

LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
JOBSHEET 11



ATHAULLA HAFIZH

244107020030

TI 1 E

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2024

2. Praktikum

Percobaan 1

Kode program class Mahasiswa05

```
package Jobsheet11;

public class Mahasiswa05 {
    String nim, nama, kelas;
    double ipk;

    public Mahasiswa05() {
    }

    public Mahasiswa05(String nm, String name, String kls, double
ip) {
        nim = nm;
        nama = name;
        kelas = kls;
        ipk = ip;
    }

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

Kode program class NodeMahasiswa05

```
package Jobsheet11;

public class NodeMahasiswa05 {
    Mahasiswa05 data;
    NodeMahasiswa05 next;

    public NodeMahasiswa05(Mahasiswa05 data, NodeMahasiswa05 next) {
        this.data = data;
        this.next = next;
    }
}
```

```
}  
}
```

Kode program class SingleLinkedList05

```
package Jobsheet11;  
  
public class SingleLinkedList05 {  
    NodeMahasiswa05 head;  
    NodeMahasiswa05 tail;  
  
    boolean isEmpty() {  
        return (head == null);  
    }  
  
    public void print() {  
        if (!isEmpty()) {  
            NodeMahasiswa05 tmp = head;  
            System.out.print("Isi Linked List:\t");  
            while (tmp != null) {  
                tmp.data.tampilInformasi();  
                tmp = tmp.next;  
            }  
            System.out.println("");  
        } else {  
            System.out.println("Linked List kosong");  
        }  
        System.out.println();  
    }  
  
    public void addFirst(Mahasiswa05 input) {  
        NodeMahasiswa05 ndInput = new NodeMahasiswa05(input, null);  
        if (isEmpty()) {  
            head = ndInput;  
            tail = ndInput;  
        } else {  
            ndInput.next = head;  
            head = ndInput;  
        }  
    }  
}
```

```

        head = ndInput;
    }
}

public void addLast(Mahasiswa05 input) {
    NodeMahasiswa05 ndInput = new NodeMahasiswa05(input, null);
    if (isEmpty()) {
        head = ndInput;
        tail = ndInput;
    } else {
        tail.next = ndInput;
        tail = ndInput;
    }
}

public void insertAfter(String key, Mahasiswa05 input) {
    NodeMahasiswa05 ndInput = new NodeMahasiswa05(input, null);
    NodeMahasiswa05 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, Mahasiswa05 input) {
    if (index < 0) {
        System.out.println("Indeks salah");
    } else if (index == 0) {
        addFirst(input);
    }
}

```

```

        } else {
            NodeMahasiswa05 temp = head;
            for (int i = 0; i < index - 1; i++) {
                temp = temp.next;
            }
            temp.next = new NodeMahasiswa05(input, temp.next);
            if (temp.next.next == null) {
                tail = temp.next;
            }
        }
    }
}

```

Kode program class SLLMain05

```

package Jobsheet11;

public class SLLMain05 {

    public static void main(String[] args) {

        SingleLinkedList05 sll = new SingleLinkedList05();

        Mahasiswa05 mhs1 = new Mahasiswa05("21212200", "Alvaro",
"1A", 4.0);
        Mahasiswa05 mhs2 = new Mahasiswa05("21212201", "Bimon",
"2B", 3.8);
        Mahasiswa05 mhs3 = new Mahasiswa05("21212202", "Cintia",
"3C", 3.5);
        Mahasiswa05 mhs4 = new Mahasiswa05("21212203", "Dirga",
"4D", 3.6);

        sll.print();
        sll.addFirst(mhs4);
        sll.print();
        sll.addLast(mhs1);
        sll.print();
        sll.insertAfter("Dirga", mhs3);
        sll.insertAt(2, mhs2);
        sll.print();
    }
}

```

```
}  
}
```

Output

```
Linked List kosong  
  
Isi Linked List:  
Dirga      21212203    4D      3,6  
  
Isi Linked List:  
Dirga      21212203    4D      3,6  
Alvaro     21212200    1A      4,0  
  
Isi Linked List:  
Dirga      21212203    4D      3,6  
Cintia     21212202    3C      3,5  
Bimon      21212201    2B      3,8  
Alvaro     21212200    1A      4,0
```

Pertanyaan

1. Karena memanggil method print dulu sedangkan belum ada data yang ditambahkan.
2. Variable temp (temporary) memiliki kegunaan umum sebagai berikut:
 - **Sebagai pointer traversal:** Digunakan untuk berpindah dari satu node ke node berikutnya dalam linked list
 - **Menyimpan referensi sementara:** Menyimpan alamat node yang sedang diproses agar tidak hilang
 - **Navigasi linked list:** Membantu dalam proses penelusuran (traversing) dari head hingga tail
 - **Operasi insert/delete:** Membantu dalam operasi penambahan atau penghapusan node dengan menyimpan posisi node yang tepat
 - **Pencarian data:** Digunakan untuk mencari node dengan data tertentu dengan cara berpindah dari node satu ke node lainnya
3. Modifikasi kode agar data dapat ditambahkan dari keyboard

