

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
JOBSHEET 11**



Disusun Oleh :

Nama : Nawaf Azril Annaufal

Nim : 244107020047

Kelas : TI 1E

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLINEMA
2025**

Percobaan 1: Pembuatan Single Linked List

a. Hasil penulisan kode program untuk class Mahasiswa21.java

```
package Praktikum_ASD.Jobsheet11;

public class Mahasiswa21 {

    String nim, nama, kelas;
    double ipk;

    Mahasiswa21(String nim, String nama, String kelas, double ipk) {
        this.nim = nim;
        this.nama = nama;
        this.kelas = kelas;
        this.ipk = ipk;
    }

    void tampilInformasi() {
        System.out.printf("%-10s %-10s %-5s %.1f\n", nama, nim, kelas, ipk);
    }
}
```

b. Hasil penulisan kode program untuk class NodeMahasiswa21.java

```
package Praktikum_ASD.Jobsheet11;

public class NodeMahasiswa21 {

    Mahasiswa21 data;
    NodeMahasiswa21 next;

    public NodeMahasiswa21(Mahasiswa21 data, NodeMahasiswa21 next) {
        this.data = data;
        this.next = next;
    }

}
```

c. Hasil penulisan kode program untuk class SingleLinkList21.java

```
package Praktikum_ASD.Jobsheet11;

public class SingleLinkedList21 {

    NodeMahasiswa21 head;
    NodeMahasiswa21 tail;

    boolean isEmpty() {
        return (head == null);
    }

    public void print() {
        if(!isEmpty()) {
            NodeMahasiswa21 tmp = head;
            System.out.println("Isi Linked List:");
            while (tmp != null) {
                tmp.data.tampilInformasi();
                tmp = tmp.next;
            }
            System.out.println("");
        } else {
            System.out.println("Linked List Kosong");
        }
    }

    public void addFirst(Mahasiswa21 input) {
        NodeMahasiswa21 ndInput = new NodeMahasiswa21(input, null);
        if(isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            ndInput.next = head;
            head = ndInput;
        }
    }

    public void addLast(Mahasiswa21 input) {
        NodeMahasiswa21 ndInput = new NodeMahasiswa21(input, null);
        if(isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            tail.next = ndInput;
            tail = ndInput;
        }
    }
}
```

```

public void insertAfter(String key, Mahasiswa21 input) {
    NodeMahasiswa21 ndInput = new NodeMahasiswa21(input, null);
    NodeMahasiswa21 temp = head;

    do {
        if(temp.data.nama.equalsIgnoreCase(key)) {
            ndInput.next = temp.next;
            temp.next = ndInput;
            if(ndInput.next == null) {
                tail = ndInput;
            }
            break;
        }
        temp = temp.next;
    } while (temp != null);
}

public void insertAt(int index, Mahasiswa21 input) {
    if(index < 0) {
        System.out.println("Index Salah");
    } else if (index == 0) {
        addFirst(input);
    } else {
        NodeMahasiswa21 temp = head;
        for (int i = 0; i < index - 1; i++) {
            temp = temp.next;
        }
        temp.next = new NodeMahasiswa21(input, temp.next);
        if(temp.next.next == null) {
            tail = temp.next;
        }
    }
}
}

```

d. Hasil penulisan kode program untuk class SLLMain21.java

```
package Praktikum_ASD.Jobsheet11;

public class SLLMain21 {

    public static void main(String[] args) {

        SingleLinkedList21 sll = new SingleLinkedList21();

        Mahasiswa21 mhs1 = new Mahasiswa21("21212203", "Dirga", "4D", 3.6);
        Mahasiswa21 mhs2 = new Mahasiswa21("24212200", "Alvaro", "1A", 4.0);
        Mahasiswa21 mhs3 = new Mahasiswa21("22212202", "Cintia", "3C", 3.5);
        Mahasiswa21 mhs4 = new Mahasiswa21("23212201", "Bimon", "2B", 3.8);

        sll.print();
        sll.addFirst(mhs1);
        sll.print();

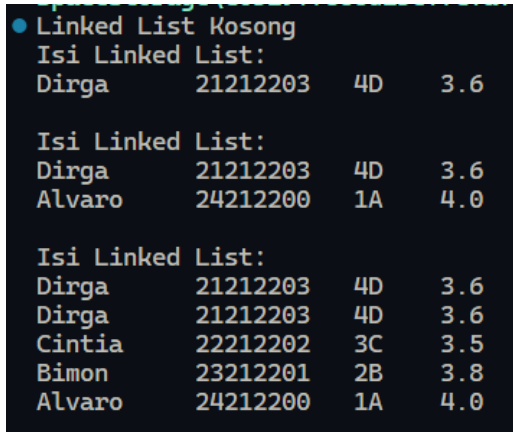
        sll.addLast(mhs2);
        sll.print();

        sll.insertAt(1, mhs3);
        sll.insertAt(2, mhs4);
        sll.print();

    }

}
```

