

PRAKTIKUM ALGORITMA DAN STRUKTUR DATA
JOBSHEET PERTEMUAN KE-16



RIO TRI PRAYOGO

TI 1A

26

2341720236

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2024

Java Collection

Praktikum 1

Percobaan :

ContohList26

```
package minggu16;

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

public class ContohList26 {
    public static void main(String[] args) {
        List l = new ArrayList();

        l.add(1);
        l.add(2);
        l.add(3);
        l.add("Cireng");

        System.out.printf("Element 0: %d total element: %d elemen terakhir: %s\n", l.get(0), l.size(), l.get(l.size() - 1));

        l.add(4);
        l.remove(0);

        System.out.printf("Element 0: %d total element: %d elemen terakhir: %s\n", l.get(0), l.size(), l.get(l.size() - 1));

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

        names.add("Noureen");
        names.add("Akhleema");
        names.add("Shannum");
        names.add("Uwais");
        names.add("Al-Qarni");
    }
}
```

```

        System.out.printf("Element 0: %s total element: %d elemen terakhir: %s\n", names.get(0), names.size(),
                           names.get(names.size() - 1));

        names.set(0, "My kid");

        System.out.printf("Element 0: %s total element: %d elemen terakhir: %s\n", names.get(0), names.size(),
                           names.get(names.size() - 1));

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

```

Output :

```

Element 0: 1 total element: 4 elemen terakhir: Cireng
Element 0: 2 total element: 4 elemen terakhir: 4
Element 0: Nouredin total element: 5 elemen terakhir: Al-Qarni
Element 0: My kid total element: 5 elemen terakhir: Al-Qarni
Names: [My kid, Akhleema, Shannum, Uwais, Al-Qarni]

```

Pertanyaan :

- Perhatikan baris kode 25-36, mengapa semua jenis data bisa ditampung ke dalam sebuah ArrayList?
 - Karena **ArrayList** menyimpan data dalam bentuk sebuah objek sehingga dapat menyimpan data dalam bentuk tipe data apapun.
- Modifikasi baris kode 25-36 sehingga data yang ditampung hanya satu jenis atau spesifik tipe tertentu!
 - Kode dibawah merupakan hasil modifikasi:

```

List<String> l = new ArrayList();

l.add("1");

l.add("2");

l.add("3");

l.add("Cireng");

System.out.printf("Element 0: %d total element: %d elemen
terakhir: %s\n", l.get(0), l.size(),
                  l.get(l.size() - 1));

l.add("4");

l.remove(0);

System.out.printf("Element 0: %d total element: %d elemen
terakhir: %s\n", l.get(0), l.size(),
                  l.get(l.size() - 1));

```

Modifikasi untuk menampung satu jenis data dapat menambahkan deklarasi **<JenisData>** setelah deklarasi **List**. Dalam modifikasi diatas mengubah supaya jenis data dispesifikasi berupa **String** sehingga dalam penambahan data atau `l.add()` nya juga ditambahkan (" ") karena jenis datanya **String**.

3. Ubah kode pada baris kode 38 menjadi seperti ini

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

- Perubahan kode:

```
LinkedList<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());
```

- Penambahan kode:

```
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());
```

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

- Output setelah penambahan kode nomor 4 di bawah sendiri:

```
Element 0: Noreen total element: 5 elemen terakhir: Al-Qarni
Element 0: My kid total element: 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]
```

Perubahan yang terjadi dimana pada nomor 3 mengubah dari **List** menjadi **LinkedList** dan pada nomor 4 melakukan `push()`, `getFirst()`, dan `getLast()` dimana ketiga method tersebut merupakan method yang dimiliki oleh **LinkedList**. Sehingga jika menggunakan **List** ketiga method tersebut tidak dapat dijalankan.

