

Laporan Modul 4: Pengenalan Object Oriented Programming

Mata Kuliah: Praktikum Pemrograman Berorientasi Objek

Nama: Riski Al Fatah

NIM: 2024573010036

Kelas: TI.2E

Abstrak

Laporan praktikum ini (Modul 4) membahas pengenalan dasar dan konsep fundamental dari Object-Oriented Programming (OOP) menggunakan bahasa Java. Tujuan utama dari modul ini adalah untuk memahami perbedaan antara pemrograman prosedural dan OOP, serta mempraktikkan konsep inti OOP yaitu Class dan Object. Metodologi yang digunakan adalah melalui studi kasus praktikum pembuatan Sistem Manajemen Kartu Rencana Studi (KRS) sederhana. Sistem ini mengimplementasikan beberapa kelas, yaitu Mahasiswa (sebagai blueprint data mahasiswa), Matakuliah (sebagai blueprint data mata kuliah), dan KartuRencanaStudi (sebagai kelas yang mengelola interaksi antar objek). Laporan ini juga mencakup modifikasi program untuk menambah fungsionalitas yang lebih kompleks, seperti validasi batas SKS, fitur hapus mata kuliah, dan kemampuan untuk menampilkan nilai terbaik/terburuk. Hasil akhir dari praktikum ini adalah sebuah aplikasi konsol fungsional yang membuktikan bagaimana OOP dapat digunakan untuk membangun program yang lebih terstruktur, modular, dan mudah dikelola.

Pendahuluan

Modul ini dirancang untuk memberikan pemahaman dasar tentang pemrograman berorientasi objek menggunakan Java melalui praktikum langsung. Setiap praktikum dilengkapi dengan teori dasar dan langkah-langkah implementasi yang detail.

Teori Dasar

Objek-Oriented Programming (OOP) adalah paradigma pemrograman yang menggunakan “objek” - struktur data yang berisi data, dalam bentuk field, sering kali dikenal sebagai atribut; dan kode, dalam bentuk prosedur, sering kali dikenal sebagai metode.

Inti dari OOP adalah mendesain software dengan membagi masalah menjadi serangkaian objek yang saling berinteraksi. Ini bertentangan dengan pemrograman prosedural, yang fokus pada fungsi/prosedur dalam memproses data. OOP memungkinkan kita untuk membuat kode yang lebih modular, reusable, dan mudah di-maintain.

Konsep Dasar OOP:

- Class: Template atau blueprint untuk membuat object
- Object: Instance dari sebuah class
- Attribute/Field: Variabel yang dimiliki oleh class
- Method: Function yang dimiliki oleh class
- Constructor: Method khusus untuk inisialisasi object

Prinsip OOP:

- Encapsulation: Menyembunyikan detail implementasi
- Inheritance: Pewarisan sifat dari class parent
- Polymorphism: Kemampuan object untuk memiliki banyak bentuk
- Abstraction: Menyederhanakan kompleksitas dengan menyembunyikan detail

Class dan Objek

kelas adalah konsep abstrak yang mendefinisikan set atribut dan metode yang akan dimiliki oleh object. Kelas menyediakan struktur atau template yang menentukan bagaimana sebuah object harus dibuat. Kelas akan menentukan jenis atribut dan metode apa yang akan dimiliki oleh object, tetapi tidak menentukan nilai dari atribut itu sendiri untuk object tertentu.

Object adalah inti dari pemrograman berorientasi objek. Setiap object memiliki dua karakteristik utama, yaitu:

Object, dalam konteks pemrograman OOP, adalah sebuah entitas yang memiliki karakteristik dan perilaku. Kelas, di sisi lain, merupakan blueprint atau cetakan untuk membuat object. Kalau kamu bandingkan dengan dunia nyata, kamu bisa menganggap kelas sebagai rencana desain bangunan, sedangkan object adalah bangunan yang sesungguhnya telah dibangun berdasarkan desain tersebut.

