Laporan Hasil Pratikum

Algoritma Dan Struktur Data

Jobsheet 2



|  |  |
| --- | --- |
| Nama : | Zacky Rio Orlando |
| NIM : | 244107020086 |
| Kelas : | 1E |

Program Studi D-IV Teknik Informatika

Jurusan Teknologi Informasi

Praktikum

2025

**2. PRAKTIKUM**

**2.1 Percobaan 1 : Deklarasi Class, Atribut dan Method**

**2.1.1 Langkah-Langkah Percobaan**

Kode Program

public class Mahasiswa27 {

String nama;

String nim;

String kelas;

double ipk;

void tampilkanInformasi () {

System.out.println("Nama: " + nama);

System.out.println("NIM: " + nim);

System.out.println("IPK: " + ipk);

System.out.println("Kelas: " + kelas);

}

void ubahKelas (String kelasBaru) {

kelas = kelasBaru;

}

void updateIpk (double ipkBaru) {

ipk = ipkBaru;

}

String nilaiKinerja() {

if (ipk >= 3.5) {

return "Kinerja sangat baik";

} else if (ipk >= 3.0) {

return "Kinerja baik";

} else if (ipk >= 2.0) {

return "Kinerja cukup";

} else {

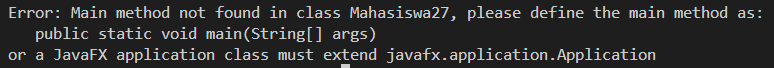
return "Kinerja kurang";

}

}

}

**2.1.2 Verifikasi Hasil Percobaan**

****

**2.1.3 Pertanyaan**

1. Sebutkan dua karakteristik class atau object!

* Memiliki atribut dan method

2. Perhatikan class Mahasiswa pada Praktikum 1 tersebut, ada berapa atribut yang dimiliki oleh class Mahasiswa? Sebutkan apa saja atributnya!

* Memiliki 4 atribut yaitu nama, nim, kelas, ipk

3. Ada berapa method yang dimiliki oleh class tersebut? Sebutkan apa saja methodnya!

* Memiliki 4 method yaitu ubahKelas, updateIpk, tampilkanInformasi, nilaiKinerja.

4. 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".

void updateIpk (double ipkBaru) {

ipk = ipkBaru;

while (true) {

if (ipk <= 0 || ipk >= 4) {

System.out.println("IPK tidak valid. Harus antara 0.0 dan 4.0");

continue;

} else {

break;

}

}

}

5. 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?

* Cara kerjanya yaitu dengan melihat dari method updateIpk yang telah diketahui nilai ipk, kemudian nilai dari ipk tersebut akan diproses dalam method nilaiKinerja dengan sesuai syarat menggunakan if-else if-else. Nilai yang direturnkan akan sesuai dengan hasil dari nilai ipk tersebut.

6. Commit dan push kode program ke Github.

**2.2 Percobaan 2 : Instansiasi Object, serta Mengakses Atribut dan Method**

**2.2.1 Langkah-Langkah Percobaan**

Kode Program

public class MahasiswaMain27 {

public static void main(String[] args) {

Mahasiswa27 mhs1 = new Mahasiswa27();

mhs1.nama = "Muhammad Ali Farhan";

mhs1.nim = "2241720171";

mhs1.kelas = "SI 2J";

mhs1.ipk = 3.55;

mhs1.tampilkanInformasi();

mhs1.ubahKelas("S1 2K");

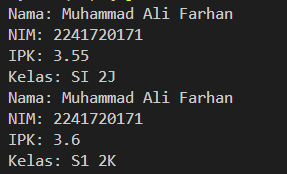
mhs1.updateIpk(3.60);

mhs1.tampilkanInformasi();

}

}

**2.2.2 Vertifikasi Hasil Percobaan**

****

**2.2.3 Pertanyaan**

1. Pada class MahasiswaMain, tunjukkan baris kode program yang digunakan untuk proses  
instansiasi! Apa nama object yang dihasilkan?

