

16.1. Kegiatan Praktikum 1

16.2. Percobaan 1

```
// LinkedList<Integer> l = new LinkedList<>();
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));

LinkedList<String> names = new LinkedList<>();
names.add (e:"Noureen");
names.add (e:"Akh leena");
names.add (e:"Shannun");
names.add (e:"Uwais");
names.add (e:"A1-0arni");

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));
```

```
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: A1-0arni
Elemen 0: My kid total elemen: 5 elemen terakhir: A1-0arni
Names: [My kid, Akh leena, Shannun, Uwais, A1-0arni]
Elemen 0: Hei-hei total elemen: 6 elemen terakhir: A1-0arni
Names: [Hei-hei, My kid, Akh leena, Shannun, Uwais, A1-0arni]
```

16.2.1. Pertanyaan Percobaan

1. Perhatikan baris kode 25-36, mengapa semua jenis data bisa ditampung ke dalam sebuah ArrayList?
2. Modifikasi baris kode 25-36 sehingga data yang ditampung hanya satu jenis atau spesifik tipe tertentu!
3. Ubah kode pada baris kode 38 menjadi seperti ini

```
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());
```

5. Dan penambeanan kode tersebut, apakah dijalankan dan apakah yang dapat Anda jelaskan:

16.3.1 Tahapan Percobaan

```

Collecton > LoopCollection0.java > LoopCollection0 > @mainString@
1  import java.util.Iterator;
2  import java.util.Stack;
3
4  public class LoopCollection0 {
5      @Override
6      public static void main(String[] args) {
7          Stack<String> fruits = new Stack<>();
8          fruits.push(item:"Banana");
9          fruits.add(e:"Orange");
10         fruits.add(e:"Watermelon");
11         fruits.add(e:"Lemon");
12         fruits.push(item:"Salak");
13
14         for (String fruit : fruits) {
15             System.out.printf (format:"%s ", fruit);
16         }
17
18         System.out.println("\n" + fruits.toString());
19
20         while (!fruits.empty()) {
21             System.out.printf (format:"%s ", fruits.pop());
22         }
23
24         fruits.push(item:"Melon");
25         fruits.push(item:"Durian");
26         System.out.println(a:"");
27
28         fruits.pop(); // Remove the last element
29         fruits.push(item:"Strawberry");
30
31         // Add New Fruits (Mango, Guava, Avocado)
32         fruits.addAll(java.util.Arrays.asList(...a:"Mango", "Guava", "Avocado"));
33
34         // Sort the fruits (ascending order)
35         fruits.sort(java.util.Comparator.naturalOrder());
36
37         for(Iterator<String> it = fruits.iterator(); it.hasNext(); ) {
38             String fruit = it.next();
39             System.out.printf (format:"%s ", fruit);
40         }
41
42         System.out.println(a:"");
43         fruits.stream().forEach(e -> {
44             System.out.printf(format:"%s ", e);
45         });
46
47         System.out.println(a:"");
48         for (int i=0; i < fruits.size(); i++) {
49             System.out.printf(format:"%s ", fruits.get(i));
50         }
51     }
52 }

```

16.3.3 Pertanyaan Percobaan

1. Apakah perbedaan fungsi push() dan add() pada objek *fruits*?
Menambahkan elemen baru dari near objek stack push seperti add tapi dengan lifo
2. Silakan hilangkan baris 43 dan 44, apakah yang akan terjadi? Mengapa bisa demikian?
Method get tidak menjalankan apa apa
3. Jelaskan fungsi dari baris 46-49?
Untuk menampilkan diseluruh stack
4. Silakan ganti baris kode 25, *Stack<String>* menjadi *List<String>* dan apakah yang terjadi?
Mengapa bisa demikian?
Error, karena program tidak dituliskan "import java.util.List"
5. Ganti elemen terakhir dari dari objek fruits menjadi "Strawberry"!

```

fruits.push(item:"Melon");
fruits.push(item:"Durian");
System.out.println(a:"");

fruits.pop();
fruits.push(item:"Strawberry");

```

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

```

fruits.addAll(java.util.Arrays.asList(...a:"Mango", "Guava", "Avocado"));

// Sort the Fruits (ascending order)
fruits.sort(java.util.Comparator.naturalOrder());