Praktikum : Dasar Class dan Object

Contoh Program yang mengikuti modul:

1. Class Mahasiswa

```
package modul_4;

public class Mahasiswa { 4 usages new *

    // Atribut/Field (private untuk enkapsulasi)
    private String nama; 2 usages
    private String nim; 2 usages
    private String jurusan; 2 usages

    // Konstruktor dengan parameter
    public Mahasiswa(String nama, String npm, String jurusan, double ipk) { 1 usage new *
        this.nama = nama;
        this.nim = npm; // Catatan: Parameter 'npm' dimasukkan ke field 'nim'
        this.jurusan = jurusan;
    }

    // Metode Getter
    public String getNama() { new *
        return nama;
    }

    public String getNim() { 1 usage new *
        return nim;
    }

    public String getJurusan() { 1 usage new *
        return jurusan;
    }
}
```

2. Class Matakuliah

```
package modul_4;

public class Matakuliah { 8 usages  new *

    private String kode; 3 usages
    private String nama; 3 usages
    private int sks; 3 usages
    private double nilai; 12 usages

    // Constructor
    public Matakuliah(String kode, String nama, int sks) { 2 usages  new *
        this.kode = kode;
        this.nama = nama;
        this.sks = sks;
        this.nilai = 0.0;
    }

    // Getter dan Setter
    public String getKode() { 1 usage  new *
        return kode;
    }

    public String getNama() { new *
        return nama;
    }

    public int getSks() { 3 usages  new *
        return sks;
    }
}
```

```
public double getNilai() { new *  
    return nilai;  
}  
  
public void setNilai(double nilai) { new *  
    if (nilai >= 0.0 && nilai <= 100.0) {  
        this.nilai = nilai;  
    } else {  
        System.out.println("Nilai harus antara 0-100");  
    }  
}  
  
// Konversi nilai angka ke huruf  
public String getNilaiHuruf() { 2 usages new *  
    if (nilai >= 85) return "A";  
    else if (nilai >= 80) return "A-";  
    else if (nilai >= 75) return "B+";  
    else if (nilai >= 70) return "B";  
    else if (nilai >= 65) return "B-";  
    else if (nilai >= 60) return "C+";  
    else if (nilai >= 55) return "C";  
    else if (nilai >= 50) return "D";  
    else return "E";  
}
```

```
// Konversi nilai huruf ke bobot
public double getBobotNilai() { 1 usage new *
    String huruf = getNilaiHuruf();
    switch (huruf) {
        case "A": return 4.0;
        case "A-": return 3.7;
        case "B+": return 3.3;
        case "B": return 3.0;
        case "B-": return 2.7;
        case "C+": return 2.3;
        case "C": return 2.0;
        case "D": return 1.0;
        default: return 0.0;
    }
}

public void tampilkanInfo() { 1 usage new *
    System.out.printf("Kode: %s | Nama: %-30s | SKS: %d | Nilai: %.2f (%s)\n",
        kode, nama, sks, nilai, getNilaiHuruf());
}
}
```

3. Class KartuRencanaStudi

```
package modul_4;

public class KartuRencanaStudi { 2 usages new *

    private Mahasiswa mahasiswa; 4 usages
    private Matakuliah[] daftarMatakuliah; 7 usages
    private int jumlahMatkul; 10 usages
    private int maxMatkul; 3 usages

    public KartuRencanaStudi(Mahasiswa mahasiswa, int maxMatkul) { 2 usages new *
        this.mahasiswa = mahasiswa;
        this.maxMatkul = maxMatkul;
        this.daftarMatakuliah = new Matakuliah[maxMatkul];
        this.jumlahMatkul = 0;
    }

    // Menambah mata kuliah ke KRS
    public boolean tambahMatakuliah(Matakuliah matkul) { 1 usage new *
        if (jumlahMatkul < maxMatkul) {
            daftarMatakuliah[jumlahMatkul] = matkul;
            jumlahMatkul++;
            System.out.println("\nNote: Mata Kuliah \" + matkul.getNama() + \" \" Berhasil ditambahkan.");
            return true;
        } else {
            System.out.println("\nNote: KRS sudah penuh! Maksimal " + maxMatkul + " mata kuliah.");
            return false;
        }
    }
}
```

```
30 // Menghitung total SKS
31 public int hitungTotalSKS() { 1 usage new *
32     int totalSKS = 0;
33     for (int i = 0; i < jumlahMatkul; i++) {
34         totalSKS += daftarMatakuliah[i].getSks();
35     }
36     return totalSKS;
37 }
38
39 // Menghitung IPK
40 public double hitungIPK() { 1 usage new *
41     if (jumlahMatkul == 0) return 0.0;
42
43     double totalBobot = 0.0;
44     int totalSKS = 0;
45
46     for (int i = 0; i < jumlahMatkul; i++) {
47         Matakuliah mk = daftarMatakuliah[i];
48         totalBobot += mk.getBobotNilai() * mk.getSks();
49         totalSKS += mk.getSks();
50     }
51
52     return totalSKS > 0 ? totalBobot / totalSKS : 0.0;
53 }
54
```

```

56     public void tampilkanKRS() { 1usage new *
57         System.out.println("*****");
58         System.out.println("***** KARTU RENCANA STUDI (KRS) *****");
59         System.out.println("*****");
60         System.out.println("* Nama Mahasiswa      : " + mahasiswa.getNama());
61         System.out.println("* NPM                : " + mahasiswa.getNim());
62         System.out.println("* Jurusan           : " + mahasiswa.getJurusan());
63         System.out.println("-----");
64         System.out.println("* KODE | NAMA MATA KULIAH | SKS | NILAI(%) *");
65         System.out.println("-----");
66
67         if (jumlahMatkul == 0) {
68             System.out.println("| Belum ada mata kuliah yang diambil.");
69         } else {
70             for (int i = 0; i < jumlahMatkul; i++) {
71                 daftarMatakuliah[i].tampilkanInfo();
72             }
73         }
74
75         System.out.println("-----");
76         System.out.printf("Total SKS : %d | IPK : %.2f\n", hitungTotalSKS(), hitungIPK());
77         System.out.println("*****\n");
78     }
79

