

Nama : Wahyudi Satriawan Hamid

Nim : 244107020137

## Verifikasi Hasil Percobaan 1

```
Linked list kosong
Isi Linked List:
Dirga      23004      4D      3.60

Isi Linked List:
Dirga      23004      4D      3.60
Alvaro     23001      1A      3.75

Isi Linked List:
Dirga      23004      4D      3.60
Cintia     23003      3C      3.90
Bimon      23002      2B      3.60
Alvaro     23001      1A      3.75
```

## Pertanyaan 1

### 1. Mengapa hasil awal “Linked List Kosong”?

Karena belum ada node ditambahkan, head == null.

### 2. Apa kegunaan temp pada traversal?

Sebagai pointer bantu untuk menjelajah node satu per satu.

```
boolean cek = true;
while (cek) {
    System.out.print("NIM : ");
    String nim = sc.nextLine();
    System.out.print("Nama : ");
    String nama = sc.nextLine();
    System.out.print("Kelas : ");
    String kelas = sc.nextLine();
    System.out.print("IPK : ");
    double ipk = sc.nextDouble(); sc.nextLine();

    Mahasiswa25 mhs = new Mahasiswa25(nim, nama, kelas, ipk);
    list.addLast(mhs);

    System.out.print("Tambah data lagi? (y/n): ");
    String ulang = sc.nextLine();
    if (!ulang.equalsIgnoreCase("y")) {
        cek = false;
    }
}
```

## Verifikasi Hasil Percobaan 2

```
data index 1:  
data mahasiswa an Bimon berada pada index: 2
```

```
Isi Linked List:
```

```
Cintia      23003      3C      3.90
```

```
Bimon       23002      2B      3.60
```

```
Isi Linked List:
```

```
Bimon       23002      2B      3.60
```

## Pertanyaan 2

1. Mengapa digunakan keyword break pada fungsi remove?

Untuk **menghentikan loop** setelah data ditemukan dan dihapus, agar tidak terus mencari ke node selanjutnya dan menghindari kesalahan logika.

2.
  - `temp = temp.next.next;` → **Melewatkan node yang ingin dihapus**, menyambungkan node sekarang ke node setelah target.
  - `if (temp.next == null)` → Jika node yang tersisa adalah node terakhir, maka **update tail** agar tetap menunjuk ke ujung list.

# Tugas

```
=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN ===
1. Tambah Mahasiswa ke Antrian
2. Panggil Antrian
3. Cek Antrian Kosong
4. Cek Antrian Penuh
5. Lihat Antrian Terdepan
6. Lihat Antrian Paling Belakang
7. Tampilkan Semua Antrian
8. Tampilkan Jumlah Mahasiswa yang Mengantri
9. Kosongkan Antrian
0. Keluar
Pilih: 1
NIM   : 132132
Nama  : Wahyu
Prodi : TI
Mahasiswa berhasil ditambahkan ke antrian.
```

```
=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN ===
1. Tambah Mahasiswa ke Antrian
2. Panggil Antrian
3. Cek Antrian Kosong
4. Cek Antrian Penuh
5. Lihat Antrian Terdepan
6. Lihat Antrian Paling Belakang
7. Tampilkan Semua Antrian
8. Tampilkan Jumlah Mahasiswa yang Mengantri
9. Kosongkan Antrian
0. Keluar
Pilih: 3
Antrian tidak kosong.
```

```
=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN ===
1. Tambah Mahasiswa ke Antrian
2. Panggil Antrian
3. Cek Antrian Kosong
4. Cek Antrian Penuh
5. Lihat Antrian Terdepan
6. Lihat Antrian Paling Belakang
7. Tampilkan Semua Antrian
8. Tampilkan Jumlah Mahasiswa yang Mengantri
9. Kosongkan Antrian
0. Keluar
Pilih: 2
Mahasiswa yang dipanggil:
NIM   : 132132
Nama  : Wahyu
Prodi : TI
```

```
=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN ===
1. Tambah Mahasiswa ke Antrian
2. Panggil Antrian
3. Cek Antrian Kosong
4. Cek Antrian Penuh
5. Lihat Antrian Terdepan
6. Lihat Antrian Paling Belakang
7. Tampilkan Semua Antrian
8. Tampilkan Jumlah Mahasiswa yang Mengantri
9. Kosongkan Antrian
0. Keluar
Pilih: 3
Antrian kosong.
```