e. Hasil run kode program



```
● Linked List Kosong
Isi Linked List:
Dirga      21212203    4D    3.6

Isi Linked List:
Dirga      21212203    4D    3.6
Alvaro     24212200    1A    4.0

Isi Linked List:
Dirga      21212203    4D    3.6
Dirga      21212203    4D    3.6
Cintia     22212202    3C    3.5
Bimon      23212201    2B    3.8
Alvaro     24212200    1A    4.0
```

Pertanyaan:

1. Mengapa hasil compile kode program di baris pertama menghasilkan “Linked List Kosong”?
2. Jelaskan kegunaan variable temp secara umum pada setiap method!
3. Lakukan modifikasi agar data dapat ditambahkan dari keyboard!

Jawab:

1. Karena pada main sistem setelah mengisi constructor, terdapat method print() yang dijalankan terlebih dahulu, dikarenakan kondisi isEmpty true atau head = null (data masih kosong) maka method print() akan mengembalikan nilai string yaitu “LinkedList kosong”
2. Pada method Print();
Tmp digunakan untuk menelusuri dan mencetak semua node dari awal (head) sampai akhir (tail), jika tidak menggunakan tmp atau head yang bergerak, maka data akan hilang.

Pada method insertAfter();

Temp digunakan untuk mencari nama Mhs yang sama dengan key, setelah sama maka akan menyisipkan node baru setelahnya.

Pada method insertAt();

Temp digunakan untuk menelusuri list dari head-tail sampai bertemu dengan posisi sebelum index yang dituju, setelah itu node baru akan disisipkan setelah temp.

3. Melakukan modifikasi pada main sistem agar data dapat di inputkan melalui keyboard serta dapat memilih method yang ingin digunakan

```
package Praktikum_ASD.Jobsheet11;
import java.util.Scanner;
public class SLLMain21 {

    static Scanner sc = new Scanner(System.in);

    public static Mahasiswa21 inputMahasiswa() {
        System.out.print("Masukkan Nama : ");
        String nama = sc.nextLine();
        System.out.print("Masukkan NIM : ");
        String nim = sc.nextLine();
        System.out.print("Masukkan Kelas : ");
        String kelas = sc.nextLine();
        System.out.print("Masukkan IPK : ");
        double ipk = sc.nextDouble();
        sc.nextLine();
        return new Mahasiswa21(nim, nama, kelas, ipk);
    }

    public static void main(String[] args) {
        SingleLinkedList21 list = new SingleLinkedList21();
        int pilihan;

        do {
            System.out.println("\n===== MENU LINKED LIST =====");
            System.out.println("1. Tambah data di awal (addFirst)");
            System.out.println("2. Tambah data di akhir (addLast)");
            System.out.println("3. Tambah data di index tertentu (insertAt)");
            System.out.println("4. Tambah data setelah nama tertentu (insertAfter)");
            System.out.println("5. Tampilkan data");
            System.out.println("0. Keluar");
            System.out.print("Pilih menu: ");
            pilihan = sc.nextInt();
            sc.nextLine();

            switch (pilihan) {
                case 1:
                    Mahasiswa21 m1 = inputMahasiswa();
                    list.addFirst(m1);
                    break;
                case 2:
                    Mahasiswa21 m2 = inputMahasiswa();
                    list.addLast(m2);
                    break;
                case 3:
                    System.out.print("Masukkan index: ");
                    int index = sc.nextInt();
                    sc.nextLine();
                    Mahasiswa21 m3 = inputMahasiswa();
                    list.insertAt(index, m3);
                    break;
            }
        } while (pilihan != 0);
    }
}
```

```
case 4:
    System.out.print("Masukkan nama setelah siapa akan disisipkan:");
    String key = sc.nextLine();
    Mahasiswa21 m4 = inputMahasiswa();
    list.insertAfter(key, m4);
    break;
case 5:
    System.out.println("Isi Linked List:");
    list.print();
    break;
case 0:
    System.out.println("Program selesai.");
    break;
default:
    System.out.println("Pilihan tidak valid!");
}
} while (pilihan != 0);

sc.close();
}
```