```

```

80     // Mencari mata kuliah berdasarkan kode
81     public Matakuliah getMatakuliahByKode(String kode) { 1usage new *
82         for (int i = 0; i < jumlahMatkul; i++) {
83             if (daftarMatakuliah[i].getKode().equalsIgnoreCase(kode)) {
84                 return daftarMatakuliah[i];
85             }
86         }
87         return null;
88     }
89 }

```


4. Class Main

```
package modul_4;

import java.util.Scanner;

public class Main { new *

    public static void main(String[] args) { new *
        // Objek Scanner untuk input
        Scanner input = new Scanner(System.in);

        System.out.println("*****");
        System.out.println("***** SISTEM KARTU RENCANA STUDI (KRS) *****");
        System.out.println("*****");

        // Input data mahasiswa
        System.out.println("// Input data mahasiswa");
        System.out.print("Masukkan Nama Mahasiswa: ");
        String nama = input.nextLine();

        System.out.print("NIM: ");
        String nim = input.nextLine();

        System.out.print("Jurusan: ");
        String jurusan = input.nextLine();

        // Membuat object Mahasiswa
        Mahasiswa mhs = new Mahasiswa(nama, nim, jurusan, ipk: 0.0);
    }
}
```

```

KartuRencanaStudi krs = new KartuRencanaStudi(mhs, maxMatkul: 10);

// Menu running
boolean running = true;

while (running) {
    System.out.println("*****");
    System.out.println("***** MENU KRS *****");
    System.out.println("*****");
    System.out.println("* 1. Tambah Mata Kuliah *");
    System.out.println("* 2. Input Nilai Mata Kuliah *");
    System.out.println("* 3. Tampilkan KRS *");
    System.out.println("* 4. Keluar *");
    System.out.println("*****");
    System.out.print("Pilih menu: ");

    // Menggunakan nextLine() lalu parsing ke int untuk menghindari masalah newline
    String pilihanStr = input.nextLine();
    int pilihan;
    try {
        pilihan = Integer.parseInt(pilihanStr);
    } catch (NumberFormatException e) {
        pilihan = -1; // Pilihan tidak valid
    }
}

```

```

switch (pilihan) {
    case 1:
        // Tambah mata kuliah
        System.out.println("\n*****");
        System.out.println("***** TAMBAH MATA KULIAH *****");
        System.out.println("-----");
        System.out.print("Masukkan Kode Mata Kuliah: ");
        String kode = input.nextLine();

        System.out.print("Nama Mata Kuliah: ");
        String namaMK = input.nextLine();

        System.out.print("Jumlah SKS: ");
        // Menggunakan nextInt() lalu dilanjutkan nextLine() di akhir
        int sks = input.nextInt();
        input.nextLine(); // Membersihkan newline

        Matakuliah mk = new Matakuliah(kode, namaMK, sks);
        krs.tambahMatakuliah(mk);
        break;
}

```

```
case 2:
    // Input nilai
    System.out.println("\n*****");
    System.out.println("***** INPUT NILAI *****");
    System.out.println("-----");
    System.out.print("Masukkan Kode Mata Kuliah: ");
    String kodeCari = input.nextLine();

    Matakuliah mkCari = krs.getMatakuliahByKode(kodeCari);
    if (mkCari != null) {
        System.out.print("Input Nilai (0-100): ");
        // Menggunakan nextDouble() lalu dilanjutkan nextLine() di akhir
        double nilai = input.nextDouble();
        input.nextLine(); // Membersihkan newline

        mkCari.setNilai(nilai);
        System.out.println("Nilai berhasil diinput!");
    } else {
        System.out.println("Mata kuliah tidak ditemukan!");
    }
    break;
```

```

        case 3:
            // Tampilkan KRS
            krs.tampilkanKRS();
            break;

        case 4:
            // Keluar
            System.out.println("Terima kasih!");
            running = false;
            break;

        default:
            System.out.println("Pilihan tidak valid!");
    }
}

// Menutup scanner setelah loop selesai
input.close();
}
}