```
int jml = 100;  
for (int i = 0; i < jml ; i++) {  
    System.out.println("Masukkan data mahasiswa ke- " + (i +  
1) + " : ");
```

```

        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();

        Mahasiswa05 mhs = new Mahasiswa05(nim, nama, kelas,
ipk);

        sll.addFirst(mhs);
        System.out.println();
        System.out.print("Apakah mau lanjut? (y/n) : ");
        String jawab = sc.nextLine();
        if (jawab.equalsIgnoreCase("n")) {
            System.out.println("Terima kasih.");
            break;
        }
    }
}

```

Percobaan 2

Penambahan kode program class SingleLinkedList05

```

    public void getData(int index) {
        NodeMahasiswa05 tmp = head;
        for (int i = 0; i < index; i++) {
            tmp = tmp.next;
        }
        tmp.data.tampilInformasi();
    }

    public int indexOf(String key) {
        NodeMahasiswa05 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 {
        NodeMahasiswa05 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 {
        NodeMahasiswa05 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 {
        NodeMahasiswa05 temp = head;
        for (int i = 0; i < index - 1; i++) {
            temp = temp.next;
        }
        temp.next = temp.next.next;
    }
}

```

```

        if (temp.next == null) {
            tail = temp;
        }
    }
}

```

Penambahan kode program class SSLMain

```

        System.out.println("data index 1 :");
        sll.getData(1);
        System.out.println();

        System.out.println("data mahasiswa an Bimon berada pada
index : " + sll.indexOf("bimon"));
        System.out.println();

        sll.removeFirst();
        sll.removeLast();
        sll.print();
        sll.removeAt(0);
        sll.print();

```

Output

```

data index 1 :

Cintia      22212202   3C          3,5
data mahasiswa an Bimon berada pada index : 2

Isi Linked List:
Cintia      22212202   3C          3,5
Bimon       23212201   2B          3,8

Isi Linked List:
Bimon       23212201   2B          3,8

```

Pertanyaan

1. Keyword break digunakan pada fungsi remove karena:
 - **Menghentikan loop**
 - **Mencegah error:** Menghindari kemungkinan penghapusan multiple node atau operasi pada node yang sudah tidak valid
2. **Update pointer:** Mengubah referensi node sebelumnya agar menunjuk ke node setelah node yang dihapus
Kemudian, jika referensi node yang ditunjuk berisi null maka node (temp) tersebut merupakan tail.

Tugas

Kode program class Mahasiswa

```
package Jobsheet11;

public class Mahasiswa {
    String nama, nim, kelas, prodi;

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

    public void tampilInformasi() {
        System.out.printf("Nama: %s\n", nama);
        System.out.printf("NIM: %s\n", nim);
        System.out.printf("Prodi: %s\n", prodi);
        System.out.printf("Kelas: %s\n", kelas);
        System.out.println();
    }
}
```

Kode program class NodeMhs

```
package Jobsheet11;

public class NodeMhs {

    Mahasiswa data;

    NodeMhs next;

    public NodeMhs(Mahasiswa data, NodeMhs next){

        this.data = data;

        this.next = next;

    }

}
```

Kode program class LinkedListLayananUnit

```
package Jobsheet11;

public class LinkedListLayananUnit {

    NodeMhs head;

    NodeMhs tail;

    int maxSize = 100;

    int size = 0;

    public boolean isEmpty() {

        return (head == null);

    }

    public boolean cekAntrianPenuh() {

        return size >= maxSize;

    }

    public void kosongkan() {

        head = tail = null;

        size = 0;

        System.out.println("Antrian telah dikosongkan.");

    }

}
```

```

    public void daftarkanData(Mahasiswa input) {
        if (cekAntrianPenuh()) {
            System.out.println("Antrian penuh! Tidak dapat
menambahkan mahasiswa baru.");
            return;
        }

