JOBSHEET II  
OBJECT

1. Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Mengenal objek dan class sebagai konsep mendasar pada pemrograman berorientasi objek
2. Mendeklarasikan class, atribut dan method
3. Membuat objek (instansiasi)
4. Mengakses atribut dan method dari suatu objek
5. Menerapkan konstruktor
6. Praktikum
7. Percobaan 1: Deklarasi Class, Atribut dan Method

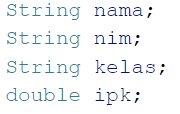
**Waktu Percobaan : 50 Menit**

Pada Percobaan 1 ini dilakukan pembuatan class beserta atribut dan method yang dimilikinya. Perhatikan Class Diagram berikut ini:

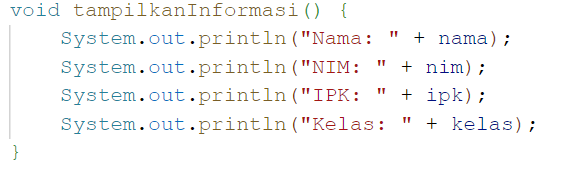
|  |
| --- |
| Mahasiswa |
| nim: String  nama: String  kelas: String  ipk: double |
| tampilkanInformasi(): void  ubahKelas(kelasBaru: String): void  updateIpk(ipkBaru: double): void  nilaiKinerja(ipk: double): String |

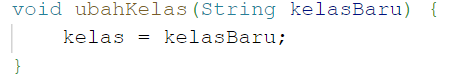
Berdasarkan class diagram tersebut, akan dibuat program menggunakan bahasa Java.

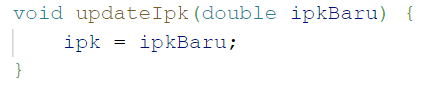
1. Langkah-langkah Percobaan
2. Buka text editor. Buat file baru, beri nama **Mahasiswa<NoAbsen>.java**
3. Lengkapi class **Mahasiswa** dengan atribut yang telah digambarkan di dalam class diagram tersebut

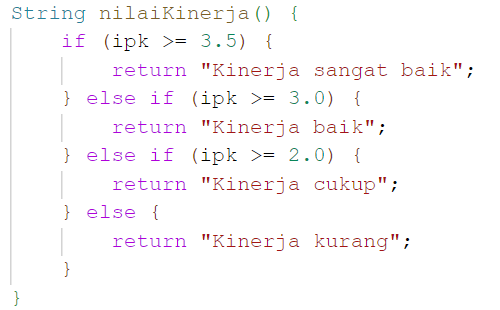


1. Lengkapi class **Mahasiswa** dengan method yang telah digambarkan di dalam class diagram tersebut.



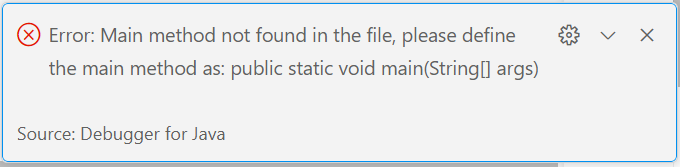






1. Compile dan run program.
2. Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.

****

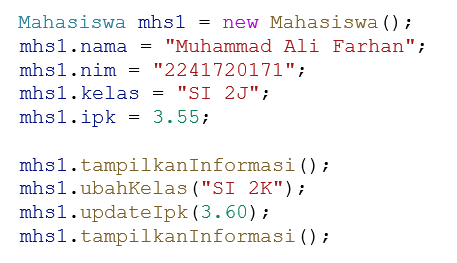
1. Pertanyaan
2. Sebutkan dua karakteristik class atau object!
3. Perhatikan class **Mahasiswa** pada Praktikum 1 tersebut, ada berapa atribut yang dimiliki oleh class Mahasiswa? Sebutkan apa saja atributnya!
4. Ada berapa method yang dimiliki oleh class tersebut? Sebutkan apa saja methodnya!
5. Perhatikan method **updateIpk()** yang terdapat di dalam class **Mahasiswa**. Modifikasi isi method tersebut sehingga IPK yang dimasukkan valid yaitu terlebih dahulu dilakukan pengecekan apakah IPK yang dimasukkan di dalam rentang 0.0 sampai dengan 4.0 (0.0 <= IPK <= 4.0). Jika IPK tidak pada rentang tersebut maka dikeluarkan pesan: "IPK tidak valid. Harus antara 0.0 dan 4.0".
6. Jelaskan bagaimana cara kerja method **nilaiKinerja()** dalam mengevaluasi kinerja mahasiswa, kriteria apa saja yang digunakan untuk menentukan nilai kinerja tersebut, dan apa yang dikembalikan (di-return-kan) oleh method nilaiKinerja() tersebut?
7. **Commit dan push kode program ke Github**
8. Percobaan 2: Instansiasi Object, serta Mengakses Atribut dan Method

**Waktu Percobaan: 50 Menit**

Sampai tahap ini, class **Mahasiswa** telah berhasil dibuat pada Percobaan 1. Selanjutnya, apabila class Mahasiswa tersebut ingin digunakan dan diakses atribut serta method-nya, maka perlu dibuat object/instance dari class **Mahasiswa** terlebih dahulu melalui proses instansiasi.