```

OutPut:

```
*****
***** SISTEM KARTU RENCANA STUDI (KRS) *****
*****
// Input data mahasiswa
Masukkan Nama Mahasiswa: Riski Al Fatah
NIM: 2024573010036
Jurusan: TIK
*****
***** MENU KRS *****
*****
* 1. Tambah Mata Kuliah *
* 2. Input Nilai Mata Kuliah *
* 3. Tampilkan KRS *
* 4. Keluar *
```

Pilih menu: 1

```
*****
***** TAMBAH MATA KULIAH *****
-----
```

Masukkan Kode Mata Kuliah: 03

Nama Mata Kuliah: Pemograman Berorientasi Objek

Jumlah SKS: 10

Note: Mata Kuliah "Pemograman Berorientasi Objek" Berhasil ditambahkan.

```
*****
***** MENU KRS *****
*****
* 1. Tambah Mata Kuliah *
* 2. Input Nilai Mata Kuliah *
* 3. Tampilkan KRS *
* 4. Keluar *
*****
Pilih menu: 2

*****
***** INPUT NILAI *****
-----
Masukkan Kode Mata Kuliah: 03
Input Nilai (0-100): 98
Nilai berhasil diinput!
```

```

*****
***** MENU KRS *****
*****
* 1. Tambah Mata Kuliah *
* 2. Input Nilai Mata Kuliah *
* 3. Tampilkan KRS *
* 4. Keluar *
*****
Pilih menu: 3
*****
***** KARTU RENCANA STUDI (KRS) *****
*****
* Nama Mahasiswa : Riski Al Fatah
* NPM : 2024573010036
* Jurusan : TIK
-----
* KODE | NAMA MATA KULIAH | SKS | NILAI(%) *
-----
Kode: 03 | Nama: Pemograman Berorientasi Objek | SKS: 10 | Nilai: 98,00 (A)
-----
* Total SKS : 10 | IPK : 4,00
*****

```

```

*****
***** MENU KRS *****
*****
* 1. Tambah Mata Kuliah *
* 2. Input Nilai Mata Kuliah *
* 3. Tampilkan KRS *
* 4. Keluar *
*****
Pilih menu: 4
Terima kasih!

```

Modifikasi Contoh Program

Untuk Class Mahasiswa masih sama yang berbeda hanya class KartuRencanaStudi.java dan Class Main.java

1. Untuk Class KartuRencanaStudi ini dimodifikasi untuk menambahkan fitur Hapus Mata Kuliah dan Tampilkan Nilai Terbaik/Terburuk, serta validasi SKS di metode tambahMatakuliah. Batas SKS Maksimal saya tetapkan 24 SKS. Contoh inputnya seperti ini:

```

package modul_4;

import java.util.Arrays;

public class KartuRencanaStudi { 2 usages new *

    private Mahasiswa mahasiswa; 4 usages
    private Matakuliah[] daftarMatakuliah; 14 usages
    private int jumlahMatkul; 16 usages
    private final int maxMatkul; 3 usages
    private final int MAX_SKS = 24; // Batas SKS maksimal yang ditambahkan 2 usages

    public KartuRencanaStudi(Mahasiswa mahasiswa, int maxMatkul) { 4 usages new *
        this.mahasiswa = mahasiswa;
        this.maxMatkul = maxMatkul;
        this.daftarMatakuliah = new Matakuliah[maxMatkul];
        this.jumlahMatkul = 0;
    }

    // ... Metode yang tidak dimodifikasi ...

    // Menambah mata kuliah ke KRS (MODIFIKASI: Tambah Validasi SKS)
    public boolean tambahMatakuliah(Matakuliah matkul) { 1 usage new *
        if (jumlahMatkul < maxMatkul) {
            int totalSKSBaru = hitungTotalSKS() + matkul.getSKS();

