

MODUL 3

SORTING AND SEARCHING

1.1 PERMASALAHAN

1.3.1 List Lagu

Suatu hari, seorang programmer bernama DIvo memiliki misi untuk mengelola koleksi lagu dengan cara yang efisien dan terorganisir. Dengan tujuan menciptakan sistem manajemen lagu yang baik, DIvo mendefinisikan kelas SpotifyVo yang berfungsi untuk menyimpan dan mengatur daftar lagu dalam sebuah struktur data double-linked list. Setiap lagu memiliki atribut seperti judul, penyanyi, tahun rilis, dan durasi, yang diorganisir dalam urutan yang dapat dengan mudah diakses dan diubah. Dalam kelas SpotifyVo, DIvo menciptakan metode untuk menambahkan lagu baru ke dalam daftar, menampilkan seluruh koleksi lagu dengan format yang rapi, serta melakukan pengurutan daftar lagu berdasarkan durasi atau tahun rilis menggunakan metode Bubble Sort. Dengan metode pengurutan ini, DIvo dapat memastikan bahwa daftar lagu selalu tertata sesuai dengan keinginan, baik secara ascending (dari durasi terpendek) maupun descending (berdasarkan tahun rilis terbaru). Tidak hanya itu, DIvo juga menambahkan fitur pencarian menggunakan metode linearSearch untuk memungkinkan pengguna mencari lagu berdasarkan judul. Dengan cara ini, DIvo dapat memudahkan pengguna menemukan lagu tertentu di antara ribuan pilihan yang ada. Untuk menguji sistem yang telah dibuatnya, DIvo membuat kelas Main, di mana ia mengisi koleksi lagu dengan beberapa lagu legendaris dari berbagai penyanyi terkenal seperti Scorpions, Europe, Helloween, dan banyak lagi. Setiap lagu memiliki informasi lengkap mengenai durasi, tahun rilis, dan penyanyi yang memudahkan DIvo untuk mengelola serta menampilkan data. Setelah mengisi daftar lagu, DIvo mencoba beberapa operasi untuk memastikan sistemnya berfungsi dengan baik. Dia menampilkan koleksi lagu yang sudah terurut berdasarkan durasi, kemudian mengurutkan ulang berdasarkan tahun rilis, dan akhirnya melakukan pencarian untuk menemukan lagu tertentu. DIvo merasa puas karena sistem manajemen lagu ini berjalan dengan lancar, efektif, dan memudahkan pengguna untuk menikmati berbagai lagu dengan cara yang terstruktur dan rapi.

1.2 HASIL PERCOBAAN

1.4.1 Program List Buku

1. Algoritma

- a. Menambahkan Lagu ke Playlist
- b. Menggabungkan Daftar Playlist.
- c. Menambahkan Lagu ke Daftar Playlist
- d. Menghapus Lagu dari Playlist.
- e. Menambahkan dan Menghapus Lagu dalam Playlist.
- f. Menukar Posisi Lagu dalam Playlist
- g. Menampilkan Daftar Playlist

2. Source Code

```
public class LaguVo {
    String judulLagu;
    String penyanyi;
    int tahunRilis;
    int menit;
    int detik;
    LaguVo next;
    LaguVo prev;

    public LaguVo(String judulLagu, String penyanyi, int tahunRilis,
int menit, int detik) {
        this.judulLagu = judulLagu;
        this.penyanyi = penyanyi;
        this.tahunRilis = tahunRilis;
        this.menit = menit;
        this.detik = detik;
        this.next = null;
        this.prev = null;
    }
}

public class SpotifyVo {
    LaguVo head;
    LaguVo tail;

    public void add(String judulLagu, String penyanyi, int tahunRilis,
int menit, int detik) {
        LaguVo newNode = new LaguVo(judulLagu, penyanyi, tahunRilis,
menit, detik);
        if (head == null) {
            head = tail = newNode;
        } else {
            tail.next = newNode;
            newNode.prev = tail;
            tail = newNode;
        }
    }

    public void display() {
        LaguVo temp = head;
        int index = 1;
```

```

        System.out.println("||===== LOL
Playlist =====||");
        System.out.println("=====
=====");

        while (temp != null) {
            System.out.printf("\t%d | %s - %s (%d)\n\t\t %d:%02d\n",
                                index, temp.penyanyi, temp.judulLagu,
temp.tahunRilis, temp.menit, temp.detik);
            System.out.println("=====
=====");

            temp = temp.next;
            index++;
        }
    }

    public void asc() {
        if (head == null) return;
        long startTime = System.nanoTime();

        boolean swapped;
        do {
            swapped = false;
            LaguVo temp = head;
            LaguVo temp1 = null;

            while (temp.next != null) {
                LaguVo ptr = temp.next;
                if (temp.menit > ptr.menit || (temp.menit == ptr.menit
&& temp.detik > ptr.detik)) {
                    if (temp == head) {
                        head = ptr;
                    } else {
                        temp1.next = ptr;
                    }
                    temp.next = ptr.next;
                    ptr.next = temp;
                    swapped = true;
                    temp1 = ptr;
                } else {
                    temp1 = temp;
                    temp = temp.next;
                }
            }
        } while (swapped);

        long endTime = System.nanoTime();
        long duration = endTime - startTime;
        System.out.println("Waktu pengurutan ASC: " + duration + "
nanodetik");
    }

    public void desc() {
        if (head == null) return;
        long startTime = System.nanoTime();

        boolean swapped;
        do {
            swapped = false;
            LaguVo temp = head;
            LaguVo temp1 = null;