Praktikum 2

Percobaan :

LoopCollection26

```
package minggu16;

import java.util.Stack;
import java.util.Iterator;

public class LoopCollection26 {
    public static void main(String[] args) {
        Stack<String> fruits = new Stack<>();
        fruits.push("Banana");
        fruits.add("Orange");
```

```

        fruits.add("Watermelon");
        fruits.add("Leci");
        fruits.add("Salak");

        for (String fruit : fruits) {
            System.out.printf("%s ", fruit);
        }

        System.out.println("\n" + fruits.toString());

        while (!fruits.empty()) {
            System.out.printf("%s ", fruits.pop());
        }

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

```

Output :

```

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`?
 - Secara fungsional kedua fungsi tersebut sama dimana menambahkan data dengan menggunakan konsep *LIFO (Last In First Out)*. Perbedaan dari dua fungsi tersebut dimana `push()` merupakan metode yang secara khusus digunakan oleh **Stack**, sedangkan `add()` merupakan metode umum dalam *Java*.
2. Silakan hilangkan baris 43 dan 44, apakah yang akan terjadi? Mengapa bisa demikian?
 - Output jika kedua baris tersebut dihapus:

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

Jika kedua baris tersebut dihapus maka output akan menjadi kosong karena pada baris sebelumnya program melakukan `pop()` atau mengeluarkan seluruh isi **Stack** dan kode dibawah kode yang dihapus yang melakukan iterasi untuk memunculkan isi **Stack** akan tetap berjalan dengan kondisi **Stack** kosong sehingga outputnya kosong.

3. Jelaskan fungsi dari baris 46-49?
 - Kode pada baris berfungsi untuk melakukan iterasi pada seluruh Stack dan memunculkannya. Iterasi dilakukan di baris `Iterator<String> it = fruits.iterator();` dan terus berjalan hingga `it.hasNext();` bernilai *false*. Lalu setiap iterasi tersebut disimpan pada baris `String fruit = it.next();` dan dimunculkan pada baris `System.out.printf("%s ", fruit);`.
4. Silakan ganti baris kode 25, `Stack<String>` menjadi `List<String>` dan apakah yang terjadi? Mengapa bisa demikian?
 - Pergantian baris kode diatas akan memunculkan error karena **List** tidak memiliki operasi yang sama dengan **Stack**.
5. Ganti elemen terakhir dari objek `fruits` menjadi "Strawberry"!
 - Baris kode mengganti elemen menjadi "Strawberry":

```
fruits.push("Strawberry");
```

Output:

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

6. Tambahkan 3 buah seperti "Mango", "guava", dan "avocado" kemudian dilakukan sorting!
 - Menambahkan 3 buah dan melakukan sorting secara Descending:

```
fruits.push("Strawberry");  
fruits.push("Mango");  
fruits.push("Guava");  
fruits.push("Avocado");  
List<String> list = new ArrayList<>();
```

```
        Collections.sort(list);  
        for (String fruit : list) {  
            fruits.push(fruit);  
        }  
    }
```

Output:

Strawberry Mango Guava Avocado

Sorting dilakukan dengan mengubah **Stack** menjadi sebuah **List** lebih dahulu dan dilakukan sort.

Praktikum 3

Percobaan :

Mahasiswa26

```
package minggul6;  
  
public class Mahasiswa26 {  
    String nim;  
    String nama;  
    String notelp;  
  
    public Mahasiswa26() {  
    }  
  
    public Mahasiswa26(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 + '}';  
    }  
}
```

ListMahasiswa26

```
package minggul6;
```

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

public class ListMahasiswa26 {
    List<Mahasiswa26> mahasiswas = new ArrayList<>();

    public void tambah(Mahasiswa26... mahasiswa) {
        mahasiswas.addAll(Arrays.asList(mahasiswa));
    }

    public void hapus(int index) {
        mahasiswas.remove(index);
    }

    public void update(int index, Mahasiswa26 mhs) {
        mahasiswas.set(index, mhs);
    }

    public void tampil() {
        mahasiswas.stream().forEach(mhs -> {
            System.out.println("'" + mhs.toString());
        });
    }

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

    public static void main(String[] args) {
```



```

ListMahasiswa26 lm = new ListMahasiswa26();

Mahasiswa26 m = new Mahasiswa26("201234", "Noureen", "021xx1");
Mahasiswa26 m1 = new Mahasiswa26("201235", "Akhleema", "021xx2");
Mahasiswa26 m2 = new Mahasiswa26("201236", "Shannum", "021xx3");

lm.tambah(m, m1, m2);

lm.tampil();

lm.update(lm.linearSearch("201235"), new Mahasiswa26("201235",
"Akhleema Lela", "021xx2"));

System.out.println("");

lm.tampil();
}
}