```

```

            if (totalSKSBaru > MAX_SKS) {
                System.out.println("\nNote: Gagal menambahkan! Total SKS akan melebihi batas " + MAX_SKS + " SKS.");
                return false;
            }

            daftarMatakuliah[jumlahMatkul] = matkul;
            jumlahMatkul++;
            System.out.println("\nNote: Mata Kuliah \" " + matkul.getNama() + "\" Berhasil ditambahkan. Total SKS sekarang: " + totalSKSBaru);
            return true;
        } else {
            System.out.println("\nNote: KRS sudah penuh! Maksimal " + maxMatkul + " mata kuliah.");
            return false;
        }
    }

    // Menghitung total SKS
    public int hitungTotalSKS() { 2 usages new *
        int totalSKS = 0;
        for (int i = 0; i < jumlahMatkul; i++) {
            totalSKS += daftarMatakuliah[i].getSKS();
        }
        return totalSKS;
    }
}

```



```

// Menghitung IPK
public double hitungIPK() { 1usage new *
    if (jumlahMatkul == 0) return 0.0;

    double totalBobot = 0.0;
    int totalSKS = 0;

    for (int i = 0; i < jumlahMatkul; i++) {
        Matakuliah mk = daftarMatakuliah[i];
        totalBobot += mk.getBobotNilai() * mk.getSks();
        totalSKS += mk.getSks();
    }

    return totalSKS > 0 ? totalBobot / totalSKS : 0.0;
}

// Mencari mata kuliah berdasarkan kode
public Matakuliah getMatakuliahByKode(String kode) { 1usage new *
    for (int i = 0; i < jumlahMatkul; i++) {
        if (daftarMatakuliah[i].getKode().equalsIgnoreCase(kode)) {
            return daftarMatakuliah[i];
        }
    }

    return null;
}

```

```

// FITUR BARU: Menghapus mata kuliah dari KRS
public boolean hapusMatakuliah(String kode) { 1usage new *
    for (int i = 0; i < jumlahMatkul; i++) {
        if (daftarMatakuliah[i].getKode().equalsIgnoreCase(kode)) {
            String namaMK = daftarMatakuliah[i].getNama();

            // Geser semua elemen setelah index i ke kiri
            for (int j = i; j < jumlahMatkul - 1; j++) {
                daftarMatakuliah[j] = daftarMatakuliah[j + 1];
            }

            // Set elemen terakhir ke null dan kurangi jumlahMatkul
            daftarMatakuliah[jumlahMatkul - 1] = null;
            jumlahMatkul--;

            System.out.println("\nNote: Mata Kuliah \"" + namaMK + "\" Berhasil dihapus.");
            return true;
        }
    }

    System.out.println("\nNote: Gagal menghapus. Mata Kuliah dengan kode " + kode + " tidak ditemukan.");
    return false;
}

```

```
// FITUR BARU: Menampilkan Mata Kuliah Terbaik
public void tampilkanTerbaikTerburuk(boolean terbaik) { 2 usages new *
    if (jumlahMatkul == 0) {
        System.out.println("\nTidak ada mata kuliah yang diambil.");
        return;
    }

    Matakuliah mkTerpilih = daftarMatakuliah[0];

    for (int i = 1; i < jumlahMatkul; i++) {
        Matakuliah currentMk = daftarMatakuliah[i];

        // Logika untuk Nilai Terbaik
        if (terbaik) {
            if (currentMk.getNilai() > mkTerpilih.getNilai()) {
                mkTerpilih = currentMk;
            }
            // Logika untuk Nilai Terburuk
        } else {
            if (currentMk.getNilai() < mkTerpilih.getNilai()) {
                mkTerpilih = currentMk;
            }
        }
    }
}
```

```
System.out.println("\n*****");
System.out.println("***** NILAI " + (terbaik ? "TERBAIK" : "TERBURUK") + " DITEMUKAN *****");
System.out.println("-----");
System.out.println("* KODE | NAMA MATA KULIAH | SKS | NILAI(%) *");
System.out.println("-----");
mkTerpilih.tampilkanInfo();
System.out.println("*****");
}

// Menampilkan KRS
public void tampilkanKRS() { 1 usage new *
    // ... (Kode sama seperti sebelumnya) ...
    System.out.println("*****");
    System.out.println("***** KARTU RENCANA STUDI (KRS) *****");
    System.out.println("*****");
    System.out.println("* Nama Mahasiswa : " + mahasiswa.getNama());
    System.out.println("* NIM : " + mahasiswa.getNim());
    System.out.println("* Jurusan : " + mahasiswa.getJurusan());
    System.out.println("-----");
    System.out.println("| KODE | NAMA MATA KULIAH | SKS | NILAI(%) |");
    System.out.println("-----");
}
```

```

        if (jumlahMatkul == 0) {
            System.out.println("| Belum ada mata kuliah yang diambil.");
        } else {
            for (int i = 0; i < jumlahMatkul; i++) {
                daftarMatakuliah[i].tampilkanInfo();
            }
        }

        System.out.println("-----");
        System.out.printf("* Total SKS : %d | IPK : %.2f\n", hitungTotalSKS(), hitungIPK());
        System.out.println("*****\n");
    }
}