```

16.4 Kegiatan Praktikum 3

16.4.1 Tahapan Percobaan

1. Membuat class mahasiswa dengan atribut, konstruktor, dan fungsi

```
Collection > J Mahasiswal0.java > Mahasiswal0 > toString()
1 public class Mahasiswal0 {
2     String nim;
3     String nama;
4     String notelp;
5
6     public Mahasiswal0() {
7     }
8     public Mahasiswal0 (String nim, String nama, String notelp) {
9         this.nim = nim;
10        this.nama = nama;
11        this.notelp = notelp;
12    }
13
14    @Override
15    public String toString() {
16        return "Mahasiswa[" + "nim" + " = " + nim + ", nama" + " = " + nama + ", notelp" + " = " + notelp + " ]";
17    }
18 }
```

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

```
1 }
2
3 public class ListMahasiswa10 {
4     List<Mahasiswal0> mahasiswas = new ArrayList<>();
5 }
```

3. Method tambah(), hapus(), update(), dan tampil() secara berurut dibuat agar bisa melakukan

```
21 public class ListMahasiswa10 {
22     List<Mahasiswal0> mahasiswas = new ArrayList<>();
23
24     public void tambah(Mahasiswal0... mahasiswa) {
25         mahasiswas.addAll(Arrays.asList(mahasiswa));
26     }
27     public void hapus(int index) {
28         mahasiswas.remove(index);
29     }
30     public void update(int index, Mahasiswal0 mhs) {
31         mahasiswas.set(index, mhs);
32     }
33
34     public void tampil() {
35         mahasiswas.stream().forEach(mhs -> {
36             System.out.println(" " + mhs.toString());
37         });
38     }
39 }
40
41
```

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< mahasiswa.size(); i++){
            if(nim.equals(mahasiswa.get(i).nim)){
                return i;
            }
        }
        return -1;
    }
}

```

5. Menambahkan method main

```

public static void main(String [] args) {
    Mahasiswa m = new Mahasiswa();
    Mahasiswa m1 = new Mahasiswa(nim:"201234", nama:"Muhammad", notelp:"021xx3");
    Mahasiswa m2 = new Mahasiswa(nim:"201235", nama:"Akhleena", notelp:"021xx2");
    Mahasiswa m3 = new Mahasiswa(nim:"201236", nama:"Shannaz", notelp:"021xx1");

    m.tambah(m1, m2);

    m.tampil();

    m.update(m.linearSearch("201235"), new Mahasiswa(nim:"201235", nama:"Akhleena Iela", notelp:"021xx2"));
    System.out.println(m);

    m.tampil();
}

```

16.4.3 Pertanyaan Percobaan

1. Pada fungsi tambah() yang menggunakan unlimited argument itu menggunakan konsep apa? Dan kelebihan apa?

Menggunakan konsep variable argument yang memungkinkan menerima jumlah argument yang terdefinisi, kelebihan yaitu menerima sejumlah argument yang tidak terdefinisi dan meningkatkan fleksibilitas kode

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

```

int binarySearch(String nim) {
    int low = 0;
    int high = mahasiswa.size() - 1;

    while(low <= high) {
        int mid = low + (high - low) / 2;
        Mahasiswa mhs = mahasiswa.get(mid);

        if(mhs.nim.equals(nim)){
            return mid;
        } else if (mhs.nim.compareTo(nim) < 0) {
            low = mid + 1;
        } else {
            high = mid - 1;
        }
    }
    return -1;
}

```

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

```

public void sortByNimAsc() {
    Collections.sort(mahasiswa, (mhs1, mhs2) -> mhs1.nim.compareTo(mhs2.nim));
}

public void sortByNimDesc() {
    Collections.sort(mahasiswa, (mhs1, mhs2) -> mhs2.nim.compareTo(mhs1.nim));
}

```

16.5 Tugas Praktikum

- 16.5.1. 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.

Ilustrasi Program

Menu Awal dan Penambahan Data

```
*****
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
*****

1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
*****
Pilih      : |
```

Pilih : 1
Masukan data
Kode : 0001
Nilai : 80.75

DAFTAR MAHASISWA

```
*****
NIM      Nama      Telf
20001    Thalhah    021xxx
20002    Zubair     021xxx
20003    Abdur-Rahman 021xxx
20004    Sa'ad      021xxx
20005    Sa'id      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 Pemrograman 2
00004    Praktikum Algoritma dan Struktur Data 3
00005    Praktikum Algoritma dan Pemrograman 3
Pilih MK by kode: 00001
```

Tampil Nilai

```
*****
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
*****
```

1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar

```
*****
Pilih : 2
```

DAFTAR NILAI MAHASISWA

```
*****
Nim      Nama      Mata Kuliah      SKS      Nilai
20001    Thalhah    Internet of Things 3         80.75
```

Pencarian Data Mahasiswa

```
*****
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
*****
```

1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar

```
*****
Pilih : 3
```

DAFTAR NILAI MAHASISWA

```
*****
Nim      Nama      Mata Kuliah      SKS      Nilai
20001    Thalhah    Internet of Things 3         90.00
20002    Zubair     Praktikum Algoritma dan Pemrograman 3         80.75
Masukkan data mahasiswa[nim] :20002
Nim      Nama      Mata Kuliah      SKS      Nilai
20002    Zubair     Praktikum Algoritma dan Pemrograman 3         80.75
Total SKS 3 telah diambil.
```

Pengurutan Data Nilai

SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER

1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar

Pilih : 4

DAFTAR NILAI MAHASISWA

Nim	Nama	Mata Kuliah	SKS	Nilai
20002	Zubair	Praktikum Algoritma dan Pemrograman	3	80.75
20001	Thalhah	Internet of Things	3	90.00

```

Collection > TugasPraktikum > J Mahasiswa10.java > ...
1 package TugasPraktikum;
2
3 public class Mahasiswa10 {
4     String nim;
5     String nama;
6     String telf;
7
8     Mahasiswa10(String nim, String nama, String telf) {
9         this.nim = nim;
10        this.nama = nama;
11        this.telf = telf;
12    }
13 }
14

```

```

Collection > TugasPraktikum > J Main10.java > Main10 > main(String[])
1 package TugasPraktikum;
2 import java.util.Scanner;
3
4 public class Main10 {
5     Run / Debug
6     public static void main(String[] args) {
7         SistemNilai10 sistem = new SistemNilai10();
8         Scanner input = new Scanner(System.in);
9         int pilihan;
10
11         // insert data mahasiswa
12         sistem.tambahMahasiswa(nim:"20001", nama:"Thalhah", telf:"021xxx");
13         sistem.tambahMahasiswa(nim:"20002", nama:"Zubair", telf:"021xxx");
14         sistem.tambahMahasiswa(nim:"20003", nama:"Abdur-Rahman", telf:"021xxx");
15         sistem.tambahMahasiswa(nim:"20004", nama:"Sa'ad", telf:"021xxx");
16         sistem.tambahMahasiswa(nim:"20005", nama:"Sa'id", telf:"021xxx");
17         sistem.tambahMahasiswa(nim:"20006", nama:"Ubaidah", telf:"021xxx");
18
19         sistem.tambahMatakuliah(kode:"00001", nama:"Internet of Things", sks:3);
20         sistem.tambahMatakuliah(kode:"00002", nama:"Algoritma dan Struktur Data", sks:3);
21         sistem.tambahMatakuliah(kode:"00003", nama:"Algoritma dan Pemrograman", sks:3);
22         sistem.tambahMatakuliah(kode:"00004", nama:"Praktikum Algoritma dan Struktur Data", sks:3);
23         sistem.tambahMatakuliah(kode:"00005", nama:"Praktikum Algoritma dan Pemrograman", sks:3);
24
25         do {
26             System.out.println(x:"-----");
27             System.out.println(x:" SISTEM PENGOLAHAN DATA NILAI MAHASISWA ");
28             System.out.println(x:"-----");
29             System.out.println(x:"1. Input Nilai");
30             System.out.println(x:"2. Tampil Nilai");
31             System.out.println(x:"3. Mencari Nilai Mahasiswa");
32             System.out.println(x:"4. Urut Data Nilai");
33             // System.out.println("5. Keluar");
34             System.out.println(x:"5. Tambah Mahasiswa ke Antrian Penghapusan");
35             System.out.println(x:"6. Hapus Mahasiswa dari Antrian");
36             System.out.println(x:"7. Keluar");
37             System.out.print(s:"Pilih: ");
38             pilihan = input.nextInt();
39             input.nextLine();
40
41             switch (pilihan) {
42                 case 1:
43                     System.out.print(s:"Masukkan NIM: ");
44                     String nim = input.nextLine();
45                     System.out.print(s:"Masukkan Kode Mata Kuliah: ");
46                     String kodeMk = input.nextLine();
47                     System.out.print(s:"Masukkan Nilai: ");
48                     double nilai = input.nextDouble();
49                     sistem.inputNilai(nim, kodeMk, nilai);
50                     break;
51                 case 2:
52                     sistem.tampilNilai();
53                     break;
54                 case 3:
55                     System.out.print(s:"Masukkan NIM: ");
56                     nim = input.nextLine();
57                     sistem.cariNilaiMahasiswa(nim);
58                     break;
59                 case 4:
60                     sistem.urutDataNilai();
61                     break;
62                 // case 5:
63                 //     return;
64                 case 5:
65                     System.out.print(s:"Masukkan NIM Mahasiswa yang akan ditambahkan ke antrian penghapusan: ");
66                     nim = input.nextLine();
67                     sistem.antrianHapusMahasiswa(nim);
68                     break;
69                 case 6:
70                     sistem.hapusMahasiswa();
71                     break;
72                 case 7:
73                     return;
74                 default:
75                     System.out.println(x:"Pilih antara 1-7.");
76             }
77         } while (pilihan != 7);
78     }
79 }

```