Mahasiswa27 mhs1 = new Mahasiswa27();

Pada line ke 3 dan nama object yang dihasilkan yaitu mhs1  
2. Bagaimana cara mengakses atribut dan method dari suatu objek?

* mhs1.nama = "Muhammad Ali Farhan";
* yaitu dengan contoh diatas namaObjek.namaAtribut = nilai yang diberikan

3. Mengapa hasil output pemanggilan method tampilkanInformasi() pertama dan kedua berbeda?

* Karena pada pemrograman yang diesekusi terlebih dahulu yaitu yang paling atas dan pada fungsi main terdapat 2 data. Data pertama yang dituliskan secara manual dan data yang kedua diisikan secara otomatis yaitu hanya memasukkan nilai saja dan kemudian memanggil method (fungsi).

**2.3 Percobaan 3 : Membuat Konstruktor**

**2.3.1 Langkah-Langkah Percobaan**

Class Mahasiswa27

public class Mahasiswa27 {

String nama;

String nim;

String kelas;

double ipk;

void tampilkanInformasi () {

System.out.println("Nama: " + nama);

System.out.println("NIM: " + nim);

System.out.println("IPK: " + ipk);

System.out.println("Kelas: " + kelas);

}

void ubahKelas (String kelasBaru) {

kelas = kelasBaru;

}

void updateIpk (double ipkBaru) {

ipk = ipkBaru;

while (true) {

if (ipk <= 0 || ipk >= 4) {

System.out.println("IPK tidak valid. Harus antara 0.0 dan 4.0");

continue;

} else {

break;

}

}

}

String nilaiKinerja() {

if (ipk >= 3.5) {

return "Kinerja sangat baik";

} else if (ipk >= 3.0) {

return "Kinerja baik";

} else if (ipk >= 2.0) {

return "Kinerja cukup";

} else {

return "Kinerja kurang";

}

}

public Mahasiswa27() {

}

public Mahasiswa27(String nm, String nim, double ipk, String kls) {

nama = nm;

this.nim = nim;

this.ipk = ipk;

kelas = kls;

}

}

Class MahasiswaMain27

public class MahasiswaMain27 {

public static void main(String[] args) {

Mahasiswa27 mhs1 = new Mahasiswa27();

mhs1.nama = "Muhammad Ali Farhan";

mhs1.nim = "2241720171";

mhs1.kelas = "SI 2J";

mhs1.ipk = 3.55;

mhs1.tampilkanInformasi();

mhs1.ubahKelas("S1 2K");

mhs1.updateIpk(3.60);

mhs1.tampilkanInformasi();

Mahasiswa27 mhs2 = new Mahasiswa27("Annisa Nabila", "2141720160", 3.25, "TI 2L");

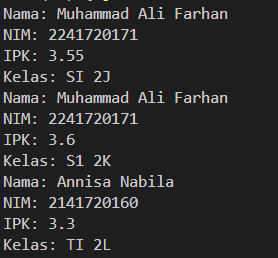
mhs2.updateIpk(3.30);

mhs2.tampilkanInformasi();

}

}

**2.3.2 Verifikasi Hasil Percobaan**

****

**2.3.3 Pertanyaan**

1. Pada class Mahasiswa di Percobaan 3, tunjukkan baris kode program yang digunakan untuk  
mendeklarasikan konstruktor berparameter!

* Konstruktor berparamater pada class Mahasiswa27 terdapat pada line 45

public Mahasiswa27(String nm, String nim, double ipk, String kls) {

nama = nm;

this.nim = nim;

this.ipk = ipk;

kelas = kls;

}

2. Perhatikan class MahasiswaMain. Apa sebenarnya yang dilakukan pada baris program  
berikut?

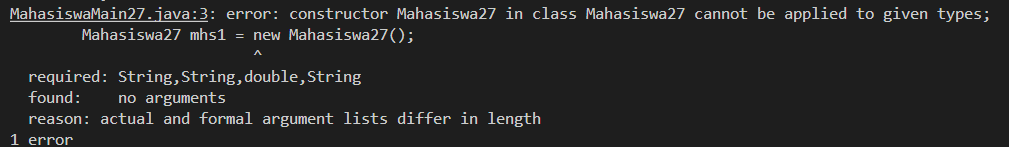


* Membuat objek baru dari class mahasiswa dengan proses instansiasi dengan memanggil konstruktor berparameter (String nm, String nim, double ipk, String kls) kemudian diberikan nilai dengan nm untuk Annisa Nabila, nim untuk 2141720160, ipk untuk 3.25, kls untuk TI 2L dan menyimpannya ke dalam objek mhs2.