```

2. Dan untuk Main.java seperti ini:

Untuk Class KartuRencanaStudi ini dimodifikasi untuk menambahkan fitur Hapus Mata Kuliah dan Tampilkan Nilai Terbaik/Terburuk, serta validasi SKS di metode tambahMatakuliah. Batas SKS Maksimal saya tetapkan 24 SKS.

Contoh inputnya seperti ini:

```

package modul_4;

import java.util.Scanner;

public class Main { new *

    public static void main(String[] args) { new *
        // Objek Scanner untuk input
        Scanner input = new Scanner(System.in);

        System.out.println("*****");
        System.out.println("***** SISTEM KARTU RENCANA STUDI (KRS) *****");
        System.out.println("*****");

        // Input data mahasiswa
        System.out.println("// Input data mahasiswa");
        System.out.print("Masukkan Nama Mahasiswa: ");
        String nama = input.nextLine();

        System.out.print("NIM: ");
        String nim = input.nextLine();

        System.out.print("Jurusan: ");
        String jurusan = input.nextLine();

        // Membuat object Mahasiswa
        Mahasiswa mhs = new Mahasiswa(nama, nim, jurusan, ipk: 0.0);
    }
}

```

```

KartuRencanaStudi krs = new KartuRencanaStudi(mhs, maxMatkul: 10);

// Menu running
boolean running = true;

while (running) {
    System.out.println("*****");
    System.out.println("***** MENU KRS *****");
    System.out.println("*****");
    System.out.println("* 1. Tambah Mata Kuliah *");
    System.out.println("* 2. Input Nilai Mata Kuliah *");
    System.out.println("* 3. Tampilkan KRS (Lengkap) *");
    System.out.println("* 4. Hapus Mata Kuliah (NEW) *");
    System.out.println("* 5. Tampilkan Nilai Terbaik/Terburuk (NEW) *");
    System.out.println("* 6. Keluar *");
    System.out.println("*****");
    System.out.print("Pilih menu: ");

    // Menggunakan nextLine() lalu parsing ke int untuk menghindari masalah newline
    String pilihanStr = input.nextLine();
    int pilihan;
    try {
        pilihan = Integer.parseInt(pilihanStr);
    } catch (NumberFormatException e) {
        pilihan = -1; // Pilihan tidak valid
    }
}

```

```

switch (pilihan) {
    case 1:
        // Tambah mata kuliah
        System.out.println("\n*****");
        System.out.println("***** TAMBAH MATA KULIAH *****");
        System.out.println("-----");
        System.out.print("Masukkan Kode Mata Kuliah: ");
        String kode = input.nextLine();

        System.out.print("Nama Mata Kuliah: ");
        String namaMK = input.nextLine();

        System.out.print("Jumlah SKS: ");
        int sks = -1;
        try {
            sks = Integer.parseInt(input.nextLine());
        } catch (NumberFormatException e) {
            System.out.println("Input SKS tidak valid.");
            break;
        }

        Matakuliah mk = new Matakuliah(kode, namaMK, sks);
        krs.tambahMatakuliah(mk);
        break;
}

```

```

case 2:
    // Input nilai
    System.out.println("\n*****");
    System.out.println("***** INPUT NILAI *****");
    System.out.println("-----");
    System.out.print("Masukkan Kode Mata Kuliah: ");
    String kodeCari = input.nextLine();

    Matakuliah mkCari = krs.getMatakuliahByKode(kodeCari);
    if (mkCari != null) {
        System.out.print("Input Nilai (0-100): ");
        double nilai = -1.0;
        try {
            nilai = Double.parseDouble(input.nextLine());
        } catch (NumberFormatException e) {
            System.out.println("Input Nilai tidak valid.");
            break;
        }

        mkCari.setNilai(nilai);
        System.out.println("Nilai berhasil diinput!");
    } else {
        System.out.println("Mata kuliah tidak ditemukan!");
    }
    break;

