

Laporan Praktikum Algoritma & Struktur Data

Jobsheet 16 – Collections



Nama: Aqil Rahmat Alifiandi

NIM: 2341760099

Prodi: D-IV Sistem Informasi Bisnis

**JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2023/2024**

PERCOBAAN 1

1. Buatlah sebuah class ContohList yang main method berisi kode program seperti di bawah ini

```
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;

public class ContohList05 {
    Run | Debug
    public static void main(String[] args) {

        List l = new ArrayList();
        l.add(e:1);
        l.add(e:2);
        l.add(e:3);
        l.add(e:"Cireng");
        System.out.printf(format:"Elemen 0: %d total elemen: %d elemen terakhir: %s\n",
        l.get(index:0), l.size(), l.get(l.size() - 1));

        l.add(e:4);
        l.remove(index:0);
        System.out.printf(format:"Elemen 0: %d total elemen: %d elemen terakhir: %s\n",
        l.get(index:0), l.size(), l.get(l.size() - 1));
    }
}
```

2. Tambahkan kode program untuk menggunakan collection dengan aturan penulisan kode program seperti berikut.

```
List<String> names = new LinkedList<>();
names.add(e:"Noureen");
names.add(e:"Akhleema");
names.add(e:"Shannum");
names.add(e:"Uwais");
names.add(e:"Al-Qarni");

System.out.printf(format:"Elemen 0: %s total elemen: %s elemen terakhir: %s\n",
names.get(index:0), names.size(), names.get(names.size() - 1));

names.set(index:0, element:"My kid");

System.out.printf(format:"Elemen 0: %s total elemen: %s elemen terakhir: %s\n",
names.get(index:0), names.size(), names.get(names.size() - 1));

System.out.println("Names: " + names.toString());
}
```

3. Hasil Run:

```
Elemen 0: 1 total elemen: 4 elemen terakhir: Cireng
Elemen 0: 2 total elemen: 4 elemen terakhir: 4
Elemen 0: Noureen total elemen: 5 elemen terakhir: Al-Qarni
Elemen 0: My kid total elemen: 5 elemen terakhir: Al-Qarni
Names: [My kid, Akhleema, Shannum, Uwais, Al-Qarni]
```

PERTANYAAN

1. Perhatikan baris kode 25-36, mengapa semua jenis data bisa ditampung ke dalam sebuah ArrayList?

Jawaban:

Karena arrayList tersebut tidak terdapat kurung sudut yang menyimpan secara spesifik data yang akan disimpan didalamnya. sehingga arrayylist pada baris kode 25-36 diatas bersifat sintak umum yang dapat menyimpan data segala jenis tipe data

2. Modifikasi baris kode 25-36 seingga data yang ditampung hanya satu jenis atau spesifik tipe tertentu!

Jawaban:

Berikut adalah hasil modifikasi, terdapat eror

```
List <Integer> l = new ArrayList<>();
l.add(e:1);
l.add(e:2);
l.add(e:3);
l.add(e:"Cireng");
```

3. Ubah kode pada baris kode 38 menjadi seperti ini

```
LinkedList<String> names = new LinkedList<>();
```

Jawaban:

```
List<String> names = new LinkedList<>();
```

4. Tambahkan juga baris berikut ini, untuk memberikan perbedaan dari tampilan yang sebelumnya

```
names.push("Mei-mei");
System.out.printf("Elemen 0: %s total elemen: %s elemen terakhir: %s\n",
    names.getFirst(), names.size(), names.getLast());
System.out.println("Names: " + names.toString());
```

Jawaban:

```
((LinkedList<String>) names).push(e:"Mei-mei");
System.out.printf(format:"Elemen 0: %s total elemen : %s elemen terakhir : %s\n",
    ((LinkedList<String>) names).getFirst(), names.size(), ((LinkedList<String>) names).getLast());
System.out.println("Names : "+ names.toString());
```