3. Hapus konstruktor default pada class Mahasiswa, kemudian compile dan run program.  
Bagaimana hasilnya? Jelaskan mengapa hasilnya.

* Jika kontruktor default dihapus akan terjadi error

****

****

Terjadi error karena sudah ada konstruktor yang memiliki parameter, jadi java tidak menyediakannya secara otomatis. Jadi caranya yaitu menuliskan kostrukstor tersebut secara manual. Tapi jika ingin menghapus konstruktor yang memiliki parameter maka class MahasiswaMain27 akan mengalami error karena objek mhs2 tidak memiliki nilai.

4. Setelah melakukan instansiasi object, apakah method di dalam class Mahasiswa harus diakses  
secara berurutan? Jelaskan alasannya!

* Tidak, metode dalam class Mahasiswa **tidak harus diakses secara berurutan**, kecuali ada ketergantungan antar-method atau bisa disebut proses ada kelanjutannya. Method dapat dipanggil kapan saja sesuai kebutuhan, asalkan objek sudah dibuat dan tidak memerlukan hasil dari metode lain terlebih dahulu.

5. Buat object baru dengan nama mhs<NamaMahasiswa> menggunakan konstruktor  
berparameter dari class Mahasiswa!

public class MahasiswaMain27 {

public static void main(String[] args) {

Mahasiswa27 mhs1 = new Mahasiswa27();

mhs1.nama = "Muhammad Ali Farhan";

mhs1.nim = "2241720171";

mhs1.kelas = "SI 2J";

mhs1.ipk = 3.55;

mhs1.tampilkanInformasi();

mhs1.ubahKelas("S1 2K");

mhs1.updateIpk(3.60);

mhs1.tampilkanInformasi();

Mahasiswa27 mhs2 = new Mahasiswa27("Annisa Nabila", "2141720160", 3.25, "TI 2L");

mhs2.updateIpk(3.30);

mhs2.tampilkanInformasi();

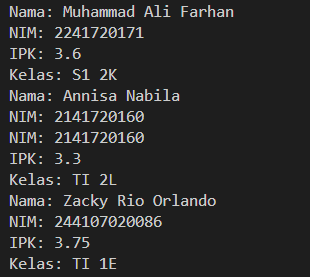
Mahasiswa27 mhs3 = new Mahasiswa27("Zacky Rio Orlando", "244107020086", 3.25, "TI 1E");

mhs3.updateIpk(3.75);

mhs3.tampilkanInformasi();

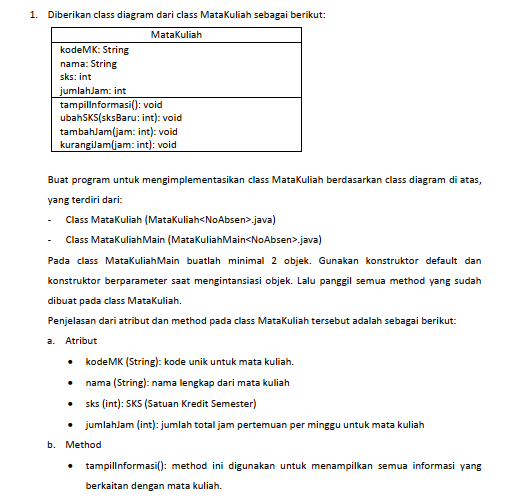
}

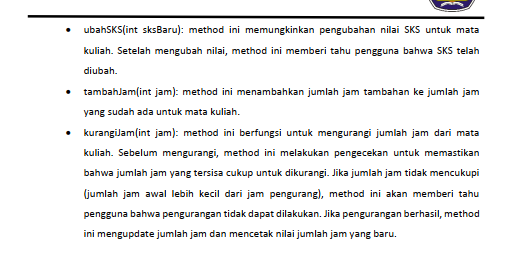
}

  
6. Commit dan push kode program ke Github

**2.4 Latihan Praktikum**

**1.**

****

****

**Class MataKuliah27**

public class MataKuliah27 {

String kodeMk;