```

Output :

```

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?
 - Fungsi tambah() yang menggunakan unlimited argument menggunakan konsep dari *varargs (variable-length arguments)* dalam Java. *Varargs* memungkinkan sebuah method untuk menerima jumlah argumen yang dinamis dari tipe yang sama. Method tersebut menambahkan satu atau lebih objek Mahasiswa26 secara bersamaan tanpa perlu membatasi jumlah argumen yang diterima oleh metode tersebut.
2. Pada fungsi linearSearch() di atas, silakan diganti dengan fungsi binarySearch() dari collection!
 - Kode untuk menambahkan binarySearch() dan sorting untuk menjalankan search tersebut:

```

...

int binarySearch(String nim) {
    sortingAscending();

    return Collections.binarySearch(mahasiswas, new
    Mahasiswa26(nim, "", ""),
        (m1, m2) -> m1.nim.compareTo(m2.nim));
}

```

```

        public void sortingAscending() {
            Collections.sort(mahasiswas, (m1, m2) ->
m1.nim.compareTo(m2.nim));
        }
        ...
        ...

        lm.update(lm.binarySearch("201235"), new
Mahasiswa26("201235", "Akhleema Lela", "021xx2"));

        System.out.println("");

        lm.tampil();

        ...

```

Output:

```

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}

```

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

➤ Kode sorting ascending dan descending

```

        ...

        public void sortingAscending() {
            Collections.sort(mahasiswas, (m1, m2) ->
m1.nim.compareTo(m2.nim));
        }

        public void sortingDescending() {
            Collections.sort(mahasiswas, (m1, m2) ->
m2.nim.compareTo(m1.nim));
        }

        ...
        ...

        lm.sortingAscending();

        System.out.println("\nSorting Ascending:");

        lm.tampil();

        lm.sortingDescending();

        System.out.println("\nSorting Descending:");

        lm.tampil();

        ...

```

Output

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

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

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.

Kode :

Mahasiswa26

```
package minggu16.Tugas;

public class Mahasiswa26 {
    String nim;
    String nama;
    String notelp;

    public Mahasiswa26() {
    }

    public Mahasiswa26(String nim, String nama, String notelp) {
        this.nim = nim;
        this.nama = nama;
        this.notelp = notelp;
    }

    public String getNim() {
        return nim;
    }

    public String getNama() {
```

```
        return nama;
    }

    public String getTelf() {
        return notelp;
    }
}
```

Nilai26

```
package minggul6.Tugas;

public class Nilai26 {
    String nim, nama, mataKuliah, sks;
    double nilai;

    public Nilai26(String nim, String nama, String mataKuliah, String
sks, double nilai) {
        this.nim = nim;
        this.nama = nama;
        this.mataKuliah = mataKuliah;
        this.sks = sks;
        this.nilai = nilai;
    }

    public String getNim() {
        return nim;
    }

    public String getNama() {
        return nama;
    }

    public String getMataKuliah() {
        return mataKuliah;
    }

    public String getSks() {
```

```
        return sks;
    }

    public double getNilai() {
        return nilai;
    }
}
```

MataKuliah26

```
package minggu16.Tugas;

public class MataKuliah26 {
    String kode;
    String matkul;
    String sks;

    public MataKuliah26() {
    }

    public MataKuliah26(String kode, String matkul, String sks) {
        this.kode = kode;
        this.matkul = matkul;
        this.sks = sks;
    }

    public String getKode() {
        return kode;
    }

    public String getMataKul() {
        return matkul;
    }

    public String getSks() {
        return sks;
    }
}
```

Main26

```
package minggul6.Tugas;

import java.util.Scanner;
import java.util.Queue;
import java.util.LinkedList;

public class Main26 {

    static Queue<Mahasiswa26> mhs = new LinkedList<>();
    static Queue<MataKuliah26> matkul = new LinkedList<>();
    static Queue<Nilai26> score = new LinkedList<>();
    static Scanner sc = new Scanner(System.in);

    static void inputNilai() {
        System.out.println("--- INPUT NILAI ---");
        System.out.print("Masukkan nilai : ");
        double nilai = sc.nextDouble();
        sc.nextLine();
        System.out.println("DAFTAR MAHASISWA");
        System.out.println("*****");
        System.out.println("NIM Nama Telf");
        mhs.forEach((mhs) -> {
            System.out.printf("%-3s %-12s %-10s\n", mhs.getNim(),
mhs.getNama(), mhs.getTelf());
        });
        System.out.print("Masukkan NIM Mahasiswa : ");
        String nim = sc.nextLine();
        System.out.println("DAFTAR MATA KULIAH");
        System.out.println("*****");
        System.out.println("Kode Mata Kuliah SKS");
        matkul.forEach((mk) -> {
            System.out.printf("%-5s %-40s %-2s\n", mk.getKode(),
mk.getMataKul(), mk.getSks());
        });
        System.out.print("Masukkan Kode Mata Kuliah : ");
    }
}
```