```

```

case 3:
    // Tampilkan KRS
    krs.tampilkanKRS();
    break;

case 4:
    // Hapus Mata Kuliah (NEW)
    System.out.println("\n*****");
    System.out.println("***** HAPUS MATA KULIAH *****");
    System.out.println("-----");
    System.out.print("Masukkan Kode Mata Kuliah yang akan dihapus: ");
    String kodeHapus = input.nextLine();
    krs.hapusMatakuliah(kodeHapus);
    break;

case 5:
    // Tampilkan Nilai Terbaik/Terburuk (NEW)
    System.out.println("\n*****");
    System.out.println("***** TAMPILKAN NILAI *****");
    System.out.println("-----");
    System.out.println("1. Nilai Terbaik");
    System.out.println("2. Nilai Terburuk");
    System.out.print("Pilih opsi: ");

```

```
String opsiStr = input.nextLine();
int opsi;
try {
    opsi = Integer.parseInt(opsiStr);
} catch (NumberFormatException e) {
    opsi = -1;
}

if (opsi == 1) {
    krs.tampilkanTerbaikTerburuk(true); // true untuk terbaik
} else if (opsi == 2) {
    krs.tampilkanTerbaikTerburuk(false); // false untuk terburuk
} else {
    System.out.println("Opsi tidak valid!");
}

break;
```

```
case 6:
    // Keluar
    System.out.println("Terima kasih!");
    running = false;
    break;

default:
    System.out.println("Pilihan tidak valid!");
```

```
}
```

```
}
```

```
}
```

```
// Menutup scanner setelah loop selesai
input.close();
```

```
}
```

```
}
```

Contoh

OutPut:

```
*****
***** SISTEM KARTU RENCANA STUDI (KRS) *****
*****
// Input data mahasiswa
Masukkan Nama Mahasiswa: riski al fatah
NIM: 2024573010036
Jurusan: tik
```

```

*****
***** MENU KRS *****
*****
* 1. Tambah Mata Kuliah *
* 2. Input Nilai Mata Kuliah *
* 3. Tampilkan KRS (Lengkap) *
* 4. Hapus Mata Kuliah (NEW) *
* 5. Tampilkan Nilai Terbaik/Terburuk (NEW) *
* 6. Keluar *
*****
Pilih menu: 1

*****
***** TAMBAH MATA KULIAH *****
-----
Masukkan Kode Mata Kuliah: 03
Nama Mata Kuliah: pemograman berorientasi objek
Jumlah SKS: 24

Note: Mata Kuliah "pemograman berorientasi objek" Berhasil ditambahkan. Total SKS sekarang: 24

```

```

*****
***** MENU KRS *****
*****
* 1. Tambah Mata Kuliah *
* 2. Input Nilai Mata Kuliah *
* 3. Tampilkan KRS (Lengkap) *
* 4. Hapus Mata Kuliah (NEW) *
* 5. Tampilkan Nilai Terbaik/Terburuk (NEW) *
* 6. Keluar *
*****
Pilih menu: 2

*****
***** INPUT NILAI *****
-----
Masukkan Kode Mata Kuliah: 03
Input Nilai (0-100): 98
Nilai berhasil diinput!
*****

```

```

*****
***** MENU KRS *****
*****
* 1. Tambah Mata Kuliah *
* 2. Input Nilai Mata Kuliah *
* 3. Tampilkan KRS (Lengkap) *
* 4. Hapus Mata Kuliah (NEW) *
* 5. Tampilkan Nilai Terbaik/Terburuk (NEW) *
* 6. Keluar *
*****
Pilih menu: 3
*****
***** KARTU RENCANA STUDI (KRS) *****
*****
* Nama Mahasiswa : riski al fatah
* NIM : 2024573010036
* Jurusan : tik
-----
| KODE | NAMA MATA KULIAH | SKS | NILAI(%) |
-----
Kode: 03 | Nama: pemograman berorientasi objek | SKS: 24 | Nilai: 98,00 (A)
-----
* Total SKS : 24 | IPK : 4,00
*****

```

***** MENU KRS

- * 1. Tambah Mata Kuliah *
- * 2. Input Nilai Mata Kuliah *
- * 3. Tampilkan KRS (Lengkap) *
- * 4. Hapus Mata Kuliah (NEW) *
- * 5. Tampilkan Nilai Terbaik/Terburuk (NEW) *
- * 6. Keluar *