        NodeMhs ndInput = new NodeMhs(input, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            tail.next = ndInput;
            tail = ndInput;
        }
        size++;
        System.out.println(input.nama + " berhasil masuk ke
antrian.");
    }

    public void memanggilAntrian() {
        if (isEmpty()) {
            System.out.println("Antrian kosong! Tidak ada yang bisa
dipanggil.");
        } else {
            System.out.println(head.data.nama + " dipanggil untuk
dilayani.");
            if (head == tail) {
                head = tail = null;
            } else {
                head = head.next;
            }
            size--;
        }
    }

    public void tampilAntrianDepanTerakhir() {

```

```

        if (isEmpty()) {
            System.out.println("Antrian kosong! Tidak ada data yang
dapat ditampilkan.");
        } else {
            System.out.println("=== ANTRIAN TERDEPAN ===");
            head.data.tampilInformasi();

            if (head != tail) {
                System.out.println("=== ANTRIAN TERAKHIR ===");
                tail.data.tampilInformasi();
            } else {
                System.out.println("Hanya ada satu mahasiswa dalam
antrian.");
            }
        }
    }

    public void tampilMahasiswaYangMengantre() {
        System.out.println("Jumlah mahasiswa yang masih mengantre: "
+ size);

        if (!isEmpty()) {
            System.out.println("\nDaftar mahasiswa yang
mengantre:");
            NodeMhs temp = head;
            int nomor = 1;
            while (temp != null) {
                System.out.println(nomor + ". " + temp.data.nama + "
(" + temp.data.nim + ")");
                temp = temp.next;
                nomor++;
            }
        }
    }
}

```

Kode program class LayananUnitKemahasiswaan

```
package Jobsheet11;

import java.util.Scanner;

public class LayananUnitKemahasiswaan {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        LinkedListLayananUnit antrian = new LinkedListLayananUnit();

        int pilihan;

        System.out.println("=====
=====");

        System.out.println("        SISTEM ANTRIAN LAYANAN UNIT
KEMAHASISWAAN");

        System.out.println("=====
=====");

        do {

            System.out.println("\n=== MENU ANTRIAN LAYANAN UNIT
KEMAHASISWAAN ===");

            System.out.println("1. Cek Antrian Kosong");
            System.out.println("2. Cek Antrian Penuh");
            System.out.println("3. Kosongkan Antrian");
            System.out.println("4. Tambah Antrian");
            System.out.println("5. Panggil Antrian");
            System.out.println("6. Lihat Antrian Terdepan Dan Paling
Akhir");

            System.out.println("7. Lihat Jumlah Antrian Mahasiswa");
            System.out.println("0. Keluar");

            System.out.println("=====
=====");

            System.out.print("Pilih menu: ");

            pilihan = sc.nextInt();

            switch (pilihan) {

                case 1:

                    if (antrian.isEmpty()) {
```

```

        System.out.println("Antrian kosong.");
    } else {
        System.out.println("Antrian tidak kosong.");
    }
    break;

case 2:
    if (antrian.cekAntrianPenuh()) {
        System.out.println("Antrian penuh.");
    } else {
        System.out.println("Antrian belum penuh.");
    }
    break;

case 3:
    if (antrian.isEmpty()) {
        System.out.println("Antrian sudah kosong.");
    } else {
        System.out.print("Apakah Anda yakin ingin
mengosongkan antrian? (y/n): ");
        sc.nextLine();
        String konfirmasi = sc.nextLine();
        if (konfirmasi.equalsIgnoreCase("y") ||
konfirmasi.equalsIgnoreCase("yes")) {
            antrian.kosongkan();
        } else {
            System.out.println("Pembatalan
mengosongkan antrian.");
        }
    }
    break;

case 4: {
    if (antrian.cekAntrianPenuh()) {
        System.out.println("Antrian penuh! Tidak
dapat menambahkan mahasiswa baru.");
    }
    break;
}

```



```

    }

    sc.nextLine();
    System.out.println("\n=== PENDAFTARAN ANTRIAN
===");

    System.out.print("Nama: ");
    String nama = sc.nextLine();
    System.out.print("NIM: ");
    String nim = sc.nextLine();
    System.out.print("Prodi: ");
    String prodi = sc.nextLine();
    System.out.print("Kelas: ");
    String kelas = sc.nextLine();

    Mahasiswa mhs = new Mahasiswa(nim, nama, prodi,
kelas);

    antrian.daftarkanData(mhs);
    break;
}

case 5:
    antrian.memanggilAntrian();
    break;

case 6:
    antrian.tampilAntrianDepanTerakhir();
    break;

case 7:
    antrian.tampilMahasiswaYangMengantre();
    break;

case 0:
    System.out.println("Terima kasih telah
menggunakan Sistem Layanan Unit Kemahasiswaan!");
    break;

```

```

                default:
                    System.out.println("Pilihan tidak valid. Silakan
pilih menu 0-7.");
                }
            } while (pilihan != 0);
        }
    }
}

```

Output

```

=====
      SISTEM ANTRIAN LAYANAN UNIT KEMAHASISWAAN
=====

=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN ===
1. Cek Antrian Kosong
2. Cek Antrian Penuh
3. Kosongkan Antrian
4. Tambah Antrian
5. Panggil Antrian
6. Lihat Antrian Terdepan Dan Paling Akhir
7. Lihat Jumlah Antrian Mahasiswa
0. Keluar
=====
Pilih menu: 4