Hasil Run:

```
Elemen 0: 1 total elemen: 4 elemen terakhir: Cireng
Elemen 0: 2 total elemen: 4 elemen terakhir: 4
Elemen 0: Nourdeen total elemen: 5 elemen terakhir: Al-Qarni
Elemen 0: My kid total elemen: 5 elemen terakhir: Al-Qarni
Names: [My kid, Akhleema, Shannum, Uwais, Al-Qarni]
Elemen 0: Mei-mei total elemen : 6 elemen terakhir : Al-Qarni
Names : [Mei-mei, My kid, Akhleema, Shannum, Uwais, Al-Qarni]
```

5. Dari penambahan kode tersebut, silakan dijalankan dan apakah yang dapat Anda jelaskan!

Jawaban: dapat membatasi tipe data yang akan dimasukkan dengan perintah dengan menggunakan class LinkedList kita dapat menggunakan method linked list seperti push(), getFirst(), getLast(), akan tetapi ketika menggunakan class List tidak dapat digunakan

PERCOBAAN 2

1. Buatlah class dengan nama LoopCollection serta tambahkan method main yang isinya adalah sebagai berikut.

```
public class LoopCollection05 {  
    Run | Debug  
    public static void main(String[] args) {  
        Stack<String> fruits = new Stack<>();  
        fruits.push(item:"Banana");  
        fruits.add(e:"Orange");  
        fruits.add(e:"Watermelon");  
        fruits.add(e:"Leci");  
        fruits.push(item:"Salak");  
  
        for(String fruit : fruits){  
            System.out.printf(format:"%s ", fruit);  
        }  
        System.out.println("\n"+ fruits.toString());  
  
        while(!fruits.empty()){  
            System.out.printf(format:"%s ", fruits.pop());  
        }  
    }  
}
```

2. Tambahkan potongan kode berikut ini dari yang sebelumnya agar proses menampilkan elemen pada sebuah stack bervariasi.

```
fruits.push(item:"Melon");  
fruits.push(item:"Strawberry");  
System.out.println(x:"");  
for(Iterator<String> it = fruits.iterator(); it.hasNext();){  
    String fruit = it.next();  
    System.out.printf(format:"%s ", fruit);  
}  
System.out.println(x:"");  
fruits.stream().forEach(e -> {  
    System.out.printf(format:"%s ",e);  
});  
System.out.println(x:"");  
for(int i=0; i<fruits.size(); i++){  
    System.out.printf(format:"%s ", fruits.get(i));  
}
```

3. Hasil Run:

```
Banana Orange Watermelon Leci Salak  
[Banana, Orange, Watermelon, Leci, Salak]  
Salak Leci Watermelon Orange Banana  
Melon Durian  
Melon Durian  
Melon Durian
```

PERTANYAAN

1. Apakah perbedaan fungsi push() dan add() pada objek fruits?

Jawaban:

Perbedaan antara fungsi push() dan add() pada objek fruits adalah:

push(item): Menambahkan satu elemen baru dengan nilai item ke akhir Stack fruits.

add(e): Menambahkan satu elemen baru dengan nilai e ke akhir LinkedList fruits.

Jadi, push() digunakan untuk menambahkan elemen baru ke Stack, sedangkan add() digunakan untuk menambahkan elemen baru ke LinkedList.

2. Silakan hilangkan baris 43 dan 44, apakah yang akan terjadi? Mengapa bisa demikian?

Jawaban:

```
Banana Orange Watermelon Leci Salak  
[Banana, Orange, Watermelon, Leci, Salak]  
Salak Leci Watermelon Orange Banana
```

Jika baris 43 dan 44 dihilangkan yang akan terjadi data melon, dan durian tidak muncul

3. Jelaskan fungsi dari baris 46-49?

Jawaban:

Fungsi dari kode tersebut adalah:

- for (Iterator<String> it = fruits.iterator(); it.hasNext();) - Ini merupakan loop for-each yang akan mengiterasi setiap elemen di dalam LinkedList fruits.
- String fruit = it.next(); - Pada setiap iterasi, variabel fruit akan menyimpan nilai elemen berikutnya dari LinkedList fruits.
- System.out.printf(format:"%s ", fruit); - Kode ini akan mencetak setiap elemen fruit yang diambil dari LinkedList, dengan format string "%s " (string diikuti spasi).

