

MODUL I

LINKED LIST

1.1 PERMASALAHAN

1.1.1 List Manusia Viral

Kak Gem ingin dibuatkan list manusia viral. harus ada method ADDFIRST, ADDLAST, DELETEDFIRST, DELETEDLAST dan method untuk menampilkan manusia-manusia itu. Buat list manusia viral dengan parameter, nama manusia, skill, umur, dan hobi. Pertama-tama tambahkan vadel dalam list, diikuti dengan loli dan agus serta fufu fafa. kemudian tambah kak gem menggunakan method addfirst. setelah itu hapus fufu fafa dari list dengan method deletelast. terakhir, hapus kak gem dengan method deletefirst dan tampilkan list manusia-manusia yang tersisa.

1.2 HASIL PERCOBAAN

1.2.1 List Manusia Viral

1. Algoritma

- Definisikan struktur node buat kelas manusia yang menyimpan data.
- Definisikan linked list buat kelas linkedlist dengan variabel head yang menunjuk ke node pertama dalam daftar.
- Fungsi Untuk menambahkan node di awal daftar
- Fungsi untuk menambahkan node di akhir daftar
- Fungsi Untuk menghapus node pertama, jika head tidak kosong, pindahkan head ke node berikutnya
- Fungsi jika ingin menghapus node terakhir, periksa apakah daftar kosong.
- Fungsi untuk menampilkan isi daftar
- Main program buat instance dari linked list. Tambahkan beberapa node menggunakan metode tambah node di akhir, lalu tampilkan daftar sebelum penambahan. Tambahkan node baru di awal, tampilkan daftar setelah penambahan. Hapus node terakhir, hapus node pertama, lalu tampilkan daftar setelah penghapusan.

2. Source Code

```
class Manusia { //isi node manusia hebat
    String nama;
    String skill;
    int umur;
    String hobi;
    Manusia next;

    public Manusia(String nama, String skill, int umur, String
hobi) {
        this.nama = nama;
        this.skill = skill;
        this.umur = umur;
        this.hobi = hobi;
        this.next = null;
    }
}

class LinkedList { //linkedlist nya
    Manusia head;

    public void addFirst(String nama, String skill, int umur,
String hobi) { //metod add fisrst
        Manusia newManusia = new Manusia(nama, skill, umur,
hobi);
        newManusia.next = head;
        head = newManusia;
    }

    public void addLast(String nama, String skill, int umur,
String hobi) { //method add last
```

```

        Manusia newManusia = new Manusia(nama, skill, umur,
hobi);
        if (head == null) {
            head = newManusia;
        } else {
            Manusia temp = head;
            while (temp.next != null) {
                temp = temp.next;
            }
            temp.next = newManusia;
        }
    }

    public void deleteFirst() { //metod delete first
        if (head != null) {
            head = head.next;
        } else {
            System.out.println("List is empty.");
        }
    }

    public void deleteLast() { //metod delete last
        if (head == null) {
            System.out.println("List is empty.");
        } else if (head.next == null) {
            head = null;
        } else {
            Manusia temp = head;
            while (temp.next.next != null) {
                temp = temp.next;
            }
            temp.next = null;
        }
    }

    public void displayList() {
        if (head == null) {
            System.out.println("List is empty.");
        } else {
            Manusia temp = head;
            while (temp != null) {
                System.out.println("Nama: " + temp.nama + ",
Skill: " + temp.skill +
                                ", Umur: " + temp.umur + ",
Hobi: " + temp.hobi);
                temp = temp.next;
            }
        }
    }

    public class easy {
        public static void main(String[] args) {
            LinkedList list = new LinkedList();

            list.addLast("Vadel", "Dance Geterrrr", 19, "Dance");
            list.addLast("Loli", "ATM Berjalan", 16, "Liat Vadel");
            list.addLast("Agus", "Agus Sakit", 35, "Donasi");
            list.addLast("Fufu Fafa", "Roasting", 30, "Buka
Kaskus");
            System.out.println("List sebelum add first:");

```

```
list.displayList();

list.addFirst("Kak Gem", "Kasi Faham", 32, "Bernafas");

System.out.println("\nList setelah penambahan:");
list.displayList();

list.deleteLast();

list.deleteFirst();

System.out.println("\nList setelah penghapusan:");
list.displayList();
    }
}
```