Percobaan 2: Modifikasi Elemen pada Single Linked List

```
public void getData(int index) {
    NodeMahasiswa21 tmp = head;
    for (int i = 0; i < index; i++) {
        tmp = tmp.next;
    }
    tmp.data.tampilInformasi();
}

public int indexOf(String key) {
    NodeMahasiswa21 tmp = head;
    int index = 0;
    while (tmp != null && !tmp.data.nama.equalsIgnoreCase(key)) {
        tmp = tmp.next;
        index++;
    }
    if(tmp == null) {
        return -1;
    } else {
        return index;
    }
}

public void removeFirst() {
    if(isEmpty()) {
        System.out.println("Linked List masih Kosong, tidak dapat dihapus!");
    } else if (head == tail) {
        head = tail = null;
    } else {
        head = head.next;
    }
}
```

```

public void removeLast() {
    if(isEmpty()) {
        System.out.println("Linked List masih Kosong, tidak dapat dihapus!");
    } else if (head == tail) {
        head = tail = null;
    } else {
        NodeMahasiswa21 temp = head;
        while (temp.next != tail) {
            temp = temp.next;
        }

        temp.next = null;
        tail = temp;
    }
}

public void remove(String key) {
    if(isEmpty()) {
        System.out.println("Linked List masih Kosong, tidak dapat dihapus!");
    } else {
        NodeMahasiswa21 temp = head;
        while(temp != null) {
            if((temp.data.nama.equalsIgnoreCase(key)) && (temp == head)) {
                this.removeFirst();
                break;
            } else if (temp.data.nama.equalsIgnoreCase(key)) {
                temp.next = temp.next.next;
                if(temp.next == null) {
                    tail = temp;
                }
                break;
            }
            temp = temp.next;
        }
    }
}

public void removeAt(int index) {
    if (index == 0) {
        removeFirst();
    } else {
        NodeMahasiswa21 temp = head;
        for (int i = 0; i < index - 1; i++) {
            temp = temp.next;
        }
    }
}
}

