MODUL I
LINKED LIST

1.1 PERMASALAHAN

1.1.1 *Single Linked list*

Buatkan list manusia viral, harus ada method *addfirst*, *addlast*, *deletefirst*, *deletelast* 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 *Single Linked List*

1. Algoritma
   a. Inisialisasi struktur data (linked list) dengan nama Easy yang pada awalnya tidak berisi elemen apa pun.
   b. Arahkan pengguna untuk mengisi informasi berupa nama, minat, keahlian, dan usia.
   c. Lakukan pembuatan objek Orang baru dengan menggunakan informasi yang telah dimasukkan.
   d. Tentukan apakah daftar tersebut tidak memiliki elemen di dalamnya dan masih dalam keadaan kosong.
   e. Jika daftar punya elemen, hapus orang pertama dan jadikan orang kedua sebagai yang pertama.
   f. Program akan terus berulang (*loop)* sampai pengguna memilih untuk keluar.
2. *Source Code*

```
class Easy {
    String nama;
    String skill;
    int umur;
    String hobi;
    Easy next;

    public Easy(String nama, String skill, int umur, String hobi) {
        this.nama = nama;
        this.skill = skill;
```

```
        this.umur = umur;
        this.hobi = hobi;
        this.next = null;
    }
}

class Viral {
    private Easy head;

    public Viral() {
        head = null;
    }

    public void addFirst(Easy newNode) {
        newNode.next = head;
        head = newNode;
    }

    public void addLast(Easy newNode) {
        if (head == null) {
            head = newNode;
        } else {
            Easy current = head;
            while (current.next != null) {
                current = current.next;
            }
            current.next = newNode;
        }
    }

    public void deleteFirst() {
        if (head == null) {
            System.out.println("Belum ada manusianya");
        } else {
            head = head.next;
        }
    }

    public void deleteLast() {
        if (head == null) {
            System.out.println("Belum ada manusianya");
            return;
        }

        if (head.next == null) {
            head = null;
        } else {
            Easy current = head;
            while (current.next.next != null) {
                current = current.next;
            }
            current.next = null;
        }
    }

    public void display() {
        if (head == null) {
```

```
            System.out.println("Belum ada manusianya");
            return;
        }

        Easy current = head;
        while (current != null) {
            System.out.println("Nama: " + current.nama);
            System.out.println("Skill: " + current.skill);
            System.out.println("Umur: " + current.umur);
            System.out.println("Hobi: " + current.hobi);
            System.out.println();
            current = current.next;
        }
    }

    public void addPerson(String nama, String skill, int umur, String
hobi, boolean isFirst) {
        Easy newNode = new Easy(nama, skill, umur, hobi);
        if (isFirst) {
            addFirst(newNode);
        } else {
            addLast(newNode);
        }
    }

    public static void main(String[] args) {
        Viral list = new Viral();

        list.addPerson("Vadel", "Dance Geterrr", 19, "Dance", false);
        list.addPerson("Loli", "ATM berjalan", 16, "Liat Vadel",
false);
        list.addPerson("Agus", "Agus sakit", 35, "Donasi", false);
        list.addPerson("Fufu Fafa", "Roasting", 30, "Buka kaskus",
false);
        list.addPerson("Kak Gem", "Kasih paham", 32, "Bernafas",
true);

        list.display();

        list.deleteLast();
        list.deleteFirst();

        System.out.println("\nSetelah menghapus Kak Gem dan Fufu
Fafa");
        System.out.println();
        list.display();
    }
}
```

1.3 Analisis Data

1.3.1 *Single Linked List*

```
class Easy {
    String nama;
    String skill;
    int umur;
    String hobi;
    Easy next;
```

kelas "`Easy`" merupakan struktur yang menyimpan informasi tentang seseorang, dengan atribut "`nama`", "`skill`", "`umur`", dan "`hobi`", semuanya bertipe "`String`" kecuali "`umur`" yang bertipe "`int`". Selain itu, terdapat atribut "`next`" yang juga bertipe "`Easy`", berfungsi sebagai referensi ke objek "`Easy`" berikutnya dalam linked list. Kelas ini memungkinkan pengorganisasian data orang secara berurutan.

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

Konstruktor "`Easy`" menginisialisasi objek baru dengan empat parameter: "`nama`", "`skill`", "`umur`", dan "`hobi`", yang disimpan dalam atribut yang sesuai. Atribut "`next`" diatur menjadi "`null`", menandakan bahwa objek ini belum terhubung dengan objek lain dalam linked list.

```
class Viral {
    private Easy head;

    public Viral() {
        head = null;
    }

    public void addFirst(Easy newNode) {
        newNode.next = head;
        head = newNode;
    }

    public void addLast(Easy newNode) {
        if (head == null) {
            head = newNode;
        } else {
            Easy current = head;
            while (current.next != null) {
                current = current.next;
            }
            current.next = newNode;
        }
```

```
    }
```