```

```

        while (temp.next != null) {
            LaguVo ptr = temp.next;

            if (temp.tahunRilis < ptr.tahunRilis) {
                if (temp == head) {
                    head = ptr;
                } else {
                    temp1.next = ptr;
                }
                temp.next = ptr.next;
                ptr.next = temp;
                swapped = true;
                temp1 = ptr;
            } else {
                temp1 = temp;
                temp = temp.next;
            }
        }
    } while (swapped);

    long endTime = System.nanoTime();
    long duration = endTime - startTime;
    System.out.println("Waktu pengurutan DESC: " + duration + "
nanodetik");
}

public void linearSearch(String judulLagu) {
    LaguVo temp = head;
    int index = 1;
    boolean found = false;

    System.out.println("||===== Hasil Pencarian
=====||");
    while (temp != null) {
        if (temp.judulLagu.equalsIgnoreCase(judulLagu)) {
            System.out.printf("Lagu ditemukan di posisi ke-
%d:\n", index);
            System.out.printf("\t%d | %s - %s
(%d)\n\t %d:%02d\n",
                                index, temp.penyanyi, temp.judulLagu,
temp.tahunRilis, temp.menit, temp.detik);
            System.out.println("=====");
            found = true;
            break;
        }
        temp = temp.next;
        index++;
    }

    if (!found) {
        System.out.println("Lagu dengan judul '" + judulLagu +
        "'" tidak ditemukan.");
    }
}

public class Main {
    public static void main(String[] args) {
        SpotifyVo list = new SpotifyVo();
    }
}

```

```

list.add("Wind of Change", "Scorpions", 1990, 5, 10);
list.add("Still Loving You", "Scorpions", 1984, 6, 26);
list.add("Rock You Like a Hurricane", "Scorpions", 1984, 4,
12);
list.add("Holiday", "Scorpions", 1979, 6, 32);

list.add("The Final Countdown", "Europe", 1986, 5, 9);
list.add("Carrie", "Europe", 1986, 4, 32);
list.add("Rock the Night", "Europe", 1985, 4, 4);
list.add("Superstitious", "Europe", 1988, 4, 35);

list.add("Forever and One", "Helloween", 1996, 3, 54);
list.add("I Want Out", "Helloween", 1988, 4, 41);
list.add("Eagle Fly Free", "Helloween", 1988, 5, 8);
list.add("A Tale That Wasn't Right", "Helloween", 1987, 4,
42);

list.add("Zombie", "The Cranberries", 1994, 5, 6);
list.add("Linger", "The Cranberries", 1993, 4, 34);
list.add("Dreams", "The Cranberries", 1993, 4, 15);
list.add("Ode to My Family", "The Cranberries", 1994, 4, 32);

list.add("She's Gone", "Steelheart", 1990, 5, 52);
list.add("I'll Never Let You Go", "Steelheart", 1990, 5, 6);
list.add("Everybody Loves Eileen", "Steelheart", 1990, 5,
10);

list.add("Sweet Child O' Mine", "Guns N' Roses", 1987, 5,
56);

list.add("November Rain", "Guns N' Roses", 1991, 8, 57);
list.add("Paradise City", "Guns N' Roses", 1988, 6, 46);
list.add("Patience", "Guns N' Roses", 1989, 5, 56);

list.add("Livin' on a Prayer", "Bon Jovi", 1986, 4, 9);
list.add("You Give Love a Bad Name", "Bon Jovi", 1986, 3,
42);

list.add("Always", "Bon Jovi", 1994, 5, 53);
list.add("Wanted Dead or Alive", "Bon Jovi", 1986, 5, 9);

list.add("With or Without You", "U2", 1987, 4, 56);
list.add("One", "U2", 1991, 4, 36);
list.add("Sunday Bloody Sunday", "U2", 1983, 4, 40);
list.add("Pride (In the Name of Love)", "U2", 1984, 3, 48);

list.add("I Want to Break Free", "Queen", 1984, 3, 43);
list.add("Radio Ga Ga", "Queen", 1984, 5, 49);
list.add("A Kind of Magic", "Queen", 1986, 4, 24);
list.add("Under Pressure", "Queen & David Bowie", 1981, 4,
8);

list.add("Here I Go Again", "Whitesnake", 1982, 5, 9);
list.add("Is This Love", "Whitesnake", 1987, 4, 43);
list.add("Fool for Your Loving", "Whitesnake", 1980, 4, 12);
list.add("Still of the Night", "Whitesnake", 1987, 6, 38);

list.add("Every Breath You Take", "The Police", 1983, 4, 13);
list.add("Roxanne", "The Police", 1978, 3, 12);
list.add("Message in a Bottle", "The Police", 1979, 4, 51);
list.add("Don't Stand So Close to Me", "The Police", 1980,
4, 3);