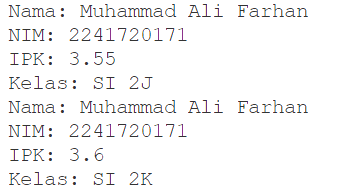
* + 1. Langkah-langkah Percobaan

1. Buat file baru, beri nama **MahasiswaMain<NoAbsen>.java**
2. Tuliskan struktur dasar bahasa pemrograman Java yang terdiri dari fungsi **main()**
3. Di dalam fungsi **main(),** lakukan instansiasi, kemudian lanjutkan dengan mengakses atribut dan method dari objek yang telah terbentuk.



1. Compile dan run program.
2. **Commit dan push kode program ke Github**
   * 1. Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.



* + 1. Pertanyaan

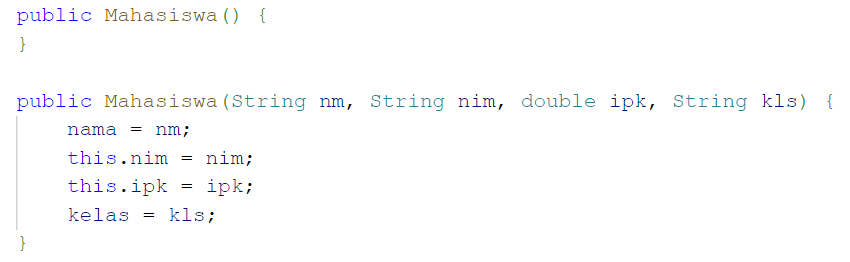
1. Pada class **MahasiswaMain**, tunjukkan baris kode program yang digunakan untuk proses instansiasi! Apa nama object yang dihasilkan?
2. Bagaimana cara mengakses atribut dan method dari suatu objek?
3. Mengapa hasil output pemanggilan method **tampilkanInformasi()** pertama dan kedua berbeda?
4. Percobaan 3: Membuat Konstruktor

**Waktu Percobaan: 60 Menit**

Pada percobaan ini, dilakukan pembuatan kode program untuk mengimplementasikan berbagai macam konstruktor berdasarkan parameternya.

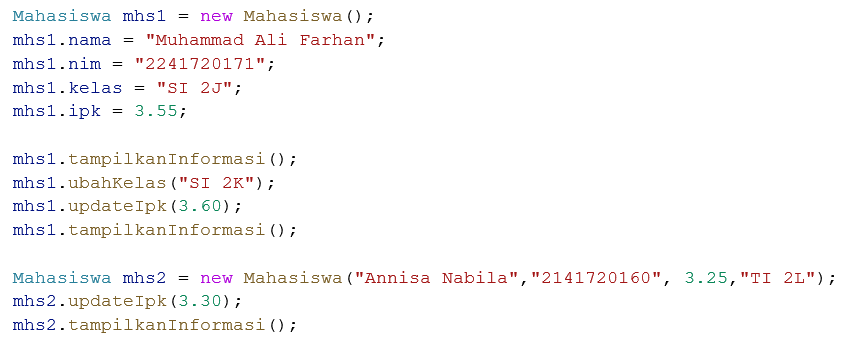
* + 1. Langkah-langkah Percobaan

1. Buka kembali class **Mahasiswa**. Tambahkan dua buah konstruktor di dalam class **Mahasiswa** tersebut, yang terdiri dari satu konstruktor default dan satu konstruktor berparameter. Konstruktor merupakan method istimewa, penempatan kode program untuk konstruktor dapat diperlakukan sama seperti method yang lain (setelah atribut).



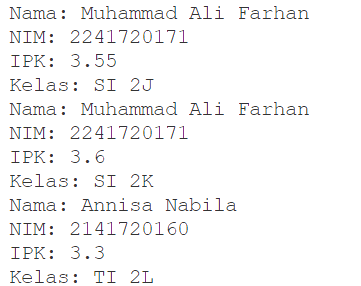
*Catatan: Apabila nama parameter sama dengan nama atribut, maka untuk merujuk pada variabel atribut ditambahkan sintaks* ***this*** *di depan nama* ***atribut***

1. Buka kembali class **MahasiswaMain**. Buat sebuah object lagi bernama **mhs2** dengan menggunakan konstruktor berparameter.



1. Compile dan run program.
2. **Commit dan push kode program ke Github**
   * 1. Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.



* + 1. Pertanyaan

1. Pada class **Mahasiswa** di Percobaan 3, tunjukkan baris kode program yang digunakan untuk mendeklarasikan konstruktor berparameter!
2. Perhatikan class **MahasiswaMain**. Apa sebenarnya yang dilakukan pada baris program berikut?



