LAPORAN HASIL PRAKTIKUM

BASIS DATA

JOBSHEET 2



RIFO ANGGI BARBARA DANUARTA

244107020063

TI\_1E

PROGRAM STUDI D\_IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

Percobaan 1

1. Buat folder baru bernama Jobsheet2 di dalam repository Praktikum ASD



1. Buka text editor. Buat file baru, beri nama Mahasiswa.java



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

String nama;

String nim;

String kelas;

String ipk;

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

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";

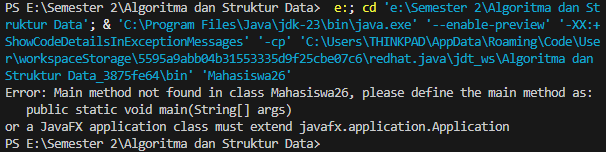
}

}

}

1. Compile dan run program.

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.



Pertanyaan

1. Sebutkan dua karakteristik class atau object!
2. Perhatikan class Mahasiswa pada Praktikum 1 tersebut, ada berapa atribut yang dimiliki oleh class Mahasiswa? Sebutkan apa saja atributnya!
3. Ada berapa method yang dimiliki oleh class tersebut? Sebutkan apa saja methodnya!
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".
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?
6. Commit dan push kode program ke Github

Jawaban

1. Atribut

Metode

1. Nama

Nim

Kelas

Ipk

1. 4, yang pertama tampilkanInformasi, yang kedua ubahKelas, yang ketiga updateIpk, yang ke empat nilaiKinerja.

void updateIpk(double ipkBaru) {

if (ipkBaru >= 0.0 && ipkBaru <= 4.0) {

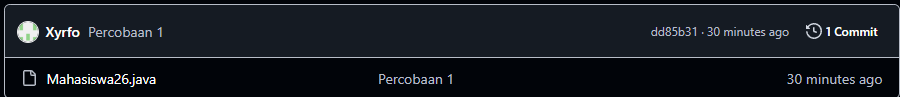
ipk = ipkBaru;

} else {

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

}

}

1. Yaitu ipk yang dikembalikan yaitu nilai ipk nya menuju method updateIpk
2. 

Percobaan 2

1. Buat file baru, beri nama MahasiswaMain.java



1. Tuliskan struktur dasar bahasa pemrograman Java yang terdiri dari fungsi main()

public static void main(String[] args) {

1. Di dalam fungsi main(), lakukan instansiasi, kemudian lanjutkan dengan mengakses atribut dan method dari objek yang telah terbentuk.

Mahasiswa26 mhs1 = new Mahasiswa26();

mhs1.nama = "Muhammad Ali Farhan";

mhs1.nim = "2241720171";

mhs1.kelas = "SI 2J";

mhs1.ipk = 3.55;

mhs1.tampilkanInformasi();

mhs1.ubahKelas("SI 2K");

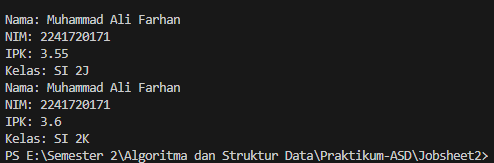
mhs1.updateIpk(3.60);

mhs1.tampilkanInformasi();

}

}

1. Compile dan run program.



1. Commit dan push kode program ke Github



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?

Jawaban

1. Yaitu baris ke tiga setelah public class dan public statis, yaitu mhs1
2. Untuk mengakses atribut yaitu namaObjek.namaAtribut Cuma menggunakan .

Untuk mengakses objek yaitu namaObjek.namaMethod(parameterJikaAda) cuma menggunakan . sama ()

1. Karena adanya perubahan data yang kedua sudah berubah karena adanya method yang merubah nilai didalamnya.

Percobaan 3

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).

public Mahasiswa26() {

}

public Mahasiswa26 (String nm, String nim, double ipk, String kls) {

nama = nm;

this.nim = nim;

this.ipk = ipk;

kelas = kls;

}

1. Buka kembali class MahasiswaMain. Buat sebuah object lagi bernama mhs2 dengan menggun

Mahasiswa26 mhs1 = new Mahasiswa26();

mhs1.nama = "Muhammad Ali Farhan";

mhs1.nim = "2241720171";

mhs1.kelas = "SI 2J";

mhs1.ipk = 3.55;

mhs1.tampilkanInformasi();

mhs1.ubahKelas("SI 2K");

mhs1.updateIpk(3.60);

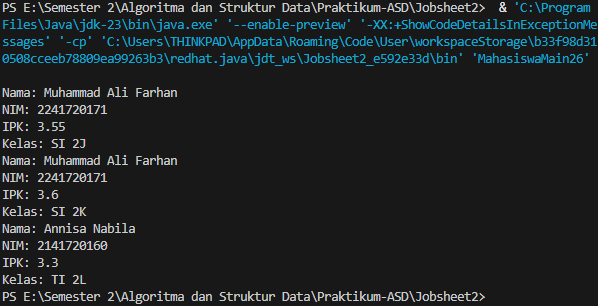
mhs1.tampilkanInformasi();