```
1 package TugasPraktikum;
2
3 public class Matakuliah10 {
4     String kode;
5     String nama;
6     int sks;
7
8     Matakuliah10(String kode, String nama, int sks) {
9         this.kode = kode;
10        this.nama = nama;
11        this.sks = sks;
12    }
13 }
14
```

```
1 package TugasPraktikum;
2
3 public class Nilai {
4     Mahasiswa10 mahasiswa;
5     Matakuliah10 matakuliah;
6     double nilai;
7
8     Nilai(Mahasiswa10 mahasiswa, Matakuliah10 matakuliah, double nilai) {
9         this.mahasiswa = mahasiswa;
10        this.matakuliah = matakuliah;
11        this.nilai = nilai;
12    }
13 }
14
```

```

1 package 'Superlatium';
2
3 import java.util.ArrayList;
4 import java.util.Comparator;
5 import java.util.ArrayList;
6 import java.util.List;
7 import java.util.Queue;
8
9 public class Superlatium {
10     List<Merkantilite> daftarMerkantilite = new ArrayList<>();
11     List<Merkantilite> daftarMerkantilite = new ArrayList<>();
12     List<Merkantilite> daftarMerkantilite = new ArrayList<>();
13     Queue<Merkantilite> daftarMerkantilite = new LinkedList<>();
14
15     void tambahMerkantilite(String nama, String alamat, String tanggal) {
16         Merkantilite merkantilite = new Merkantilite(nama, alamat, tanggal);
17         daftarMerkantilite.add(merkantilite);
18     }
19
20     void tambahMerkantilite(String nama, String alamat, int saldo) {
21         Merkantilite merkantilite = new Merkantilite(nama, alamat, saldo);
22         daftarMerkantilite.add(merkantilite);
23     }
24
25     void hapusMerkantilite(String nama, String alamat, double saldo) {
26         Merkantilite merkantilite = daftarMerkantilite.stream().filter(m -> m.nam.equals(nama)).findFirst().orElse(null);
27         Merkantilite merkantilite = daftarMerkantilite.stream().filter(m -> m.alam.equals(alamat)).findFirst().orElse(null);
28
29         if (merkantilite != null && merkantilite != null) {
30             Merkantilite merkantilite = new Merkantilite(merkantilite.nam, merkantilite.alam);
31             daftarMerkantilite.remove(merkantilite);
32         }
33         if (merkantilite == null) {
34             System.out.println("Merkantilite dengan nama " + nama + " tidak ditemukan.");
35         }
36         if (merkantilite == null) {
37             System.out.println("Merkantilite dengan alamat " + alamat + " tidak ditemukan.");
38         }
39     }
40
41     void tampilMerkantilite() {
42         System.out.println("Merkantilite yang terdaftar");
43         System.out.println("Merkantilite yang terdaftar");
44         for (Merkantilite merkantilite : daftarMerkantilite) {
45             System.out.println("Merkantilite " + merkantilite.nam);
46             System.out.println("Merkantilite " + merkantilite.alam);
47             System.out.println("Merkantilite " + merkantilite.saldo);
48             System.out.println("Merkantilite " + merkantilite.alam);
49             System.out.println("Merkantilite " + merkantilite.saldo);
50             System.out.println("Merkantilite " + merkantilite.alam);
51         }
52     }
53
54     void cariMerkantilite(String nama) {
55         System.out.println("Merkantilite yang terdaftar");
56         System.out.println("Merkantilite yang terdaftar");
57         System.out.println("Merkantilite yang terdaftar");
58         for (Merkantilite merkantilite : daftarMerkantilite) {
59             if (merkantilite.nam.equals(nama)) {
60                 System.out.println("Merkantilite " + merkantilite.nam);
61                 System.out.println("Merkantilite " + merkantilite.alam);
62                 System.out.println("Merkantilite " + merkantilite.saldo);
63                 System.out.println("Merkantilite " + merkantilite.alam);
64                 System.out.println("Merkantilite " + merkantilite.saldo);
65                 System.out.println("Merkantilite " + merkantilite.alam);
66             }
67         }
68     }
69 }

```