```

String kode = sc.nextLine();
String nama = "";
for (Mahasiswa26 mhs : mhs) {
    if (mhs.getNim().equals(nim)) {
        nama = mhs.getNama();
    }
}
String mataKuliah = "";
String sks = "";
for (MataKuliah26 mk : matkul) {
    if (mk.getKode().equals(kode)) {
        mataKuliah = mk.getMataKul();
        sks = mk.getSks();
    }
}
score.add(new Nilai26(nim, nama, mataKuliah, sks, nilai));
}

static void tampilNilai() {
    System.out.println("DAFTAR NILAI MAHASISWA");
    System.out.println("*****");
    System.out.println("NIM Nama Mata Kuliah SKS Nilai");
    score.forEach((n) -> {
        System.out.printf("%-5s %-10s %-10s %-2s %-4s\n",
n.getNim(),
n.getNama(), n.getMataKuliah(), n.getSks(),
n.getNilai());
    });
}

static void cariNilai() {
    tampilNilai();
    System.out.print("Masukkan NIM Mahasiswa : ");
    String nim = sc.nextLine();
    boolean found = false;
    for (Nilai26 n : score) {

```

```

        if (n.getNim().equals(nim)) {
            System.out.println("NIM Nama Mata Kuliah SKS Nilai");
            System.out.printf("%-5s %-10s %-10s %-2s %-4s\n",
n.getNim(),
                                n.getNama(), n.getMataKuliah(), n.getSks(),
n.getNilai());
            found = true;
        }
    }
    if (!found) {
        System.out.println("Data tidak ditemukan");
    }
}

static void sortData() {
    Nilai26[] nilaiArr = new Nilai26[score.size()];
    int i = 0;
    for (Nilai26 n : score) {
        nilaiArr[i] = n;
        i++;
    }
    for (int j = 0; j < nilaiArr.length - 1; j++) {
        for (int k = 0; k < nilaiArr.length - j - 1; k++) {
            if (nilaiArr[k].getNilai() < nilaiArr[k + 1].getNilai())
{
                Nilai26 temp = nilaiArr[k];
                nilaiArr[k] = nilaiArr[k + 1];
                nilaiArr[k + 1] = temp;
            }
        }
    }
    score.clear();
    for (Nilai26 n : nilaiArr) {
        score.add(n);
    }
}
}

```



```

public static void main(String[] args) {
    mhs.add(new Mahasiswa26("20001", "Thalhah", "021xxx"));
    mhs.add(new Mahasiswa26("20002", "Zubair", "021xxx"));
    mhs.add(new Mahasiswa26("20003", "Abdul-Rahman", "021xxx"));
    mhs.add(new Mahasiswa26("20004", "Sa'ad", "021xxx"));
    mhs.add(new Mahasiswa26("20005", "Sa'id", "021xxx"));
    mhs.add(new Mahasiswa26("20006", "Ubaidah", "021xxx"));
    matkul.add(new MataKuliah26("00001", "Internet Of Things",
"3"));
    matkul.add(new MataKuliah26("00002", "Algoritma dan Struktur
Data", "2"));
    matkul.add(new MataKuliah26("00003", "Algoritma dan Pemograman",
"2"));
    matkul.add(new MataKuliah26("00004", "Praktikum Algoritma dan
Struktur Data", "3"));
    matkul.add(new MataKuliah26("00004", "Praktikum Algoritma dan
Pemograman", "3"));
    score.add(new Nilai26("20002", "Zubair", "Internet Of Things",
"3", 90));

    boolean run = true;
    do {
        System.out.println("*****
*****");
        System.out.println("SISTEM PENGOLAHAN DATA NILAI MAHASISWA
SEMESTER");
        System.out.println("*****
*****");
        System.out.println("1. Input Nilai");
        System.out.println("2. Tampil Nilai");
        System.out.println("3. Mencari Nilai Mahasiswa");
        System.out.println("4. Urut Data Nilai");
        System.out.println("0. Keluar");
        System.out.println("*****
*****");
        System.out.print("Pilih: ");
        int pilihan = sc.nextInt();
        sc.nextLine();
        switch (pilihan) {
            case 1:
                inputNilai();

```