4. Silakan ganti baris kode 25, Stack menjadi List dan apakah yang terjadi? Mengapa bisa demikian?

Jawaban:

Terjadi eror pada bagian push, empty, pop karena tidak dapat dijalankan pada interface list, yang mana perintah yang diatas adalah fungsi dari interface stack

5. Ganti elemen terakhir dari objek fruits menjadi "Strawberry"!

Jawaban:

```
fruits.push(item:"Melon");  
fruits.push(item:"Strawberry");  
System.out.println(x:"");
```

6. Tambahkan 3 buah seperti "Mango", "guava", dan "avocado" kemudian dilakukan sorting!

Jawaban:

```
fruits.push(item:"Mango");  
fruits.push(item:"Guava");  
fruits.push(item:"Avocado");  
  
System.out.println(x:"");  
System.out.println(x:"\nData sesudah disorting ");  
Collections.sort(fruits);  
Iterator i= fruits.iterator();
```

PERCOBAAN 3

1. Buatlah sebuah class Mahasiswa dengan attribute, konstruktor, dan fungsi sebagai berikut

```
public class Mahasiswa05 {  
    String nim;  
    String nama;  
    String notelp;  
  
    public Mahasiswa05 () {  
    }  
    public Mahasiswa05(String nim, String nama, String notelp){  
        this.nim = nim;  
        this.nama = nama;  
        this.notelp = notelp;  
    }  
  
    @Override  
    public String toString(){  
        return "Mahasiswa{" + "nim= " + nim + ", nama=" + nama + ", notelp=" + notelp + '}';  
    }  
}
```

2. Selanjutnya, buatlah sebuah class ListMahasiswa yang memiliki attribute seperti di bawah ini

```
public class ListMahasiswa05 {  
    List<Mahasiswa05> mahasiswa = new ArrayList<>();  
}
```

3. Method tambah(), hapus(), update(), dan tampil() secara berurut dibuat agar bisa melakukan operasi-operasi seperti yang telah disebutkan

```
public void tambah(Mahasiswa05... mahasiswa) {  
    mahasiswa.addAll(Arrays.asList(mahasiswa));  
}  
  
public void hapus(int index) {  
    mahasiswa.remove(index);  
}  
  
public void update(int index, Mahasiswa05 mhs) {  
    mahasiswa.set(index, mhs);  
}  
  
public void tampil() {  
    mahasiswa.stream().forEach(mhs -> {  
        System.out.println(mhs.toString());  
    });  
}
```

4. Untuk proses hapus, update membutuhkan fungsi pencarian terlebih dahulu yang potongan kode programnya adalah sebagai berikut

```
int LinearSearch(String nim) {  
    for(int i=0; i<mahasiswas.size(); i++) {  
        if(nim.equals(mahasiswas.get(i).nim)) {  
            return i;  
        }  
    }  
    return -1;  
}
```

5. Pada class yang sama, tambahkan main method seperti potongan program berikut dan amati hasilnya!

```
public static void main(String[] args) {  
    ListMahasiswa05 lm = new ListMahasiswa05();  
    Mahasiswa05 m = new Mahasiswa05(nim:"201234", nama:"Noureen", notelp:"021xx1");  
    Mahasiswa05 m1 = new Mahasiswa05(nim:"201235", nama:"Akhleema", notelp:"021xx2");  
    Mahasiswa05 m2 = new Mahasiswa05(nim:"201236", nama:"Shannum", notelp:"021xx3");  
  
    lm.tambah(m, m1, m2);  
  
    lm.tampil();  
  
    lm.update(lm.LinearSearch(nim:"201235"), new Mahasiswa05(nim:"201235", nama:"Akhleema Lela",  
notelp:"021xx2"));  
    System.err.println(x:"");  
    lm.tampil();  
}
```