```

Hasil run kode program

```
===== MENU LINKED LIST =====
1. Tambah data di awal (addFirst)
2. Tambah data di akhir (addLast)
3. Tambah data di index tertentu (insertAt)
4. Tambah data setelah nama tertentu (insertAfter)
5. Tampilkan data (print)
6. Ambil data berdasarkan index (getData)
7. Cari index berdasarkan nama (indexOf)
8. Hapus data pertama (removeFirst)
9. Hapus data terakhir (removeLast)
10. Hapus data berdasarkan nama (remove)
11. Hapus data berdasarkan index (removeAt)
0. Keluar
Pilih menu: 5
Isi Linked List:
Isi Linked List:
Nico      145      1G      3.0
adi       123      1E      4.0
Luhung    124      1B      3.0
```

```
===== MENU LINKED LIST =====
1. Tambah data di awal (addFirst)
2. Tambah data di akhir (addLast)
3. Tambah data di index tertentu (insertAt)
4. Tambah data setelah nama tertentu (insertAfter)
5. Tampilkan data (print)
6. Ambil data berdasarkan index (getData)
7. Cari index berdasarkan nama (indexOf)
8. Hapus data pertama (removeFirst)
9. Hapus data terakhir (removeLast)
10. Hapus data berdasarkan nama (remove)
11. Hapus data berdasarkan index (removeAt)
0. Keluar
Pilih menu: 6
Masukkan index yang ingin diambil: 1
adi       123      1E      4.0
```

```
===== MENU LINKED LIST =====
1. Tambah data di awal (addFirst)
2. Tambah data di akhir (addLast)
3. Tambah data di index tertentu (insertAt)
4. Tambah data setelah nama tertentu (insertAfter)
5. Tampilkan data (print)
6. Ambil data berdasarkan index (getData)
7. Cari index berdasarkan nama (indexOf)
8. Hapus data pertama (removeFirst)
9. Hapus data terakhir (removeLast)
10. Hapus data berdasarkan nama (remove)
11. Hapus data berdasarkan index (removeAt)
0. Keluar
Pilih menu: 7
Masukkan nama mahasiswa yang dicari: luhung
Index: 2
```

```
===== MENU LINKED LIST =====
1. Tambah data di awal (addFirst)
2. Tambah data di akhir (addLast)
3. Tambah data di index tertentu (insertAt)
4. Tambah data setelah nama tertentu (insertAfter)
5. Tampilkan data (print)
6. Ambil data berdasarkan index (getData)
7. Cari index berdasarkan nama (indexOf)
8. Hapus data pertama (removeFirst)
9. Hapus data terakhir (removeLast)
10. Hapus data berdasarkan nama (remove)
11. Hapus data berdasarkan index (removeAt)
0. Keluar
Pilih menu: 8
```

```
===== MENU LINKED LIST =====
1. Tambah data di awal (addFirst)
2. Tambah data di akhir (addLast)
3. Tambah data di index tertentu (insertAt)
4. Tambah data setelah nama tertentu (insertAfter)
5. Tampilkan data (print)
6. Ambil data berdasarkan index (getData)
7. Cari index berdasarkan nama (indexOf)
8. Hapus data pertama (removeFirst)
9. Hapus data terakhir (removeLast)
10. Hapus data berdasarkan nama (remove)
11. Hapus data berdasarkan index (removeAt)
0. Keluar
Pilih menu: 11
Masukkan index data yang ingin dihapus: 0
```

```
===== MENU LINKED LIST =====
1. Tambah data di awal (addFirst)
2. Tambah data di akhir (addLast)
3. Tambah data di index tertentu (insertAt)
4. Tambah data setelah nama tertentu (insertAfter)
5. Tampilkan data (print)
6. Ambil data berdasarkan index (getData)
7. Cari index berdasarkan nama (indexOf)
8. Hapus data pertama (removeFirst)
9. Hapus data terakhir (removeLast)
10. Hapus data berdasarkan nama (remove)
11. Hapus data berdasarkan index (removeAt)
0. Keluar
Pilih menu: 9
```

```
===== MENU LINKED LIST =====
1. Tambah data di awal (addFirst)
2. Tambah data di akhir (addLast)
3. Tambah data di index tertentu (insertAt)
4. Tambah data setelah nama tertentu (insertAfter)
5. Tampilkan data (print)
6. Ambil data berdasarkan index (getData)
7. Cari index berdasarkan nama (indexOf)
8. Hapus data pertama (removeFirst)
9. Hapus data terakhir (removeLast)
10. Hapus data berdasarkan nama (remove)
11. Hapus data berdasarkan index (removeAt)
0. Keluar
Pilih menu: 5
Isi Linked List:
Isi Linked List:
adi       123      1E      4.0
```

```
===== MENU LINKED LIST =====
1. Tambah data di awal (addFirst)
2. Tambah data di akhir (addLast)
3. Tambah data di index tertentu (insertAt)
4. Tambah data setelah nama tertentu (insertAfter)
5. Tampilkan data (print)
6. Ambil data berdasarkan index (getData)
7. Cari index berdasarkan nama (indexOf)
8. Hapus data pertama (removeFirst)
9. Hapus data terakhir (removeLast)
10. Hapus data berdasarkan nama (remove)
11. Hapus data berdasarkan index (removeAt)
0. Keluar
Pilih menu: 5
Isi Linked List:
Linked List Kosong
```

Pertanyaan:

1. Mengapa digunakan keyword break pada fungsi remove? Jelaskan!
2. Jelaskan kegunaan kode dibawah pada method remove

Jawab:

1. Fungsi break digunakan untuk menghentikan proses perulangan pada saat data sudah ditemukan dan dihapus, tanpa break perulangan akan terus berjalan sampai akhir (tail) padahal data sudah ditemukan.
2. Kode tersebut digunakan untuk menghapus node yang sekarang (temp) dengan cara melompati kode tersebut, lalu mengecek apakah node yang baru saja disambungkan adalah null, jika null maka node saat ini menjadi tail yang baru.

Tugas:

a. Hasil penulisan kode program class Mahasiswa21.java

```
package Praktikum_ASD.Jobsheet11.Percobaan;

public class Mahasiswa21 {

    String nim, nama, kelas;
    double ipk;

    Mahasiswa21(String nim, String nama, String kelas, double ipk) {
        this.nim = nim;
        this.nama = nama;
        this.kelas = kelas;
        this.ipk = ipk;
    }

    void tampilInformasi() {
        System.out.printf("%-10s %-10s %-5s %.1f\n", nama, nim, kelas, ipk);
    }

}
```

b. Hasil penulisan kode program class NodeMahasiswa21.java

```
package Praktikum_ASD.Jobsheet11.Percobaan;

public class NodeMahasiswa21 {

    Mahasiswa21 data;
    NodeMahasiswa21 next;

    public NodeMahasiswa21(Mahasiswa21 data, NodeMahasiswa21 next) {
        this.data = data;
        this.next = next;
    }

}
```

c. Hasil penulisan kode program class QueueAntrian21.java

```
package Praktikum_ASD.Jobsheet11.Tugas;

public class QueueAntrian21 {

    NodeMahasiswa21 front;
    NodeMahasiswa21 rear;
    int size = 0;
    int max = 10;

    public boolean isEmpty() {
        return front == null;
    }

    public boolean isFull() {
        return size == max;
    }

    public void enqueue(Mahasiswa21 mhs) {
        if(isFull()) {
            System.out.println("Antrian sudah penuh");
        } else {
            NodeMahasiswa21 data = new NodeMahasiswa21(mhs, null);
            if (front == null) {
                front = rear = data;
            } else {
                rear.next = data;
                rear = data;
            }
            size++;
            System.out.println("Mahasiswa berhasil ditambahkan ke antrian.");
        }
    }

    public void dequeue() {
        if(isEmpty()) {
            System.out.println("Antrian Kosong");
        } else {
            System.out.println("Memanggil Mahasiswa: ");
            front.data.tampilInformasi();
            front = front.next;
            size--;
            if(front == null) {
                rear = null;
            }
        }
    }
}
```

```

public void peekFront() {
    if(isEmpty()) {
        System.out.println("Antrian Kosong");
    } else {
        System.out.println("Antrian Terdepan: ");
        front.data.tampilInformasi();
    }
}

public void peekrear() {
    if(isEmpty()) {
        System.out.println("Antrian Kosong");
    } else {
        System.out.println("Antrian Paling Akhir: ");
        rear.data.tampilInformasi();
    }
}

public void clear() {
    front = rear = null;
    size = 0;
    System.out.println("Antrian Dikosongkan");
}

public void print() {
    if (isEmpty()) {
        System.out.println("Antrian Kosong");
    } else {
        System.out.println("Daftar Mahasiswa Dalam Antrian: ");
        NodeMahasiswa21 temp = front;
        while (temp != null) {
            temp.data.tampilInformasi();
            temp = temp.next;
        }
    }
}

public int getSize() {
    return size;
}
}