String nama;

int sks;

int jumlahJam;

void tampilkanInformasi () {

System.out.println("Nama : " + nama);

System.out.println("Kode mata kuliah : " + kodeMk);

System.out.println("SKS : " + sks);

System.out.println("Jumlah Jam : " + jumlahJam);

}

void ubahSKS(int sksBaru) {

System.out.println("SKS diubah menjadi: " + sks + " SKS");

}

void tambahJam(int jam) {

this.jumlahJam += jam;

System.out.println("Total jam setelah ditambah : " + jumlahJam + " jam");

}

void kurangiJam(int jam) {

if (jam > this.jumlahJam) {

System.out.println("Pengurangan tidak dapat dilakukan");

} else {

this.jumlahJam -= jam;

System.out.println("Total jam setelah dikurangi: " + jumlahJam + " jam");

}

}

public MataKuliah27() {

}

public MataKuliah27(String nama, String kodeMK, int sks, int jumlahJam) {

this.nama = nama;

this.kodeMk = kodeMK;

this.sks = sks;

this.jumlahJam = jumlahJam;

}

}

**Class MataKuliahMain27**

public class MataKuliahMain27 {

public static void main(String[] args) {

MataKuliah27 mhs1 = new MataKuliah27();

System.out.println("---Data Mahasiswa 1---");

mhs1.nama = "Zacky";

mhs1.kodeMk = "BIG01";

mhs1.sks = 1;

mhs1.jumlahJam = 2;

mhs1.tampilkanInformasi();

mhs1.ubahSKS(3);

mhs1.tambahJam(2);

mhs1.kurangiJam(1);

System.out.println("Perubahan Data Informasi Mahasiswa");

mhs1.tampilkanInformasi();

MataKuliah27 mhs2 = new MataKuliah27("Rio Orlando", "ASD01", 3, 6);

System.out.println("---Data Mahasiswa 2---");

mhs2.tampilkanInformasi();

mhs2.ubahSKS(4);

mhs2.tambahJam(3);

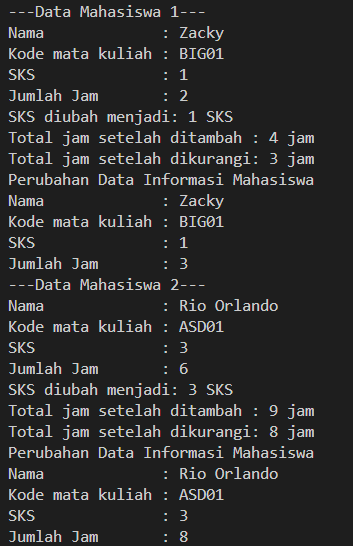
mhs2.kurangiJam(1);

System.out.println("Perubahan Data Informasi Mahasiswa");

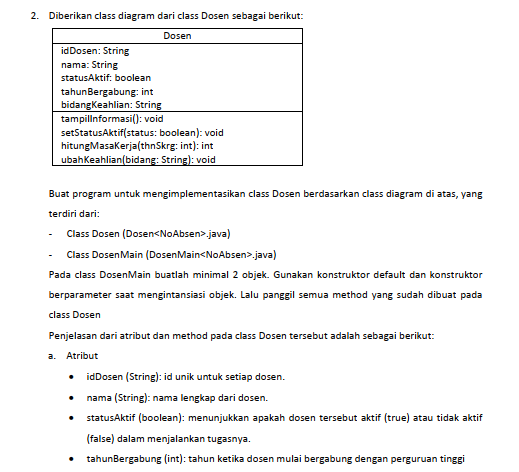
mhs2.tampilkanInformasi();

