

BAB 5

Class Diagram



5.1 Tujuan

1. Dapat memahami konsep class diagram
2. Dapat mengimplementasikan pembuatan class diagram dalam pengembangan pemrograman berbasis objek

5.2 Pengantar

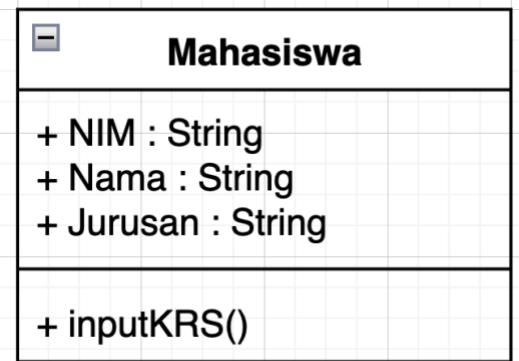
5.2.1 Pengertian Class Diagram

Class Diagram adalah salah satu jenis diagram dalam Unified Modeling Language (UML), sebuah bahasa standar untuk merancang dan mendokumentasikan sistem perangkat lunak. Class Diagram digunakan untuk memodelkan struktur sistem dengan menunjukkan class-class dalam sistem, atribut dan metode dalam class tersebut, serta hubungan antara class-class tersebut.

Class Diagram memberikan gambaran statis tentang sistem. Dalam kata lain, diagram ini menunjukkan apa yang ada di sistem dan bagaimana mereka saling terkait, tetapi tidak menunjukkan bagaimana sistem beroperasi dan bagaimana objek-objek dalam sistem berinteraksi satu sama lain. Untuk itu, kita menggunakan jenis diagram UML lainnya seperti sequence diagram atau activity diagram.

Class Diagram sangat penting dalam proses pengembangan perangkat lunak berorientasi objek. Diagram ini akan membantu perancang perangkat lunak untuk memvisualisasikan struktur kode yang mereka rancang sebelum benar-benar menulis kode tersebut. Hal ini akan memudahkan dalam membuat perubahan dan optimasi pada struktur tersebut sebelum melakukan pekerjaan yang lebih detail dan memakan waktu. Selain itu, Class Diagram juga berguna sebagai alat komunikasi antara berbagai stakeholder dalam proyek, seperti programmer, manajer proyek, dan klien.

Dalam Class Diagram, class biasanya digambarkan sebagai kotak yang dibagi menjadi tiga bagian: bagian atas berisi nama class, bagian tengah berisi atribut class, dan bagian bawah berisi metode class. Hubungan antara class digambarkan dengan berbagai jenis garis dan panah. Berikut ini contoh class diagram dengan nama class Mahasiswa, Atribut berisi NIM, nama dan Jurusan serta method dengan nama daftarKursus().



Gambar 6. 1 Class Diagram

5.2.2 Komponen Class Diagram

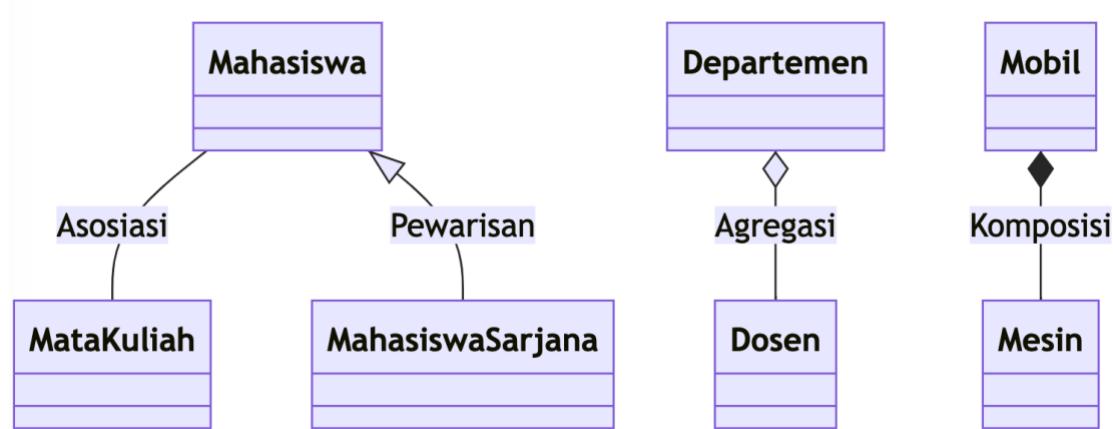
Class Diagram terdiri dari beberapa komponen utama yang membentuk struktur dasar diagram. Berikut adalah penjelasan tentang komponen-komponen tersebut:

- **Class :** Class adalah entitas utama dalam class diagram. Class digambarkan sebagai kotak dengan tiga bagian. Bagian atas berisi nama class, bagian tengah berisi atribut class, dan bagian bawah berisi metode class. Nama class biasanya ditulis dengan huruf kapital di awal, sementara atribut dan metode ditulis dengan huruf kecil di awal.
- **Atribut:** Atribut adalah variabel yang menyimpan nilai atau status objek. Atribut didefinisikan dalam bagian tengah kotak class. Setiap atribut biasanya memiliki tipe data, seperti int, string, atau boolean. Atribut biasanya ditulis dengan format: `+ namaAtribut : tipeData`.
- **Metode:** Metode adalah fungsi atau operasi yang dapat dilakukan oleh objek. Metode didefinisikan dalam bagian bawah kotak class. Setiap metode biasanya memiliki daftar parameter dan tipe pengembalian. Metode biasanya ditulis dengan format: `+ namaMetode(parameter : tipeData) : tipePengembalian`.

- Hubungan: Hubungan adalah tautan antara dua class yang menunjukkan bagaimana class tersebut berinteraksi satu sama lain. Hubungan dalam class diagram antara lain: Asosiasi, Agregasi, Komposisi dan Pewarisan

5.2.3 Jenis-Jenis Hubungan dalam Class Diagram

Dalam class diagram, hubungan antara class digambarkan menggunakan berbagai jenis garis dan simbol.



Gambar 6. 2 Jenis Hubungan dalam Class Diagram

Asosiasi: Asosiasi adalah hubungan dasar antara dua class. Asosiasi menunjukkan bahwa objek dalam satu class terkait dengan objek dalam class lain. Misalnya, class 'Mahasiswa' mungkin memiliki asosiasi dengan class 'MataKuliah', yang menunjukkan bahwa seorang mahasiswa dapat mengambil beberapa mata kuliah. Asosiasi biasanya digambarkan sebagai garis lurus antara dua class.

Agregasi: Agregasi adalah hubungan "memiliki" di mana class satu adalah bagian dari class lain, tetapi masih dapat ada secara independen. Misalnya, class 'Departemen' mungkin memiliki agregasi dengan class 'Dosen', yang menunjukkan bahwa departemen memiliki beberapa dosen, tetapi seorang dosen masih dapat ada tanpa

departemen. Agregasi biasanya digambarkan sebagai garis dengan simbol berlian di satu ujung.

Komposisi: Komposisi adalah hubungan "memiliki" yang lebih kuat di mana class satu tidak dapat ada tanpa class lain. Misalnya, class 'Mobil' mungkin memiliki komposisi dengan class 'Mesin', yang menunjukkan bahwa mobil memiliki mesin dan mesin tidak dapat ada tanpa mobil. Komposisi biasanya digambarkan sebagai garis dengan simbol berlian yang diisi di satu ujung.

Pewarisan (Inheritance): Pewarisan adalah hubungan "adalah jenis dari" di mana satu class adalah subclass dari class lain. Misalnya, class 'MahasiswaSarjana' mungkin mewarisi dari class 'Mahasiswa', yang menunjukkan bahwa MahasiswaSarjana adalah jenis khusus dari Mahasiswa. Pewarisan biasanya digambarkan sebagai garis dengan simbol panah berbentuk segitiga di satu ujung.