Pilih menu: 4

***** HAPUS MATA KULIAH

Masukkan Kode Mata Kuliah yang akan dihapus: 03

Note: Mata Kuliah "pemograman berorientasi objek" Berhasil dihapus.

```
***** MENU KRS *****
*****
* 1. Tambah Mata Kuliah *
* 2. Input Nilai Mata Kuliah *
* 3. Tampilkan KRS (Lengkap) *
* 4. Hapus Mata Kuliah (NEW) *
* 5. Tampilkan Nilai Terbaik/Terburuk (NEW) *
* 6. Keluar *
*****
Pilih menu: 5
```

```
*****
***** TAMPILKAN NILAI *****
-----
1. Nilai Terbaik
2. Nilai Terburuk
Pilih opsi: 2

Tidak ada mata kuliah yang diambil.
```

```
*****
***** MENU KRS *****
*****
* 1. Tambah Mata Kuliah *
* 2. Input Nilai Mata Kuliah *
* 3. Tampilkan KRS (Lengkap) *
* 4. Hapus Mata Kuliah (NEW) *
* 5. Tampilkan Nilai Terbaik/Terburuk (NEW) *
* 6. Keluar *
*****
Pilih menu: 6
Terima kasih!
```

Analisis Program Kartu Rencana Studi (KRS)

Program ini merupakan aplikasi berbasis Java yang mengelola **Kartu Rencana Studi (KRS)** mahasiswa. Program memungkinkan pengguna untuk menambah mata kuliah, menginput nilai, menampilkan KRS, menghapus mata kuliah, dan melihat nilai terbaik atau terburuk dari mata kuliah yang diambil.

Fitur Utama:

1. Input Data Mahasiswa:

- Mengambil input nama mahasiswa, NIM, dan jurusan untuk membuat objek `Mahasiswa`.

2. Menu KRS:

- Program menampilkan menu dengan opsi untuk:
 - Menambah mata kuliah.
 - Menginput nilai mata kuliah.
 - Menampilkan KRS.
 - Menghapus mata kuliah.
 - Menampilkan nilai terbaik atau terburuk.
 - Keluar dari program.

3. Tambah Mata Kuliah:

- Mengambil input kode mata kuliah, nama mata kuliah, dan jumlah SKS, kemudian menambahkannya ke dalam KRS.

4. Input Nilai Mata Kuliah:

- Mencari mata kuliah berdasarkan kode, kemudian input nilai (0-100) untuk mata kuliah yang ditemukan.

5. Tampilkan KRS:

- Menampilkan daftar mata kuliah yang diambil mahasiswa beserta SKS dan nilai masing-masing mata kuliah.

6. Hapus Mata Kuliah:

- Menghapus mata kuliah berdasarkan kode mata kuliah yang diberikan.

7. Tampilkan Nilai Terbaik/Terburuk:

- Menampilkan mata kuliah dengan nilai terbaik atau terburuk, sesuai pilihan pengguna.

8. Keluar:

- Menghentikan eksekusi program.

Alur Program:

1. Program dimulai dengan meminta input data mahasiswa (nama, NIM, dan jurusan).
2. Program kemudian menampilkan menu untuk interaksi pengguna.
3. Pengguna dapat memilih berbagai opsi melalui menu:
 - Menambah mata kuliah dengan mengisi kode mata kuliah, nama, dan SKS.
 - Menginput nilai untuk mata kuliah yang sudah ada.
 - Menampilkan KRS yang berisi daftar mata kuliah, SKS, dan nilai.
 - Menghapus mata kuliah berdasarkan kode.
 - Melihat nilai terbaik atau terburuk.
4. Program akan terus berjalan hingga pengguna memilih untuk keluar.

Kesimpulan

Program Kartu Rencana Studi (KRS) ini berhasil mengimplementasikan konsep Object-Oriented Programming dengan baik, dengan fitur-fitur seperti menambah, menghapus mata kuliah, serta menginput nilai dan menampilkan KRS. Dengan penambahan validasi SKS dan fitur baru untuk menampilkan nilai terbaik atau terburuk, program ini menjadi lebih fungsional dan mudah digunakan. Modifikasi yang dilakukan pada Class KartuRencanaStudi dan Class Main menjadikan program lebih dinamis, memungkinkan pengguna untuk mengelola mata kuliah dengan lebih fleksibel. Pemrograman berorientasi objek membantu dalam pembuatan aplikasi yang lebih terstruktur dan modular, meningkatkan pemeliharaan dan pengembangan lebih lanjut.

Referensi

Oracle Corporation. (2024). The Java™ Tutorials: Object-Oriented Programming Concepts. Diakses pada 6 November 2025, dari <https://docs.oracle.com/javase/tutorial/java/concepts/> Oracle Corporation. (2024). The Java™ Tutorials: Classes and Objects. Diakses pada 6 November 2025, dari <https://docs.oracle.com/javase/tutorial/java/javaOO/index.html>