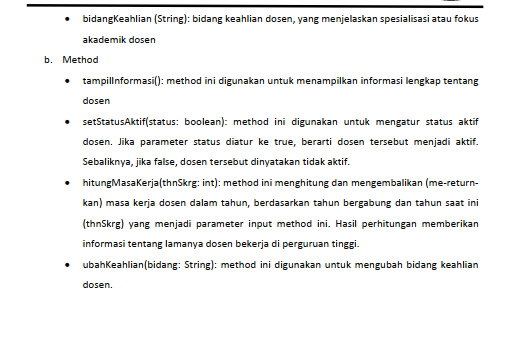
}

}

****

**2.**

****

****

**Class Dosen27**

public class Dosen27 {

String idDosen;

String nama;

boolean statusAktif;

int tahunBergabung;

String bidangKeahlian;

void tampilInformasi () {

System.out.println("Nama : " + nama);

System.out.println("ID Dosen : " + idDosen);

System.out.println("Status Aktif : " + (statusAktif ? "Aktif" : "Tidak Aktif"));

System.out.println("Tahun Bergabung : " + tahunBergabung);

System.out.println("Masa Kerja : " + hitungMasaKerja(2025) + " tahun");

System.out.println("Bidang Keahlian : " + bidangKeahlian);

}

void setStatusAktif(boolean status) {

this.statusAktif = status;

System.out.println("Status dosen sudah diubah menjadi: " + (statusAktif ? "Aktif" : "Tidak Aktif"));

}

int hitungMasaKerja(int thnSkrg) {

return thnSkrg - tahunBergabung;

}

void ubahKeahlian(String bidang) {

this.bidangKeahlian = bidang;

System.out.println("Bidang keahlian sudah diubah menjadi: " + bidangKeahlian);

}

public Dosen27(String idDosen, String nama, boolean statusAktif, int tahunBergabung, String bidangKeahlian) {

this.idDosen = idDosen;

this.nama = nama;

this.statusAktif = statusAktif;

this.tahunBergabung = tahunBergabung;

this.bidangKeahlian = bidangKeahlian;

}

public Dosen27() {

}

}

**Class DosenMain27**

public class DosenMain27 {

public static void main(String[] args) {

Dosen27 dsn1 = new Dosen27();

dsn1.nama = "Vivi Nur Wijayaningrum, S.kom., M.kom.";

dsn1.idDosen = "VNV";

dsn1.tahunBergabung = 2000;

System.out.println("Data Dosen 1:");

dsn1.bidangKeahlian = "Dasar Pemrograman";

dsn1.setStatusAktif(true);

dsn1.tampilInformasi();

dsn1.ubahKeahlian("Algoritma dan Struktur Data");

Dosen27 dsn2 = new Dosen27("ZRO", "Zacky Rio Orlando, S.kom., M.kom.", false, 2024, "Rekayasa Perangkat Lunak");

System.out.println("Data Dosen 2:");

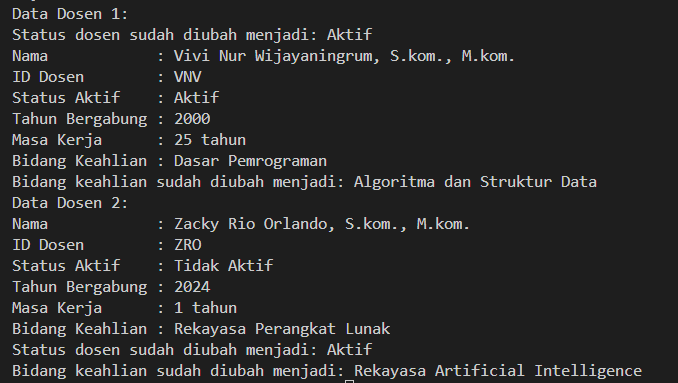
dsn2.tampilInformasi();

dsn2.setStatusAktif(true);

dsn2.ubahKeahlian("Rekayasa Artificial Intelligence");

}

}

****