1.3 ANALISIS DATA

1.3.1 Nama Program

```
class Manusia { //isi node manusia hebat
    String nama;
    String skill;
    int umur;
    String hobi;
    Manusia next;

    public Manusia(String nama, String skill, int umur, String hobi) {
        this.nama = nama;
        this.skill = skill;
        this.umur = umur;
        this.hobi = hobi;
        this.next = null;
    }
}
```

Kode ini merupakan kelas manusia yang mendefinisikan struktur data untuk setiap node dalam linked list. Di dalam kelas ini, terdapat beberapa atribut: nama, skill, umur, dan hobi, yang masing-masing menyimpan informasi terkait manusia. Atribut next adalah referensi ke node berikutnya dalam linked list, dan diinisialisasi dengan null di dalam konstruktor. Konstruktor ini digunakan untuk membuat objek manusia baru dengan nilai yang diberikan untuk setiap atribut.

```
class LinkedList { //linkedlist nya
    Manusia head;

    public void addFirst(String nama, String skill, int umur, String hobi) { //metod add firsrt
        Manusia newManusia = new Manusia(nama, skill, umur, hobi);
        newManusia.next = head;
        head = newManusia;
    }

    public void addLast(String nama, String skill, int umur, String hobi) { //method add last
        Manusia newManusia = new Manusia(nama, skill, umur, hobi);
        if (head == null) {
            head = newManusia;
        } else {
            Manusia temp = head;
            while (temp.next != null) {
                temp = temp.next;
            }
            temp.next = newManusia;
        }
    }

    public void deleteFirst() { //metod delete first
        if (head != null) {
            head = head.next;
        } else {
            System.out.println("List is empty.");
        }
    }
}
```

```

public void deleteLast() { //metod delete last
    if (head == null) {
        System.out.println("List is empty.");
    } else if (head.next == null) {
        head = null;
    } else {
        Manusia temp = head;
        while (temp.next.next != null) {
            temp = temp.next;
        }
        temp.next = null;
    }
}

public void displayList() {
    if (head == null) {
        System.out.println("List is empty.");
    } else {
        Manusia temp = head;
        while (temp != null) {
            System.out.println("Nama: " + temp.nama + ", Skill: " +
temp.skill +
                                ", Umur: " + temp.umur + ", Hobi: " +
temp.hobi);
            temp = temp.next;
        }
    }
}
}

```

Kode ini adalah kelas linkedlist yang berfungsi untuk mengelola linked list dari objek manusia. Di dalam kelas ini, atribut head menyimpan referensi ke node pertama dalam linked list. Terdapat beberapa metode untuk melakukan operasi dasar pada linked list metode menambahkan node baru di awal linked list, metode menambahkan node baru di akhir linked list, metode menghapus node pertama dari linked list, metode menghapus node terakhir dari linked list, metode ini menampilkan semua node dalam linked list.

```

public class easy {
    public static void main(String[] args) {
        LinkedList list = new LinkedList();

        list.addLast("Vadel", "Dance Geterrrr", 19, "Dance");
        list.addLast("Loli", "ATM Berjalan", 16, "Liat Vadel");
        list.addLast("Agus", "Agus Sakit", 35, "Donasi");
        list.addLast("Fufu Fafa", "Roasting", 30, "Buka Kaskus");
        System.out.println("List sebelum add first:");
        list.displayList();

        list.addFirst("Kak Gem", "Kasi Faham", 32, "Bernafas");

        System.out.println("\nList setelah penambahan:");
        list.displayList();

        list.deleteLast();

        list.deleteFirst();
    }
}

```

```
        System.out.println("\nList setelah penghapusan:");  
        list.displayList();  
    }  
}
```

Kode di atas adalah kelas `easy` yang berfungsi sebagai program utama untuk menjalankan kode. Dalam metode `main`, objek `linkedlist` dibuat untuk mengelola daftar manusia. Beberapa node ditambahkan menggunakan metode `addlast`, dan hasilnya ditampilkan sebelum dan setelah penambahan node pertama dengan menggunakan metode `displaylist`. Setelah itu, program menghapus node terakhir dan pertama menggunakan metode `deletelast` dan `deletefirst`, lalu menampilkan daftar lagi untuk menunjukkan hasil akhir setelah penghapusan.