5.2.4 Kardinalitas

Kardinalitas dalam konteks pemrograman dan basis data merujuk pada jumlah elemen dalam satu set atau, lebih spesifik, jumlah hubungan antara entitas dalam model data. Kardinalitas digunakan untuk mendefinisikan hubungan antara entitas dalam model entitas-relasi (ER) dalam basis data. Ada beberapa jenis kardinalitas, termasuk:

1. **One-to-One (1:1):** Dalam hubungan ini, satu entitas dari set entitas A hanya dapat terkait dengan satu entitas dari set entitas B, dan sebaliknya. Misalnya, dalam sebuah perusahaan, setiap karyawan memiliki satu nomor identifikasi unik.
2. **One-to-Many (1:N) atau Many-to-One (N:1):** Dalam hubungan ini, satu entitas dari set entitas A dapat terkait dengan banyak entitas dari set entitas B. Namun, satu entitas dari set entitas B hanya dapat terkait dengan satu entitas dari set entitas A.

Misalnya, satu ibu bisa memiliki banyak anak, tetapi setiap anak hanya memiliki satu ibu.

3. **Many-to-Many (M:N):** Dalam hubungan ini, satu entitas dari set entitas A dapat terkait dengan banyak entitas dari set entitas B dan sebaliknya. Misalnya, dalam model basis data untuk sebuah toko buku, satu buku bisa dibeli oleh banyak pelanggan dan satu pelanggan bisa membeli banyak buku.

5.2.5 Langkah Membuat Class Diagram

Membuat class diagram melibatkan beberapa langkah penting. Berikut adalah langkah-langkah umum dalam proses pembuatan class diagram:

1. Identifikasi Class: Langkah pertama dalam membuat class diagram adalah mengidentifikasi class-class yang akan digambarkan dalam diagram. Class biasanya mewakili objek nyata atau konsep dalam sistem, seperti 'Mahasiswa', 'MataKuliah', atau 'Fakultas'.
2. Definisikan Atribut dan Metode: Setelah class diidentifikasi, langkah selanjutnya adalah mendefinisikan atribut dan metode untuk setiap class. Atribut adalah variabel yang menyimpan nilai atau status objek, sementara metode adalah fungsi atau operasi yang dapat dilakukan oleh objek.
3. Identifikasi Hubungan: Langkah selanjutnya adalah mengidentifikasi hubungan antara class. Hubungan bisa berupa asosiasi (hubungan dasar antara dua class), agregasi (hubungan "memiliki" di mana class satu adalah bagian dari class lain), komposisi (hubungan "memiliki" yang lebih kuat di mana class satu tidak dapat ada tanpa class lain), atau pewarisan (hubungan "adalah jenis dari" di mana satu class adalah subclass dari class lain).
4. Gambar Diagram: Setelah semua class, atribut, metode, dan hubungan diidentifikasi, langkah selanjutnya adalah menggambar diagram. Class

digambarkan sebagai kotak dengan tiga bagian, sementara hubungan digambarkan sebagai garis antara kotak.

5. Review dan Refine: Langkah terakhir adalah meninjau dan menyempurnakan diagram. Pastikan semua class, atribut, metode, dan hubungan digambarkan dengan benar dan jelas. Jika perlu, buat perubahan atau penyesuaian untuk meningkatkan kejelasan atau akurasi diagram.

5.2.6 Penerapan Class Diagram dalam Pemodelan Sistem

Class Diagram memiliki peran penting dalam pemodelan sistem, terutama dalam konteks pengembangan perangkat lunak berorientasi objek. Berikut adalah beberapa penerapan class diagram dalam pemodelan sistem:

- Perancangan Sistem: Class Diagram digunakan dalam tahap perancangan sistem untuk memvisualisasikan struktur dasar sistem yang akan dibangun. Dengan class diagram, pengembang dapat memahami class-class yang ada dalam sistem, atribut dan metode dalam class tersebut, serta hubungan antara class-class tersebut.
- Dokumentasi Sistem: Class Diagram juga berfungsi sebagai alat dokumentasi yang efektif. Dengan class diagram, struktur sistem dapat didokumentasikan dengan jelas dan sistematis. Ini memudahkan pengembang lain untuk memahami sistem, terutama saat melakukan pemeliharaan atau peningkatan sistem.
- Komunikasi antar Stakeholder: Class Diagram dapat digunakan sebagai alat komunikasi antara berbagai stakeholder dalam proyek, seperti programmer, manajer proyek, analis sistem, dan klien. Dengan class diagram, semua pihak dapat memiliki pemahaman yang sama tentang struktur sistem.
- Pengujian dan Validasi Sistem: Class Diagram juga dapat digunakan dalam tahap pengujian dan validasi sistem. Dengan membandingkan class diagram