```

89
90 void urutDataNilai() {
91     daftarNilai.sort(Comparator.comparing(n -> n.mahasiswa.nama));
92     tampilNilai();
93 }
94
95 void antrianHapusMahasiswa(String nim) {
96     try {
97         Mahasiswa10 mahasiswa = daftarMahasiswa.stream().filter(m -> m.nim.equals(nim)).findFirst().orElse(null);
98         if (mahasiswa != null) {
99             antrianHapus.add(mahasiswa);
100             System.out.println("Mahasiswa dengan NIM " + nim + " ditambahkan ke dalam antrian penghapusan.");
101         } else {
102             System.out.println("Mahasiswa dengan NIM " + nim + " tidak ditemukan.");
103         }
104     } catch (NullPointerException e) {
105         System.out.println("Terjadi kesalahan saat mencari mahasiswa. Silakan periksa input NIM.");
106     }
107 }
108
109 void hapusMahasiswa() {
110     Mahasiswa10 mahasiswa = antrianHapus.poll();
111     if (mahasiswa != null) {
112         daftarMahasiswa.remove(mahasiswa);
113         daftarNilai.removeIf(nilai -> nilai.mahasiswa.equals(mahasiswa));
114         System.out.println("Mahasiswa dengan NIM " + mahasiswa.nim + " telah dihapus.");
115     } else {
116         System.out.println("Tidak ada mahasiswa dalam antrian penghapusan.");
117     }
118 }
119 }

```

```

=====
SISTEM PENGOLAHAN DATA NILAI MAHASISWA
=====
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Tambah Mahasiswa ke Antrian Penghapusan
6. Hapus Mahasiswa dari Antrian
7. Keluar
Pilih: 1
Masukkan NIM: 20001
Masukkan Kode Mata Kuliah: 00001
Masukkan Nilai: 89
=====

```

```

=====
SISTEM PENGOLAHAN DATA NILAI MAHASISWA
=====
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Tambah Mahasiswa ke Antrian Penghapusan
6. Hapus Mahasiswa dari Antrian
7. Keluar
Pilih: 2
=====
DAFTAR NILAI MAHASISWA
=====
NIM      : 20001
Nama     : Thalhah
Mata Kuliah : Internet of Things
SKS      : 3
Nilai    : 89.0
=====

```

```

=====
SISTEM PENGOLAHAN DATA NILAI MAHASISWA
=====
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Tambah Mahasiswa ke Antrian Penghapusan
6. Hapus Mahasiswa dari Antrian
7. Keluar
Pilih: 3
Masukkan NIM: 20002
=====
NILAI MAHASISWA NIM: 20002
=====
SISTEM PENGOLAHAN DATA NILAI MAHASISWA
=====
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Tambah Mahasiswa ke Antrian Penghapusan
6. Hapus Mahasiswa dari Antrian
7. Keluar
Pilih: 4
=====
DAFTAR NILAI MAHASISWA
=====
NIM      : 20002
Nama     : Thalheh
Mata Kuliah : Internet of Things
SKS      : 3
Nilai    : 89.0
=====

```

```

=====
SISTEM PENGOLAHAN DATA NILAI MAHASISWA
=====
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Tambah Mahasiswa ke Antrian Penghapusan
6. Hapus Mahasiswa dari Antrian
7. Keluar
Pilih: 5
Masukkan NIM Mahasiswa yang akan ditambahkan ke antrian penghapusan: 200005
Mahasiswa dengan NIM 200005 ditambahkan ke dalam antrian penghapusan.
=====
SISTEM PENGOLAHAN DATA NILAI MAHASISWA
=====
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Tambah Mahasiswa ke Antrian Penghapusan
6. Hapus Mahasiswa dari Antrian
7. Keluar
Pilih: 6
Mahasiswa dengan NIM 200005 telah dihapus.
=====
SISTEM PENGOLAHAN DATA NILAI MAHASISWA
=====
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Tambah Mahasiswa ke Antrian Penghapusan
6. Hapus Mahasiswa dari Antrian
7. Keluar
Pilih:

```