Mahasiswa26 mhs2 = new Mahasiswa26("Annisa Nabila","2141720160", 3.25, "TI 2L" );

mhs2.updateIpk(3.30);

mhs2.tampilkanInformasi();

1. Compile dan run program.



1. Commit dan push kode program ke Github



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 menggunakan konstruktor berparameter dari class Mahasiswa!
4. Commit dan push kode program ke Github

Jawaban

public Mahasiswa26 (String nm, String nim, double ipk, String kls) {

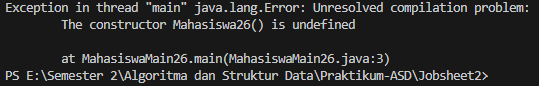
nama = nm;

this.nim = nim;

this.ipk = ipk;

kelas = kls;

}

1. Yaitu melakukan instansiasi objek karena ingin membuat objek baru dengan atribut yang berbeda.
2. 

Setelah constructor dihapus maka java tidak lagi menyediakan constructor default secara otomatis dan akan menyebabkan error.

1. Tidak, karena setiap method tidak ada ketergantungan antara method, method adalah fungsi endependen yang bisa dipanggil kapan aja dan tidak perlu berurutan pemanggilan method tergantung pada kebutuhan.

Mahasiswa26 mhsRifo = new Mahasiswa26("Rifo Danuarta", "244107020063", 3.28, "TI 1E" );

mhsRifo.updateIpk(3.36);

mhsRifo.tampilkanInformasi();

1. Commit dan push kode program ke Github



Latihan

1. Kode Program

public class MataKuliah26 {

String kodeMK;

String nama;

int sks;

int jumlahJam;

public MataKuliah26() {

this.kodeMK = "";

this.nama = "";

this.sks = 0;

this.jumlahJam = 0;

}

public MataKuliah26(String kodeMK, String nama, int sks, int jumlahJam) {

this.kodeMK = kodeMK;

this.nama = nama;

this.sks = sks;

this.jumlahJam = jumlahJam;

}

public void tampilInformasi() {

System.out.println("Kode Mata Kuliah: " + kodeMK);

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

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

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

}

public void ubahSKS(int sksBaru) {

this.sks = sksBaru;

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

}

public void tambahJam(int jam) {

this.jumlahJam += jam;

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

}

public void kurangiJam(int jam) {

if (jumlahJam >= jam) {

jumlahJam -= jam;

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

} else {

System.out.println("Pengurangan jam tidak dapat dilakukan, jumlah jam tidak mencukupi.");

}

}

}

Output

public class MatakuliahMain26 {

public static void main(String[] args) {

MataKuliah26 mk1 = new MataKuliah26();

mk1.ubahSKS(3);

mk1.tambahJam(4);

mk1.tampilInformasi();

MataKuliah26 mk2 = new MataKuliah26("MK002", "Pemrograman Lanjut", 4, 6);

mk2.tampilInformasi();

mk2.ubahSKS(5);

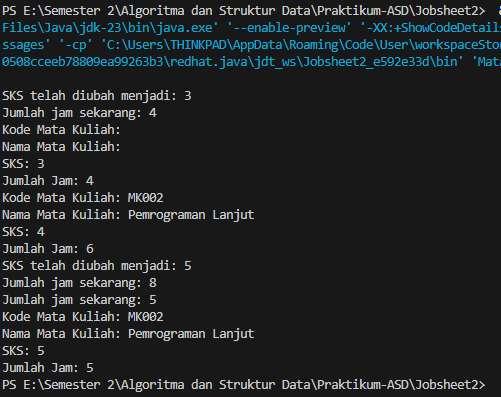
mk2.tambahJam(2);

mk2.kurangiJam(3);

mk2.tampilInformasi();

}

}



1. Kode Program

public class Dosen26 {

String idDosen;

String nama;

boolean statusAktif;

int tahunBergabung;

String bidangKeahlian;

public Dosen26() {

this.idDosen = "";

this.nama = "";

this.statusAktif = false;

this.tahunBergabung = 0;

this.bidangKeahlian = "";

}

public Dosen26(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 void tampilInformasi() {

System.out.println("ID Dosen26: " + idDosen);

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

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

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

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

}

public void setStatusAktif(boolean status) {

this.statusAktif = status;

System.out.println("Status aktif telah diubah menjadi: " + statusAktif);

}

public int hitungMasaKerja(int thnSkrg) {

return thnSkrg - tahunBergabung;

}

public void ubahKeahlian(String bidang) {

this.bidangKeahlian = bidang;

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

}

}

Output

public class DosenMain26 {

public static void main(String[] args) {

Dosen26 dosen1 = new Dosen26();

dosen1.setStatusAktif(true);

dosen1.ubahKeahlian("Matematika");

dosen1.tampilInformasi();

System.out.println("Masa kerja: " + dosen1.hitungMasaKerja(2025) + " tahun");

Dosen26 dosen2 = new Dosen26("D002", "Dr. Budi", true, 2010, "SIB");

dosen2.tampilInformasi();

dosen2.setStatusAktif(false);

dosen2.ubahKeahlian("K3");

System.out.println("Masa kerja: " + dosen2.hitungMasaKerja(2025) + " tahun");

dosen2.tampilInformasi();

}

}