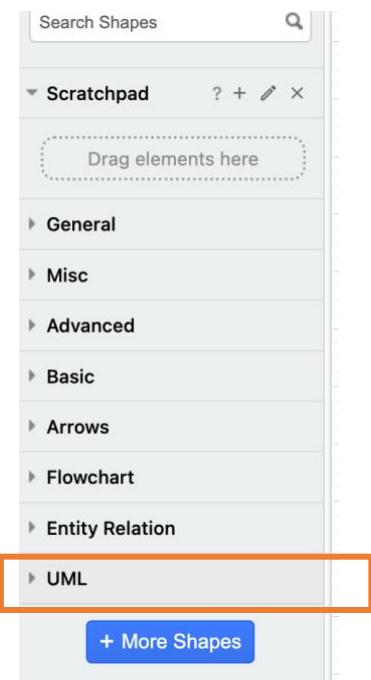
dengan implementasi sistem yang sebenarnya, pengembang dapat memastikan bahwa sistem telah dibangun sesuai dengan desain awal.

- Pendidikan dan Pelatihan: Dalam konteks pendidikan dan pelatihan, class diagram digunakan untuk mengajarkan konsep-konsep pemrograman berorientasi objek, seperti class, atribut, metode, dan pewarisan.

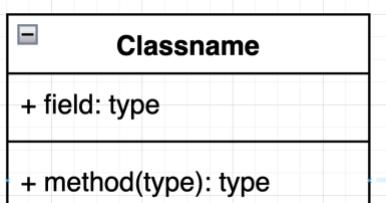
5.3 Kegiatan Praktikum

5.3.1 Kegiatan 1 : Membuat Class Diagram

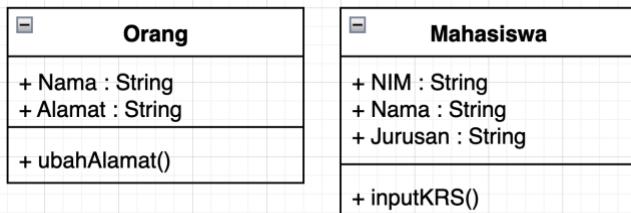
1. Kunjungi website <https://app.diagrams.net/> kemudian Pilih Create New Diagram
2. Pilih UML pada bagian Shape



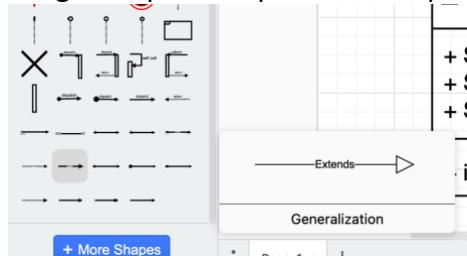
3. Pilih Class Diagram dengan cara drag n drop ke area kerja, Untuk melakukan perubahan klik 2x pada simbol class tersebut.