```

        break;

    case 2:

        tampilNilai();

        break;

    case 3:

        cariNilai();

        break;

    case 4:

        sortData();

        tampilNilai();

        break;

    case 0:

        System.out.println("Keluar program\nTerimakasih!");

        run = false;

        break;

    default:

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

    }

} while (run);

}

}

```

Output :

Menu Awal	Penambahan Data
<pre> ***** SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER ***** 1. Input Nilai 2. Tampil Nilai 3. Mencari Nilai Mahasiswa 4. Urut Data Nilai 0. Keluar ***** Pilih: </pre>	<pre> Pilih: 1 --- INPUT NILAI --- Masukkan nilai : 80.75 DAFTAR MAHASISWA ***** NIM Nama Telf 20001 Thalhhah 02100x 20002 Zubair 02100x 20003 Abdul-Rahman 02100x 20004 Sa'ad 02100x 20005 Sa'id 02100x 20006 Ubaidah 02100x Masukkan NIM Mahasiswa : 20001 DAFTAR MATA KULIAH ***** Kode Mata Kuliah SKS 00001 Internet Of Things 3 00002 Algoritma dan Struktur Data 2 00003 Algoritma dan Pemrograman 2 00004 Praktikum Algoritma dan Struktur Data 3 00004 Praktikum Algoritma dan Pemrograman 3 Masukkan Kode Mata Kuliah : 00001 </pre>
Tampil Nilai	Pencarian Data Mahasiswa
<pre> Pilih: 2 DAFTAR NILAI MAHASISWA ***** NIM Nama Mata Kuliah SKS Nilai 20002 Zubair Internet Of Things 3 90.0 20001 Thalhhah Internet Of Things 3 80.75 </pre>	<pre> Pilih: 3 DAFTAR NILAI MAHASISWA ***** NIM Nama Mata Kuliah SKS Nilai 20002 Zubair Internet Of Things 3 90.0 20001 Thalhhah Internet Of Things 3 80.75 Masukkan NIM Mahasiswa : 20002 NIM Nama Mata Kuliah SKS Nilai 20002 Zubair Internet Of Things 3 90.0 </pre>
Pengurutan Data Nilai	
<pre> Pilih: 4 DAFTAR NILAI MAHASISWA ***** NIM Nama Mata Kuliah SKS Nilai 20002 Zubair Internet Of Things 3 90.0 20001 Thalhhah Internet Of Things 3 80.75 </pre>	

2. Tambahkan prosedur hapus data mahasiswa melalui implementasi Queue pada collections
Tugas nomor 1!

Kode :

Main26

```
...

static void hapusData() {
    tampilNilai();
    System.out.print("Masukkan NIM Mahasiswa : ");
    String nim = sc.nextLine();
    boolean found = false;
    for (Nilai26 n : score) {
        if (n.getNim().equals(nim)) {
            score.remove(n);
            found = true;
        }
    }
    if (!found) {
        System.out.println("Data tidak ditemukan");
    }
}

...

...

    System.out.println("5. Hapus Data Nilai");

...

...

        case 5:
            hapusData();
            break;

...

```

Output :

<i>Tampil Data Sebelum Hapus</i>	<i>Hapus Data</i>	<i>Tampil Data Setelah Hapus</i>
<pre>Pilih: 2 DAFTAR NILAI MAHASISWA ***** NIM Nama Mata Kuliah SKS Nilai 20002 Zubair Internet Of Things 3 90.0 20001 Thalbah Internet Of Things 3 80.75</pre>	<pre>Pilih: 5 DAFTAR NILAI MAHASISWA ***** NIM Nama Mata Kuliah SKS Nilai 20002 Zubair Internet Of Things 3 90.0 20001 Thalbah Internet Of Things 3 80.75 Masukkan NIM Mahasiswa : 20001</pre>	<pre>Pilih: 2 DAFTAR NILAI MAHASISWA ***** NIM Nama Mata Kuliah SKS Nilai 20002 Zubair Internet Of Things 3 90.0</pre>