```

```
list.add("Another One Bites the Dust", "Queen", 1980, 3, 35);
list.add("Take on Me", "A-ha", 1985, 3, 46);
list.add("Africa", "Toto", 1982, 4, 55);
list.add("Eye of the Tiger", "Survivor", 1982, 4, 4);

list.add("Alone", "Heart", 1987, 3, 39);
list.add("I Want to Know What Love Is", "Foreigner", 1984,
5, 4);
list.add("Jump", "Van Halen", 1984, 4, 4);
list.add("Beat It", "Michael Jackson", 1982, 4, 18);

list.display();
System.out.println();

System.out.println("Sort By Duration:");
list.asc();
list.display();
System.out.println();

System.out.println("Sort By Year:");
list.desc();
list.display();
System.out.println();

list.linearSearch("The Final Countdown");
}
```

1.3 ANALISIS DATA

1.5.1 LaguVo

```
public class LaguVo {
    String judulLagu;
    String penyanyi;
    int tahunRilis;
    int menit;
    int detik;
    LaguVo next;
    LaguVo prev;

    public LaguVo(String judulLagu, String penyanyi, int tahunRilis, int
menit, int detik) {
        this.judulLagu = judulLagu;
        this.penyanyi = penyanyi;
        this.tahunRilis = tahunRilis;
        this.menit = menit;
        this.detik = detik;
        this.next = null;
        this.prev = null;
    }
}
```

Script “`public class LaguVo {`” adalah kelas yang digunakan untuk merepresentasikan objek lagu dalam sebuah daftar playlist. Kelas ini memiliki atribut-atribut yang menyimpan informasi tentang lagu, seperti `judulLagu` untuk menyimpan judul lagu, `penyanyi` untuk nama penyanyi atau band, `tahunRilis` untuk tahun perilisan lagu, serta `menit` dan `detik` yang menyimpan durasi lagu dalam format menit dan detik. Kelas ini juga menggunakan dua referensi, yaitu `next` dan `prev`, yang menghubungkan objek lagu dalam struktur data doubly linked list, memungkinkan navigasi maju dan mundur antar lagu. Konstruktor `LaguVo` digunakan untuk menginisialisasi atribut-atribut lagu dengan nilai yang diberikan, serta mengatur nilai default `next` dan `prev` menjadi `null`.