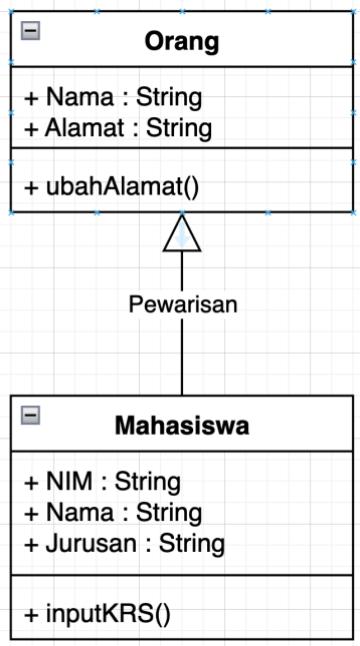
4. Buatlah 2 class seperti berikut



5. Drag n drop notasi pewarisan seperti berikut untuk 2 class tersebut

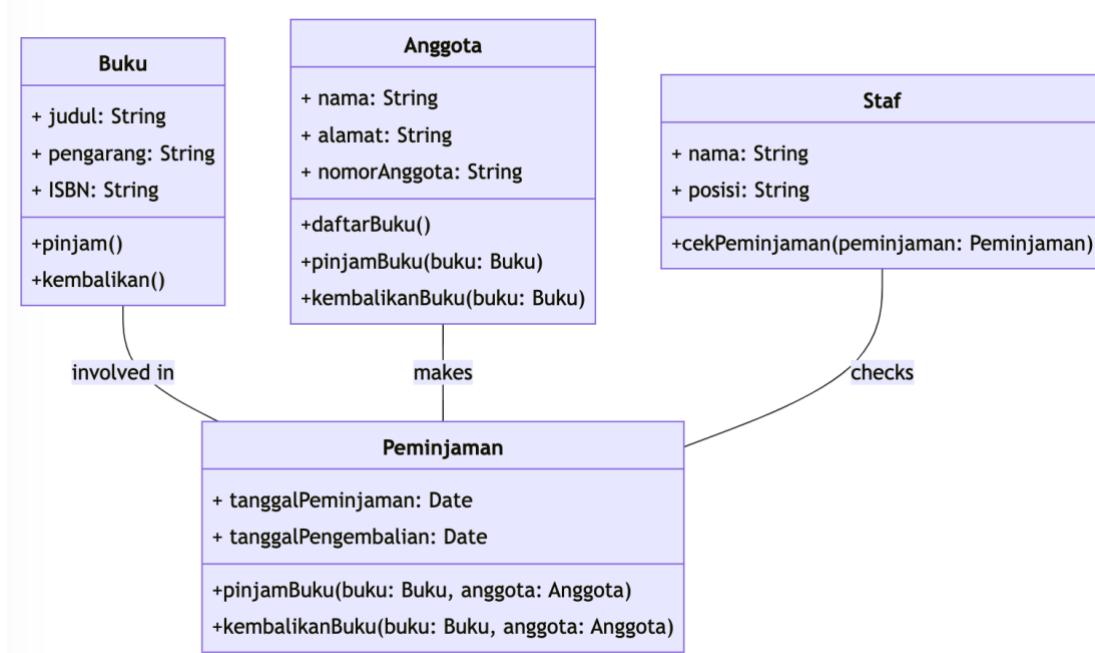


6. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan ini



5.3.2 Studi Kasus 1 : Sistem Perpustakaan

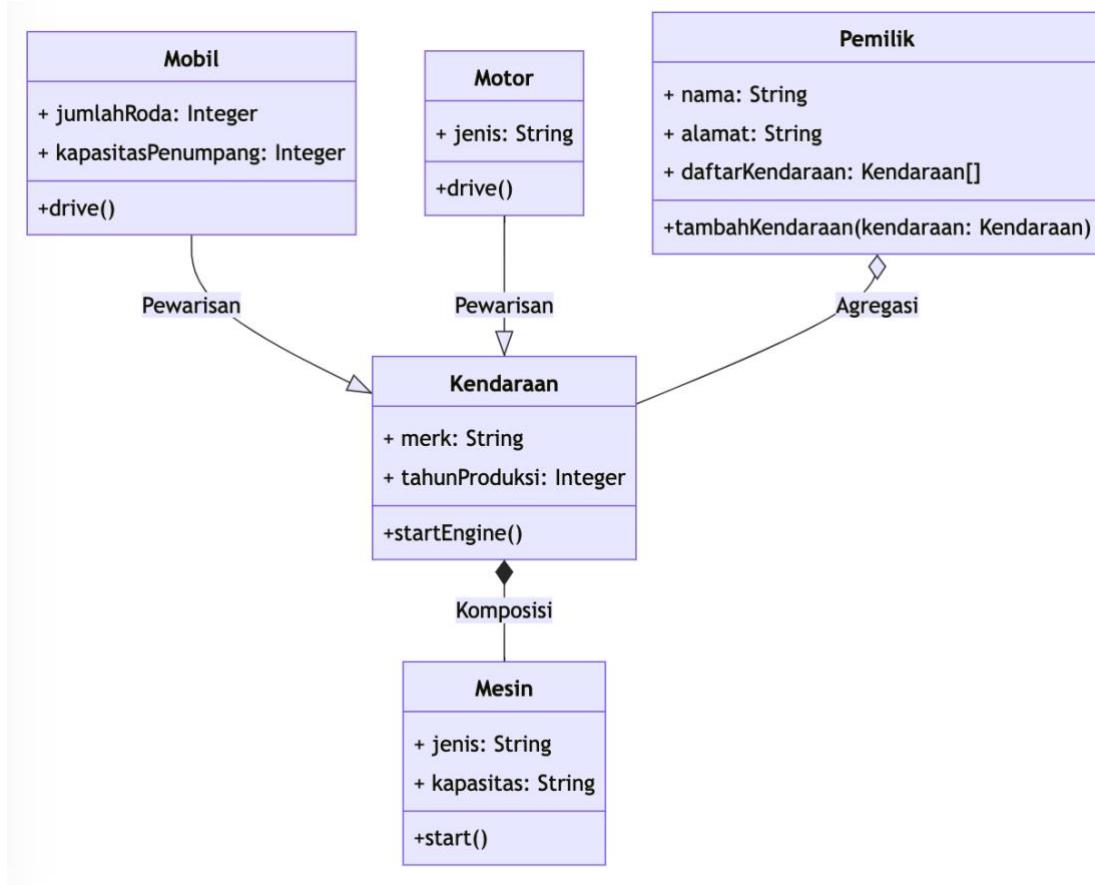
1. Buatlah Class Diagram seperti berikut ini



2. Buatlah analisa terhadap asosiasi yang terjadi pada setiap classnya

5.3.3 Studi Kasus 2 : Memahami Hubungan antar Class

1. Buatlah Class Diagram seperti berikut ini



2. Buatlah analisa terhadap asosiasi yang terjadi pada setiap classnya

5.4 Tugas

Sebuah perusahaan teknologi memiliki beberapa tim yang berbeda, termasuk tim pengembangan, tim pemasaran, dan tim dukungan pelanggan. Setiap tim memiliki sejumlah anggota dan setiap anggota memiliki peran tertentu dalam timnya. Selain itu, setiap anggota memiliki sejumlah keterampilan yang berbeda.

Buatlah class diagram yang mencakup kelas-kelas berikut:

- Perusahaan: Perusahaan memiliki nama dan alamat. Perusahaan juga memiliki beberapa tim.

- Tim: Setiap tim memiliki nama dan deskripsi. Tim juga memiliki beberapa anggota.
- Anggota: Setiap anggota memiliki nama dan email. Anggota juga memiliki peran dalam tim dan beberapa keterampilan.
- Peran: Setiap peran memiliki nama dan deskripsi.
- Keterampilan: Setiap keterampilan memiliki nama dan tingkat (misalnya, pemula, menengah, mahir).

Dalam diagram Anda, pastikan untuk menunjukkan hubungan berikut:

- **Asosiasi** antara Tim dan Anggota (setiap tim memiliki beberapa anggota dan setiap anggota adalah bagian dari satu tim).
- **Agregasi** antara Perusahaan dan Tim (setiap perusahaan memiliki beberapa tim, tetapi tim dapat ada tanpa perusahaan).
- **Komposisi** antara Anggota dan Peran (setiap anggota memiliki satu peran dan peran tidak dapat ada tanpa anggota).
- **Pewarisan** antara kelas baru "AnggotaPengembangan" dan "Anggota" (AnggotaPengembangan adalah jenis khusus dari Anggota).

Jelaskan setiap kelas, atribut, metode, dan hubungan dalam diagram Anda.