6. Hasil Run:

```
Mahasiswa{nim= 201234, nama=Noureen, notelp=021xx1}  
Mahasiswa{nim= 201235, nama=Akhleema, notelp=021xx2}  
Mahasiswa{nim= 201236, nama=Shannum, notelp=021xx3}  
  
Mahasiswa{nim= 201234, nama=Noureen, notelp=021xx1}  
Mahasiswa{nim= 201235, nama=Akhleema Lela, notelp=021xx2}  
Mahasiswa{nim= 201236, nama=Shannum, notelp=021xx3}
```

PERTANYAAN

1. Pada fungsi tambah() yang menggunakan unlimited argument itu menggunakan konsep apa? Dan kelebihan apa?
- Jawaban:** Pada fungsi tambah() menggunakan konsep atau method addAll() dari collections dimana konsep ini memiliki kelebihan yaitu dapat menambahkan element tanpa ada batas jumlah elemen yang ditambahkan.

2. Pada fungsi linearSearch() di atas, silakan diganti dengan fungsi binarySearch() dari collection!

Jawaban:

```
Mahasiswa05 key = new Mahasiswa05(nim:"201235", nama:null, notelp:null);  
lm.update(Collections.binarySearch(lm.mahasiswas, key, new MhsComparator05()),  
    new Mahasiswa05(nim:"201235", nama:"Akhleema", notelp:"021xx2"));  
System.out.println(x:"");  
lm.tampil();
```

```
import java.util.Comparator;
public class MhsComparator05 implements Comparator<Mahasiswa05> {

    public int compare(Mahasiswa05 mhs1, Mahasiswa05 mhs2) {
        if (mhs1.nim == mhs2.nim) {
            return 0;
        } else {
            return -1;
        }
    }

    @Override
    public Comparator<Mahasiswa05> reversed() {
        return Comparator.super.reversed();
    }
}
```

Hasil Run:

```
Mahasiswa{nim= 201234, nama=Noureen, notelp=021xx1}
Mahasiswa{nim= 201235, nama=Akhleema, notelp=021xx2}
Mahasiswa{nim= 201236, nama=Shannum, notelp=021xx3}

Mahasiswa{nim= 201234, nama=Noureen, notelp=021xx1}
Mahasiswa{nim= 201235, nama=Akhleema Lela, notelp=021xx2}
Mahasiswa{nim= 201236, nama=Shannum, notelp=021xx3}

Mahasiswa{nim= 201234, nama=Noureen, notelp=021xx1}
Mahasiswa{nim= 201235, nama=Akhleema, notelp=021xx2}
Mahasiswa{nim= 201236, nama=Shannum, notelp=021xx3}
```

3. Tambahkan fungsi sorting baik secara ascending ataupun descending pada class tersebut!

Jawaban:

```
void ascendingSort() {
    this.mahasiswas.sort((Mahasiswa05 d1, Mahasiswa05 d2) -> d1.nama.compareTo(d2.nama));
}

void descending() {
    this.mahasiswas.sort((Mahasiswa05 d1, Mahasiswa05 d2) -> d2.nama.compareTo(d1.nama));
}
```

```
System.out.println(x:"Ascending : ");
lm.ascendingSort();
lm.tampil();

System.out.println(x:"\n");

System.out.println(x:"Descending : ");
lm.descending();
lm.tampil();
```