=== PENDAFTARAN ANTRIAN ===
Nama: Athaulla
NIM: 1231312
Prodi: Ti
Kelas: 1E
Athaulla berhasil masuk ke antrian.

```

```

Pilih menu: 4

=== PENDAFTARAN ANTRIAN ===
Nama: Hafizh
NIM: 123131
Prodi: TI
Kelas: 1E
Hafizh berhasil masuk ke antrian.

```

Pilih menu: 4

=== PENDAFTARAN ANTRIAN ===

Nama: Raihan

NIM: 1231231451

Prodi: TI

Kelas: 1D

Raihan berhasil masuk ke antrian.

Pilih menu: 4

=== PENDAFTARAN ANTRIAN ===

Nama: Syarif

NIM: 1239140

Prodi: TI

Kelas: 1E

Syarif berhasil masuk ke antrian.

=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN ===

1. Cek Antrian Kosong
2. Cek Antrian Penuh
3. Kosongkan Antrian
4. Tambah Antrian
5. Panggil Antrian
6. Lihat Antrian Terdepan Dan Paling Akhir
7. Lihat Jumlah Antrian Mahasiswa
0. Keluar

=====

Pilih menu: 1

Antrian tidak kosong.

=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN ===

1. Cek Antrian Kosong
2. Cek Antrian Penuh
3. Kosongkan Antrian
4. Tambah Antrian
5. Panggil Antrian
6. Lihat Antrian Terdepan Dan Paling Akhir
7. Lihat Jumlah Antrian Mahasiswa
0. Keluar

=====

Pilih menu: 2

Antrian belum penuh.

```
=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN ===
1. Cek Antrian Kosong
2. Cek Antrian Penuh
3. Kosongkan Antrian
4. Tambah Antrian
5. Panggil Antrian
6. Lihat Antrian Terdepan Dan Paling Akhir
7. Lihat Jumlah Antrian Mahasiswa
0. Keluar
=====
Pilih menu: 5
Athaulla dipanggil untuk dilayani.

=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN ===
1. Cek Antrian Kosong
2. Cek Antrian Penuh
3. Kosongkan Antrian
4. Tambah Antrian
5. Panggil Antrian
6. Lihat Antrian Terdepan Dan Paling Akhir
7. Lihat Jumlah Antrian Mahasiswa
0. Keluar
=====
Pilih menu: 6
=== ANTRIAN TERDEPAN ===
Nama: Hafizh
NIM: 123131
Prodi: TI
Kelas: 1E

=== ANTRIAN TERAKHIR ===
Nama: Syarif
NIM: 1239140
Prodi: TI
Kelas: 1E
```

```
Pilih menu: 7
Jumlah mahasiswa yang masih mengantre: 3

Daftar mahasiswa yang mengantre:
1. Hafizh (123131)
2. Raihan (1231231451)
3. Syarif (1239140)
```

```
Pilih menu: 3
Apakah Anda yakin ingin mengosongkan antrian? (y/n): n
Pembatalan mengosongkan antrian.

=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN ===
1. Cek Antrian Kosong
2. Cek Antrian Penuh
3. Kosongkan Antrian
4. Tambah Antrian
5. Panggil Antrian
6. Lihat Antrian Terdepan Dan Paling Akhir
7. Lihat Jumlah Antrian Mahasiswa
0. Keluar
=====
Pilih menu: 7
Jumlah mahasiswa yang masih mengantre: 3

Daftar mahasiswa yang mengantre:
1. Hafizh (123131)
2. Raihan (1231231451)
3. Syarif (1239140)
```

```
Pilih menu: 3
Apakah Anda yakin ingin mengosongkan antrian? (y/n): y
Antrian telah dikosongkan.

=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN ===
1. Cek Antrian Kosong
2. Cek Antrian Penuh
3. Kosongkan Antrian
4. Tambah Antrian
5. Panggil Antrian
6. Lihat Antrian Terdepan Dan Paling Akhir
7. Lihat Jumlah Antrian Mahasiswa
0. Keluar
=====
Pilih menu: 1
Antrian kosong.

=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN ===
1. Cek Antrian Kosong
2. Cek Antrian Penuh
3. Kosongkan Antrian
4. Tambah Antrian
5. Panggil Antrian
6. Lihat Antrian Terdepan Dan Paling Akhir
7. Lihat Jumlah Antrian Mahasiswa
0. Keluar
=====
Pilih menu: 0
Terima kasih telah menggunakan Sistem Layanan Unit Kemahasiswaan!
```