1.5.2 SpotifyVo

```
public class SpotifyVo {
    LaguVo head;
    LaguVo tail;

    public void add(String judulLagu, String penyanyi, int tahunRilis,
int menit, int detik) {
        LaguVo newNode = new LaguVo(judulLagu, penyanyi, tahunRilis,
menit, detik);
        if (head == null) {
            head = tail = newNode;
        } else {
            tail.next = newNode;
            newNode.prev = tail;
            tail = newNode;
        }
    }

    public void display() {
        LaguVo temp = head;
    }
}
```

```

        int index = 1;
        System.out.println("||=====                                LOL
Playlist =====||");

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

        while (temp != null) {
            System.out.printf("\t%d | %s - %s (%d)\n\t\t %d:%02d\n",
                index,          temp.penyanyi,          temp.judulLagu,
temp.tahunRilis, temp.menit, temp.detik);

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

            temp = temp.next;
            index++;
        }
    }

    public void asc() {
        if (head == null) return;
        long startTime = System.nanoTime();

        boolean swapped;
        do {
            swapped = false;
            LaguVo temp = head;
            LaguVo temp1 = null;

            while (temp.next != null) {
                LaguVo ptr = temp.next;
                if (temp.menit > ptr.menit || (temp.menit == ptr.menit
&& temp.detik > ptr.detik)) {
                    if (temp == head) {
                        head = ptr;
                    } else {
                        temp1.next = ptr;
                    }
                    temp.next = ptr.next;
                    ptr.next = temp;
                    swapped = true;
                    temp1 = ptr;
                } else {
                    temp1 = temp;
                    temp = temp.next;
                }
            }
        } while (swapped);

        long endTime = System.nanoTime();
        long duration = endTime - startTime;
        System.out.println("Waktu pengurutan ASC: " + duration + "
nanodetik");
    }

    public void desc() {
        if (head == null) return;
        long startTime = System.nanoTime();

        boolean swapped;
        do {

```



```

        swapped = false;
        LaguVo temp = head;
        LaguVo temp1 = null;

        while (temp.next != null) {
            LaguVo ptr = temp.next;

            if (temp.tahunRilis < ptr.tahunRilis) {
                if (temp == head) {
                    head = ptr;
                } else {
                    temp1.next = ptr;
                }
                temp.next = ptr.next;
                ptr.next = temp;
                swapped = true;
                temp1 = ptr;
            } else {
                temp1 = temp;
                temp = temp.next;
            }
        }
    } while (swapped);

    long endTime = System.nanoTime();
    long duration = endTime - startTime;
    System.out.println("Waktu pengurutan DESC: " + duration + "
nanodetik");
}

public void linearSearch(String judulLagu) {
    LaguVo temp = head;
    int index = 1;
    boolean found = false;

    System.out.println("||===== Hasil Pencarian
=====||");
    while (temp != null) {
        if (temp.judulLagu.equalsIgnoreCase(judulLagu)) {
            System.out.printf("Lagu ditemukan di posisi ke- %d:\n",
index);
            System.out.printf("\t%d | %s - %s (%d)\n\t %d:%02d\n",
index, temp.penyanyi, temp.judulLagu,
temp.tahunRilis, temp.menit, temp.detik);

            System.out.println("=====
=====");
            found = true;
            break;
        }
        temp = temp.next;
        index++;
    }

    if (!found) {
        System.out.println("Lagu dengan judul '" + judulLagu + "'
tidak ditemukan.");
    }
}
}

```

Script “`public class Buku{`” adalah kelas yang digunakan untuk merepresentasikan sebuah playlist lagu yang disusun menggunakan struktur data doubly linked list. Kelas ini memiliki atribut head dan tail yang menunjuk ke lagu pertama dan terakhir dalam playlist. Metode utama dalam kelas ini mencakup `add()` untuk menambahkan lagu baru ke akhir playlist, `display()` untuk menampilkan semua lagu dalam playlist, serta `asc()` dan `desc()` untuk mengurutkan playlist berdasarkan durasi lagu (menaik) dan tahun rilis (menurun). Selain itu, terdapat metode `linearSearch()` yang memungkinkan pencarian lagu berdasarkan judul, dengan menampilkan posisi dan detail lagu yang ditemukan.

1.5.3 Main

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

        list.add("Wind of Change", "Scorpions", 1990, 5, 10);
        list.add("Still Loving You", "Scorpions", 1984, 6, 26);
        list.add("Rock You Like a Hurricane", "Scorpions", 1984, 4, 12);
        list.add("Holiday", "Scorpions", 1979, 6, 32);

        list.add("The Final Countdown", "Europe", 1986, 5, 9);
        list.add("Carrie", "Europe", 1986, 4, 32);
        list.add("Rock the Night", "Europe", 1985, 4, 4);
        list.add("Superstitious", "Europe", 1988, 4, 35);