Hasil Run:

```
Mahasiswa{nim= 201234, nama=Noureen, notelp=021xx1}
Mahasiswa{nim= 201235, nama=Akhleema, notelp=021xx2}
Mahasiswa{nim= 201236, nama=Shannum, notelp=021xx3}

Mahasiswa{nim= 201234, nama=Noureen, notelp=021xx1}
Mahasiswa{nim= 201235, nama=Akhleema lala, notelp=021xx2}
Mahasiswa{nim= 201236, nama=Shannum, notelp=021xx3}

Mahasiswa{nim= 201234, nama=Noureen, notelp=021xx1}
Mahasiswa{nim= 201235, nama=Akhleema, notelp=021xx2}
Mahasiswa{nim= 201236, nama=Shannum, notelp=021xx3}

Ascending :
Mahasiswa{nim= 201235, nama=Akhleema, notelp=021xx2}
Mahasiswa{nim= 201234, nama=Noureen, notelp=021xx1}
Mahasiswa{nim= 201236, nama=Shannum, notelp=021xx3}

Descending :
Mahasiswa{nim= 201236, nama=Shannum, notelp=021xx3}
Mahasiswa{nim= 201234, nama=Noureen, notelp=021xx1}
Mahasiswa{nim= 201235, nama=Akhleema, notelp=021xx2}
```

TUGAS PRAKTIKUM

1. Buatlah implementasi program daftar nilai mahasiswa semester, minimal memiliki 3 class yaitu Mahasiswa, Nilai, dan Mata Kuliah. Data Mahasiswa dan Mata Kuliah perlu melalui penginputan data terlebih dahulu

Class Mahasiswa

```
public class Mahasiswa {
    String nim, nama, telf;

    public Mahasiswa(String nim, String nama, String telf) {
        this.nim = nim;
        this.nama = nama;
        this.telf = telf;
    }

    void tampil() {
        System.out.printf(format: "%-10s %-15s %s", nim, nama, telf);
        System.out.println(x: "");
    }
}
```

Class MataKuliah

```
public class MataKuliah {
    String kode, namaMatkul;
    int sks;

    public MataKuliah(String kode, String namaMatkul, int sks) {
        this.kode = kode;
        this.namaMatkul = namaMatkul;
        this.sks = sks;
    }

    void tampil() {
        System.out.printf(format: "%-10s %-40s %d", kode, namaMatkul, sks);
        System.out.println(x: "");
    }
}
```

Class Nilai

```
public class Nilai {  
    double nilai;  
    String kode, kdMatkul, nimMhs;  
  
    public Nilai(String kode, double nilai, String kdMatkul, String nimMhs) {  
        this.nilai = nilai;  
        this.kode = kode;  
        this.kdMatkul = kdMatkul;  
        this.nimMhs = nimMhs;  
    }  
  
    void tampil() {  
        System.out.printf("%d", nilai);  
    }  
}
```

Class ListNilai

```
public class ListNilai {  
    List<Mahasiswa> mhs = new ArrayList<>();  
    List<MataKuliah> matkul = new ArrayList<>();  
    List<Nilai> listNilai = new ArrayList<>();  
  
    public void inputDataMahasiswa(Mahasiswa... mahasiswa) {  
        mhs.addAll(Arrays.asList(mahasiswa));  
    }  
  
    public void inputNilaiMahasiswa(Nilai n) {  
        listNilai.add(n);  
    }  
  
    public void inputDataMataKuliah(MataKuliah... mk) {  
        matkul.addAll(Arrays.asList(mk));  
    }  
  
    public void tampilDataMahasiswa() {  
        mhs.stream().forEach(mhs -> {  
            mhs.tampil();  
        });  
    }  
  
    public void tampilDataMataKuliah() {  
        matkul.stream().forEach(matkul -> {  
            matkul.tampil();  
        });  
    }  
  
    public void tampilListNilai() {  
        for (int i = 0; i < listNilai.size(); i++) {  
            int indexMhs = linearSearchMHS(listNilai.get(i).nimMhs);  
            int indexMatkul = linearSearchMatkul(listNilai.get(i).kdMatkul);  
            System.out.printf(format: "%-10s %-12s %-10.2f\n",  
                mhs.get(indexMhs).nama, matkul.get(indexMatkul).namaMatkul,  
                listNilai.get(i).nilai);  
        }  
    }  
  
    int linearSearchMHS(String nim) {  
        for (int i = 0; i < mhs.size(); i++) {  
            if (nim.equals(mhs.get(i).nim)) {  
                return i;  
            }  
        }  
        return -1;  
    }  
  
    int linearSearchMatkul(String kode) {  
        for (int i = 0; i < matkul.size(); i++) {  
            if (kode.equals(matkul.get(i).kode)) {  
                return i;  
            }  
        }  
        return -1;  
    }  
  
    int linearSearchNilaiMahasiswa(String nimMahasiswa) {  
        for (int i = 0; i < listNilai.size(); i++) {  
            if (nimMahasiswa.equals(listNilai.get(i).nimMhs)) {  
                return i;  
            }  
        }  
        return -1;  
    }  
  
    public void sortingData() {  
        Collections.sort(listNilai, new Comparator<Nilai>() {  
            public int compare(Nilai n1, Nilai n2) {  
                return Double.compare(n1.nilai, n2.nilai);  
            }  
        });  
    }  
  
    public void removeList () {  
        listNilai.remove(index:0);  
    }  
}
```