Kelas "`Viral`" mengelola linked list objek "`Easy`" dengan atribut "`head`" sebagai penunjuk ke elemen pertama. Konstruktor menginisialisasi "`head`" ke "`null`". Metode "`addFirst`" menambahkan elemen baru di awal, dan "`addLast`" menambahkannya di akhir, menghubungkan elemen baru ke elemen terakhir jika daftar tidak kosong. Kelas ini menyediakan fungsi dasar untuk linked list.

```
    public void deleteFirst() {
        if (head == null) {
            System.out.println("Belum ada manusianya");
        } else {
            head = head.next;
        }
    }

    public void deleteLast() {
        if (head == null) {
            System.out.println("Belum ada manusianya");
            return;
        }

        if (head.next == null) {
            head = null;
        } else {
            Easy current = head;
            while (current.next.next != null) {
                current = current.next;
            }
            current.next = null;
        }
    }
```

Metode "`deleteFirst`" dalam kelas "`Viral`" menghapus elemen pertama dari linked list. Jika "`head`" kosong, ia mencetak "Belum ada manusianya." Jika tidak, "`head`" diperbarui untuk menunjuk ke elemen kedua. Metode "`deleteLast`" menghapus elemen terakhir; jika daftar kosong, ia mencetak pesan yang sama. Jika hanya ada satu elemen, "`head`" diatur ke "`null`", dan jika ada lebih dari satu, metode ini menelusuri hingga elemen sebelum yang terakhir dan mengatur referensi "`next`" menjadi "`null`". Kedua metode ini memungkinkan penghapusan elemen dari awal atau akhir daftar.

```
public void display() {
        if (head == null) {
            System.out.println("Belum ada manusianya");
            return;
        }

        Easy current = head;
```

```
        while (current != null) {
            System.out.println("Nama: " + current.nama);
            System.out.println("Skill: " + current.skill);
            System.out.println("Umur: " + current.umur);
            System.out.println("Hobi: " + current.hobi);
            System.out.println();
            current = current.next;
        }
    }
```

Metode "`display`" dalam kelas "`Viral`" menampilkan informasi tentang semua elemen dalam linked list. Jika "`head`" kosong, metode ini mencetak "Belum ada manusianya" dan keluar. Jika tidak, metode ini menelusuri setiap elemen dari "`head`", mencetak atribut "`nama`", "`skill`", "`umur`", dan "`hobi`" untuk setiap objek "`Easy`". Setelah menampilkan satu elemen, metode ini melanjutkan ke elemen berikutnya hingga mencapai akhir daftar, sehingga memberikan tampilan terstruktur dari semua data yang tersimpan.

```
public void addPerson(String nama, String skill, int umur, String hobi,
boolean isFirst) {
        Easy newNode = new Easy(nama, skill, umur, hobi);
        if (isFirst) {
            addFirst(newNode);
        } else {
            addLast(newNode);
        }
    }
```

Metode "`addPerson`" dalam kelas "`Viral`" digunakan untuk menambahkan objek "`Easy`" ke dalam linked list. Metode ini menerima lima parameter: "`nama`", "`skill`", "`umur`", dan "`hobi`" untuk membuat objek baru, serta boolean "`isFirst`" yang menentukan posisi penambahan. Jika "`isFirst`" bernilai "`true`", objek baru ditambahkan di awal daftar menggunakan "`addFirst`"; jika tidak, objek ditambahkan di akhir dengan "`addLast`". Metode ini memberikan fleksibilitas dalam menambah elemen ke dalam linked list.

```
public static void main(String[] args) {
        Viral list = new Viral();

        list.addPerson("Vadel", "Dance Geterrr", 19, "Dance", false);
        list.addPerson("Loli", "ATM berjalan", 16, "Liat Vadel", false);
        list.addPerson("Agus", "Agus sakit", 35, "Donasi", false);
        list.addPerson("Fufu Fafa", "Roasting", 30, "Buka kaskus", false);
        list.addPerson("Kak Gem", "Kasih paham", 32, "Bernafas", true);

        list.display();

        list.deleteLast();
        list.deleteFirst();
```

```
        System.out.println("\nSetelah menghapus Kak Gem dan Fufu Fafa");
        System.out.println();
        list.display();
    }
}
```

Metode "`main`" dalam kelas ini berfungsi sebagai titik awal program. Di dalamnya, objek "`Viral`" bernama "`list`" dibuat untuk mengelola linked list. Beberapa orang ditambahkan ke daftar menggunakan metode "`addPerson`", dengan sebagian besar ditambahkan di akhir dan "Kak Gem" di awal. Setelah itu, metode "`display`" dipanggil untuk menampilkan informasi semua orang. Elemen terakhir dan pertama kemudian dihapus menggunakan "`deleteLast`" dan "`deleteFirst`". Program mencetak pesan bahwa "Kak Gem" dan "Fufu Fafa" telah dihapus, diikuti dengan tampilan daftar yang diperbarui.