempunyai perannya masing-masing seperti :

- **Donatur**

Donatur bisa memberikan donasi berupa uang yang bisa dikirimkan via transfer bank, menggunakan *virtual account* atau menggunakan metode pembayaran instan seperti *online banking*. Jika menggunakan cara transfer bank maka donatur harus meng-*upload* bukti transfer ke dalam sistem agar donasi bisa diverifikasi oleh admin. Donatur juga bisa melihat *history* dari donasi yang ia lakukan.

- **Donatur**

Selain uang, donasi yang bisa diberikan bisa berupa barang. Barang berupa bahan makanan bisa dikirimkan melalui kurir ekspedisi atau dengan menggunakan *drop point* yang disediakan oleh pemilik sistem. *User* dapat melihat di mana saja letak *drop point* terdekat dengan lokasi *user* dan *user* dapat menaruh barang yang ia ingin sumbangkan. Petugas pada *drop point* kemudian akan men-scan barcode yang dimiliki *user* sebagai penanda bahwa *user* telah menyumbangkan barang. *User* kemudian dapat melihat *history* donasi yang telah ia lakukan.

- **Campaigner**

User yang ingin membuat *campaign* disebut *campaigner*. Mereka harus meng-*upload* foto diri mereka dan foto KTP mereka untuk diverifikasi oleh admin. *Campaign* yang dibuat memiliki judul, foto dan video yang terkait juga deskripsi yang menjelaskan lebih tentang *campaign*. *Campaign* juga memiliki batasan waktu selesai serta target donasi yang dibutuhkan. *Campaign* juga dapat memiliki rekomendasi donasi barang yang kiranya dibutuhkan.

- **Admin**

Untuk setiap *user* yang ingin membuat *campaign* harus diverifikasi oleh admin. Admin juga harus memverifikasi *campaign* yang telah dibuat. Bila *campaign* dirasa mencurigakan *campaign* mendapat status *suspend* atau bisa saja ditolak.

Soal:

1. Tuliskan daftar objek/entitas yang berpotensi menjadi kelas beserta atributnya dan tipe data dari atributnya.
2. Buatlah *class diagram* berdasarkan soal di atas lengkap dengan *method*, atribut dan relasi antara masing-masing kelas.