Class Main

```
import java.util.Scanner;
public class Main {
    Run | Debug
    public static void main(String[] args) {
        ListNilai ln = new ListNilai();

        Mahasiswa m1 = new Mahasiswa(nim:"20001", nama:"Thalhah", telf:"021xxx");
        Mahasiswa m2 = new Mahasiswa(nim:"20002", nama:"Zubair", telf:"021xxx");
        Mahasiswa m3 = new Mahasiswa(nim:"20003", nama:"Abdur-Rahman", telf:"021xxx");
        Mahasiswa m4 = new Mahasiswa(nim:"20004", nama:"Sa'ad", telf:"021xxx");
        Mahasiswa m5 = new Mahasiswa(nim:"20005", nama:"Sa'ad", telf:"021xxx");
        Mahasiswa m6 = new Mahasiswa(nim:"20006", nama:"Ubaidah", telf:"021xxx");
        ln.inputDataMahasiswa(m1, m2, m3, m4, m5, m6);

        MataKuliah mt11 = new MataKuliah(kode:"00001", "Internet " + "of Things",sks:3);
        MataKuliah mt12 = new MataKuliah(kode:"00002", "Algoritma " + "dan Struktur Data", sks:2);
        MataKuliah mt13 = new MataKuliah(kode:"00003", "Algoritma " + "dan Struktur Data", sks:2);
        MataKuliah mt14 = new MataKuliah(kode:"00004", "Praktikum " + "Algoritma dan Struktur Data", sks:3);
        MataKuliah mt15 = new MataKuliah(kode:"00005", "Praktikum " + "Algoritma dan Pemrograman", sks:3);
        ln.inputDataMataKuliah(mt11, mt12, mt13, mt14, mt15);

        Scanner sc = new Scanner(System.in);
        System.out.println(x:"=====");
        System.out.println(x:"SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER");
        System.out.println(x:"=====");
        boolean isTrue = true;
        while (isTrue){

            System.out.println(x:"\n1. Input Nilai");
            System.out.println(x:"2. Tampil Nilai");
            System.out.println(x:"3. Mencari Nilai Mahasiswa");
            System.out.println(x:"4. Urut Data Nilai");
            System.out.println(x:"5. Menghapus data pertama");
            System.out.println(x:"6. Keluar");
            System.out.println(x:"*****");
            System.out.print(s:"Pilih : ");
            int choice = sc.nextInt();
            switch (choice) {

                case 1:
                    System.out.println(x:"Masukkan data");
                    System.out.print(s:"Kode : ");
                    String kd = sc.next();
                    System.out.print(s:"Nilai : ");
                    double inputNilai = sc.nextDouble();

                    System.out.println(x:"DAFTAR MAHASISWA");
                    System.out.println(x:"*****");
                    System.out.printf(format:"%-10s %-15s %s\n",
                        ...args:"NIM", "Nama", "Telp");
                    ln.tampilDataMahasiswa();
                    System.out.println(x:"Pilih mahasiswa by nim: ");
                    String tempNim = sc.next();
```

```

System.out.println(x:"DAFTAR MATA KULIAH");
System.out.println(x:"*****");
System.out.printf(format:"%-10s %-40s %s\n",
...args:"Kode","Mata Kuliah","SKS");

ln.tampilDataMataKuliah();
System.out.print(s:"Pilih MK by kode: ");
String tempKd = sc.next();

Nilai temp = new Nilai(kd, inputNilai,
tempKd, tempNim);
ln.inputNilaiMahasiswa(temp);
break;

case 2:
System.out.println(x:"DAFTAR NILAI MAHASISWA");
System.out.println(x:"*****");
System.out.printf(format:"%-10s %-15s %-40s %-10s %s\n",
...args:"NIM", "Nama", "Mata Kuliah", "SKS", "Nilai");
ln.tampilListNilai();
break;

case 3:
System.out.println(x:"DAFTAR NILAI MAHASISWA");
System.out.println(x:"*****");
System.out.printf(format:"%-10s %-15s %-40s %-10s %s\n",
...args:"NIM", "Nama", "Mata Kuliah", "SKS", "Nilai");
ln.tampilListNilai();
System.out.print(s:"Masukkan data mahasiswa[nim]: ");
String tempNIM = sc.next();
int indexSearch = ln.linearSearchNilaiMahasiswa(tempNIM);
int indexMatkul = 0;
for(int i = 0; i < ln.matkul.size(); i++){
    if(ln.listNilai.get(indexSearch).
kdMatkul.equals(ln.matkul.get(i).kode)){
        indexMatkul = i;
        break;
    }
}
System.out.printf(format:"%-10s %-15s %-40s %-10s %s\n",
...args:"NIM", "Nama", "Mata Kuliah", "SKS", "Nilai");
System.out.printf(format:"%-10s %-15s %-40s %-10d %.2f\n",
ln.mhs.get(indexSearch).nim,
ln.mhs.get(indexSearch).nama,
ln.matkul.get(indexMatkul).namaMatkul,
ln.matkul.get(indexMatkul).sks,
ln.listNilai.get(indexSearch).nilai);
System.out.println("Total SKS "+ ln.matkul.get(indexMatkul).sks+" telah diambil");
break;

case 4:
System.out.println(x:"DAFTAR NILAI MAHASISWA");
System.out.println(x:"*****");
System.out.printf(format:"%-10s %-15s %-40s %-10s %s\n",
"NIM", "Nama", "Mata Kuliah", "SKS", "Nilai");
ln.sortingData();
ln.tampilListNilai();
break;

case 5:
System.out.println(x:"Data berhasil dihapus");
ln.removeList();
break;

case 6:
isTrue = false;
break;
default:
System.out.println(x:"Inputan salah");