        list.add("Forever and One", "Helloween", 1996, 3, 54);
        list.add("I Want Out", "Helloween", 1988, 4, 41);
        list.add("Eagle Fly Free", "Helloween", 1988, 5, 8);
        list.add("A Tale That Wasn't Right", "Helloween", 1987, 4, 42);

        list.add("Zombie", "The Cranberries", 1994, 5, 6);
        list.add("Linger", "The Cranberries", 1993, 4, 34);
        list.add("Dreams", "The Cranberries", 1993, 4, 15);
        list.add("Ode to My Family", "The Cranberries", 1994, 4, 32);

        list.add("She's Gone", "Steelheart", 1990, 5, 52);
        list.add("I'll Never Let You Go", "Steelheart", 1990, 5, 6);
        list.add("Everybody Loves Eileen", "Steelheart", 1990, 5, 10);

        list.add("Sweet Child O' Mine", "Guns N' Roses", 1987, 5, 56);
        list.add("November Rain", "Guns N' Roses", 1991, 8, 57);
        list.add("Paradise City", "Guns N' Roses", 1988, 6, 46);
        list.add("Patience", "Guns N' Roses", 1989, 5, 56);

        list.add("Livin' on a Prayer", "Bon Jovi", 1986, 4, 9);
        list.add("You Give Love a Bad Name", "Bon Jovi", 1986, 3, 42);
        list.add("Always", "Bon Jovi", 1994, 5, 53);
        list.add("Wanted Dead or Alive", "Bon Jovi", 1986, 5, 9);

        list.add("With or Without You", "U2", 1987, 4, 56);
        list.add("One", "U2", 1991, 4, 36);
        list.add("Sunday Bloody Sunday", "U2", 1983, 4, 40);
        list.add("Pride (In the Name of Love)", "U2", 1984, 3, 48);

        list.add("I Want to Break Free", "Queen", 1984, 3, 43);
    }
}
```

```

list.add("Radio Ga Ga", "Queen", 1984, 5, 49);
list.add("A Kind of Magic", "Queen", 1986, 4, 24);
list.add("Under Pressure", "Queen & David Bowie", 1981, 4, 8);

list.add("Here I Go Again", "Whitesnake", 1982, 5, 9);
list.add("Is This Love", "Whitesnake", 1987, 4, 43);
list.add("Fool for Your Loving", "Whitesnake", 1980, 4, 12);
list.add("Still of the Night", "Whitesnake", 1987, 6, 38);

list.add("Every Breath You Take", "The Police", 1983, 4, 13);
list.add("Roxanne", "The Police", 1978, 3, 12);
list.add("Message in a Bottle", "The Police", 1979, 4, 51);
list.add("Don't Stand So Close to Me", "The Police", 1980, 4,
3);

list.add("Another One Bites the Dust", "Queen", 1980, 3, 35);
list.add("Take on Me", "A-ha", 1985, 3, 46);
list.add("Africa", "Toto", 1982, 4, 55);
list.add("Eye of the Tiger", "Survivor", 1982, 4, 4);

list.add("Alone", "Heart", 1987, 3, 39);
list.add("I Want to Know What Love Is", "Foreigner", 1984, 5,
4);

list.add("Jump", "Van Halen", 1984, 4, 4);
list.add("Beat It", "Michael Jackson", 1982, 4, 18);

list.display();
System.out.println();

System.out.println("Sort By Duration:");
list.asc();
list.display();
System.out.println();

System.out.println("Sort By Year:");
list.desc();
list.display();
System.out.println();

list.linearSearch("The Final Countdown");
}
}

```

Script “public class Main {” adalah kelas utama yang digunakan untuk menjalankan program yang mengelola playlist lagu menggunakan kelas SpotifyVo. Dalam program ini, objek SpotifyVo bernama list dibuat untuk menyimpan playlist lagu-lagu yang ditambahkan menggunakan metode add(). Lagu-lagu tersebut memiliki atribut seperti judul, penyanyi, tahun rilis, serta durasi dalam menit dan detik. Setelah lagu-lagu ditambahkan, program menampilkan playlist menggunakan metode display(). Kemudian, playlist disortir berdasarkan durasi lagu (ascending) dengan metode asc(), dan berdasarkan tahun rilis (descending) dengan metode desc(). Setelah setiap pengurutan, playlist ditampilkan kembali. Terakhir, program melakukan pencarian lagu tertentu menggunakan metode linearSearch(), seperti mencari lagu dengan judul "The Final Countdown".