```

d. Hasil penulisan kode program MainQueue21.java

```
package Praktikum_ASD.Jobsheet11.Tugas;

import java.util.Scanner;

public class MainQueue21 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        QueueAntrian21 antrian = new QueueAntrian21();
        int pilih;

        do {
            System.out.println("\n===== MENU ANTRIAN UNIT KEMAHASISWAAN =====");
            System.out.println("1. Tambah Antrian");
            System.out.println("2. Panggil Antrian");
            System.out.println("3. Lihat Antrian Terdepan");
            System.out.println("4. Lihat Antrian Paling Akhir");
            System.out.println("5. Tampilkan Semua Antrian");
            System.out.println("6. Tampilkan Jumlah Antrian");
            System.out.println("7. Kosongkan Antrian");
            System.out.println("0. Keluar");
            System.out.print("Pilih menu: ");
            pilih = sc.nextInt(); sc.nextLine();

            switch (pilih) {
                case 1:
                    System.out.print("Masukkan NIM: ");
                    String nim = sc.nextLine();
                    System.out.print("Masukkan Nama: ");
                    String nama = sc.nextLine();
                    System.out.print("Masukkan Kelas: ");
                    String kelas = sc.nextLine();
                    System.out.print("Masukkan IPK: ");
                    double ipk = sc.nextDouble();
                    sc.nextLine();
                    Mahasiswa21 mhs = new Mahasiswa21(nim, nama, kelas, ipk);
                    antrian.enqueue(mhs);
                    break;

                case 2:
                    antrian.dequeue();
                    break;

                case 3:
                    antrian.peekFront();
                    break;
            }
        } while (pilih != 0);
    }
}
```



```
case 4:
    antrian.peekrear();
    break;

case 5:
    antrian.print();
    break;

case 6:
    System.out.println("Jumlah mahasiswa dalam antrian: " +
antrian.getSize());
    break;

case 7:
    antrian.clear();
    break;

case 0:
    System.out.println("Terima kasih. Keluar dari program.");
    break;

default:
    System.out.println("Pilihan tidak valid!");
}
} while (pilih != 0);
sc.close();
}
```

e. Hasil run kode program

```
===== MENU ANTRIAN UNIT KEMAHASISWAAN =====
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Terdepan
4. Lihat Antrian Paling Akhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 1
Masukkan NIM: 123
Masukkan Nama: Adi
Masukkan Kelas: 1E
Masukkan IPK: 4
Mahasiswa berhasil ditambahkan ke antrian.

===== MENU ANTRIAN UNIT KEMAHASISWAAN =====
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Terdepan
4. Lihat Antrian Paling Akhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 1
Masukkan NIM: 134
Masukkan Nama: Budi
Masukkan Kelas: 1A
Masukkan IPK: 3
Mahasiswa berhasil ditambahkan ke antrian.

===== MENU ANTRIAN UNIT KEMAHASISWAAN =====
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Terdepan
4. Lihat Antrian Paling Akhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 5
Daftar Mahasiswa Dalam Antrian:
Adi      123      1E      4.0
Budi     134      1A      3.0

===== MENU ANTRIAN UNIT KEMAHASISWAAN =====
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Terdepan
4. Lihat Antrian Paling Akhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 6
Jumlah mahasiswa dalam antrian: 2
```

```
===== MENU ANTRIAN UNIT KEMAHASISWAAN =====
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Terdepan
4. Lihat Antrian Paling Akhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 3
Antrian Terdepan:
Adi      123      1E      4.0

===== MENU ANTRIAN UNIT KEMAHASISWAAN =====
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Terdepan
4. Lihat Antrian Paling Akhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 4
Antrian Paling Akhir:
Budi     134      1A      3.0

===== MENU ANTRIAN UNIT KEMAHASISWAAN =====
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Terdepan
4. Lihat Antrian Paling Akhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 2
Memanggil Mahasiswa:
Adi      123      1E      4.0

===== MENU ANTRIAN UNIT KEMAHASISWAAN =====
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Terdepan
4. Lihat Antrian Paling Akhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 7
Antrian Dikongkan
```

```
===== MENU ANTRIAN UNIT KEMAHASISWAAN =====
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Terdepan
4. Lihat Antrian Paling Akhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 5
Antrian Kosong

===== MENU ANTRIAN UNIT KEMAHASISWAAN =====
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Terdepan
4. Lihat Antrian Paling Akhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Kosongkan Antrian
0. Keluar
Pilih menu: 0
Terima kasih. Keluar dari program.
pwsh main # ?3
09:15:36 | 24 May, Saturday | in
```