1. Hapus konstruktor default pada class **Mahasiswa**, kemudian compile dan run program. Bagaimana hasilnya? Jelaskan mengapa hasilnya demikian!
2. Setelah melakukan instansiasi object, apakah method di dalam class **Mahasiswa** harus diakses secara berurutan? Jelaskan alasannya!
3. Buat object baru dengan nama **mhs<NamaMahasiswa>** menggunakan konstruktor berparameter dari class **Mahasiswa**!
4. **Commit dan push kode program ke Github**
   1. Latihan Praktikum

**Waktu : 150 Menit**

1. Diberikan class diagram dari class MataKuliah sebagai berikut:

|  |
| --- |
| MataKuliah |
| kodeMK: String  nama: String  sks: int  jumlahJam: int |
| tampilInformasi(): void  ubahSKS(sksBaru: int): void  tambahJam(jam: int): void  kurangiJam(jam: int): void |

Buat program untuk mengimplementasikan class MataKuliah berdasarkan class diagram di atas, yang terdiri dari:

* Class MataKuliah (MataKuliah<NoAbsen>.java)
* Class MataKuliahMain (MataKuliahMain<NoAbsen>.java)

Pada class MataKuliahMain buatlah minimal 2 objek. Gunakan konstruktor default dan konstruktor berparameter saat mengintansiasi objek. Lalu panggil semua method yang sudah dibuat pada class MataKuliah.

Penjelasan dari atribut dan method pada class MataKuliah tersebut adalah sebagai berikut:

* 1. Atribut
* kodeMK (String): kode unik untuk mata kuliah.
* nama (String): nama lengkap dari mata kuliah
* sks (int): SKS (Satuan Kredit Semester)
* jumlahJam (int): jumlah total jam pertemuan per minggu untuk mata kuliah
  1. Method
* tampilInformasi(): method ini digunakan untuk menampilkan semua informasi yang berkaitan dengan mata kuliah.
* ubahSKS(int sksBaru): method ini memungkinkan pengubahan nilai SKS untuk mata kuliah. Setelah mengubah nilai, method ini memberi tahu pengguna bahwa SKS telah diubah.
* tambahJam(int jam): method ini menambahkan jumlah jam tambahan ke jumlah jam yang sudah ada untuk mata kuliah.
* kurangiJam(int jam): method ini berfungsi untuk mengurangi jumlah jam dari mata kuliah. Sebelum mengurangi, method ini melakukan pengecekan untuk memastikan bahwa jumlah jam yang tersisa cukup untuk dikurangi. Jika jumlah jam tidak mencukupi (jumlah jam awal lebih kecil dari jam pengurang), method ini akan memberi tahu pengguna bahwa pengurangan tidak dapat dilakukan. Jika pengurangan berhasil, method ini mengupdate jumlah jam dan mencetak nilai jumlah jam yang baru.

1. Diberikan class diagram dari class Dosen sebagai berikut:

|  |
| --- |
| Dosen |
| idDosen: String  nama: String  statusAktif: boolean  tahunBergabung: int  bidangKeahlian: String |
| tampilInformasi(): void  setStatusAktif(status: boolean): void  hitungMasaKerja(thnSkrg: int): int  ubahKeahlian(bidang: String): void |

Buat program untuk mengimplementasikan class Dosen berdasarkan class diagram di atas, yang terdiri dari:

* Class Dosen (Dosen<NoAbsen>.java)
* Class DosenMain (DosenMain<NoAbsen>.java)

Pada class DosenMain buatlah minimal 2 objek. Gunakan konstruktor default dan konstruktor berparameter saat mengintansiasi objek. Lalu panggil semua method yang sudah dibuat pada class Dosen

Penjelasan dari atribut dan method pada class Dosen tersebut adalah sebagai berikut:

* 1. Atribut
* idDosen (String): id unik untuk setiap dosen.
* nama (String): nama lengkap dari dosen.
* statusAktif (boolean): menunjukkan apakah dosen tersebut aktif (true) atau tidak aktif (false) dalam menjalankan tugasnya.
* tahunBergabung (int): tahun ketika dosen mulai bergabung dengan perguruan tinggi
* bidangKeahlian (String): bidang keahlian dosen, yang menjelaskan spesialisasi atau fokus akademik dosen
  1. Method
* tampilInformasi(): method ini digunakan untuk menampilkan informasi lengkap tentang dosen
* setStatusAktif(status: boolean): method ini digunakan untuk mengatur status aktif dosen. Jika parameter status diatur ke true, berarti dosen tersebut menjadi aktif. Sebaliknya, jika false, dosen tersebut dinyatakan tidak aktif.
* hitungMasaKerja(thnSkrg: int): method ini menghitung dan mengembalikan (me-return-kan) masa kerja dosen dalam tahun, berdasarkan tahun bergabung dan tahun saat ini (thnSkrg) yang menjadi parameter input method ini. Hasil perhitungan memberikan informasi tentang lamanya dosen bekerja di perguruan tinggi.
* ubahKeahlian(bidang: String): method ini digunakan untuk mengubah bidang keahlian dosen.