```

Hasil Run:

```
=====
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
=====

1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Menghapus data pertama
6. Keluar
*****

Pilih : 1
Masukkan data
Kode : 0001
Nilai : 80.75
DAFTAR MAHASISWA
*****

NIM      Nama      Telp
20001    Thalhah    021xxx
20002    Zubair      021xxx
20003    Abdur-Rahman 021xxx
20004    Sa'ad       021xxx
20005    Sa'ad       021xxx
20006    Ubaidah     021xxx
Pilih mahasiswa by nim:
20001
DAFTAR MATA KULIAH
*****

Kode      Mata Kuliah      SKS
00001     Internet of Things 3
00002     Algoritma dan Struktur Data 2
00003     Algoritma dan Struktur Data 2
00004     Praktikum Algoritma dan Struktur Data 3
00005     Praktikum Algoritma dan Pemrograman 3
```

```
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Menghapus data pertama
6. Keluar
*****

Pilih : 3
DAFTAR NILAI MAHASISWA
*****

NIM      Nama      Mata Kuliah      SKS      Nilai
Thalhah  Internet of Things 80.75
Thalhah  Internet of Things 80.75
Masukkan data mahasiswa[nim]: 20001
NIM      Nama      Mata Kuliah      SKS      Nilai
20001    Thalhah  Internet of Things 3      80.75
Total SKS 3 telah diambil
```