MODUL III
SORTING AND SEARCHING

3.1 PERMASALAHAN

3.1.1 Playlist Lagu

Kalian masih ingat dengan sosok bernama Rijal yang memiliki rental PS? Ia telah kembali pada cerita kali ini dengan banyak tokoh baru lain-nya. Di siang hari yang sangat terik ini. Rijal sedang sibuk di meja kasir rental PS miliknya ketika Eysar masuk. Eysar adalah teman dekat Rijal semasa perkuliahan dulu. Biasanya, Eysar datang ke sini dengan semangat untuk bermain PES, tapi kali ini wajahnya terlihat murung.

“Kenapa Sar? Biasanya langsung nyari stick, kok sekarang malah bengong?” tanya Rijal, penasaran.

Eysar duduk di sofa yang biasa digunakan pelanggan untuk menunggu giliran bermain. "Nggak tahu, Jal. Lagi galau Jal. Lagi nggak mood buat main" jawabnya sambil menghela napas. Rijal mengerutkan dahi. "LOL, galau kenapa tuh? Cerita dah. Kalau duduk doang di sini, nggak seru kan".

Eysar tersenyum tipis. "Kayaknya saya mau bikin playlist deh di Spotify. Buat nemenin kalau lagi galau begini". Rijal tertawa kecil. "LOL, playlist galau nih. Biar playlist-nya penuh nuansa sendu kayak hatimu?" godanya.

“LOL” kata Eysar sambil mengangguk, mengeluarkan ponsel dan membuka aplikasi Spotify yang masih kosong tanpa lagu di playlist barunya. Dengan sabar, ia mulai memilih lagu-lagu yang cocok. Rijal, yang duduk di sebelahnya sambil mengecek pemesanan pelanggan lain, sesekali memberikan rekomendasi lagu.

“Apa nama playlist buat galamu tu Sar?” kata Rijal yang tertarik dengan playlist Eysar. Eysar tertawa “Rijal Ganteng gimana?”. “LOL” kata Rijal sambil ikut tertawa. Eysar berkata sambil bersemangat “Nah itu aja gimana Jal? LOL Playlist?”. “LOL, iyadah” kata Rijal sambal tertawa.

```
==========================================================================
||                          LOL Playlist                                ||
==========================================================================
                There is no song in the playlist yet.
==========================================================================
```

*** tampilan di saat belum ada lagu yang ditambahkan**

“Saya masukin sepuluh lagu dulu ga sih, ” kata Eysar sambil menambahkan lagu-lagu pilihan-nya ke dalam playlist.

***Note : gunakan double linked list.**

*** tampilan sesudah ada lagu yang ditambahkan**

***Note : atributnya ada judul lagu, nama artis, tahun rilis, durasi (dalam menit dan detik), dan genre.**

Setelah playlistnya terisi dengan sepuluh lagu, Eysar mulai berpikir untuk mengurutkan lagu-lagu ini dengan lebih spesifik lagi. Kali ini, ia ingin mengurutkannya berdasarkan durasi lagu agar alurnya terasa lebih pas.

Rijal, yang sedang membereskan stik PS di rak, mendekat lagi setelah mendengar keinginan Eysar. “Kalau urutin dari durasi, berarti lagu paling pendek dulu, baru yang lebih panjang kan?”.

Eysar mengangguk. Dengan sedikit bantuan dari Rijal, ia memeriksa durasi setiap lagu di playlist dan mulai menyusun ulang. Satu per satu, lagu-lagu tersebut diurutkan dari yang terpendek hingga yang terpanjang. Setelah selesai, playlist itu terlihat lebih rapi dan teratur.

***Note : Mengurutkan menggunakan algoritma Bubble Sort (Asc) berdasarkan durasi menit dan detik.**

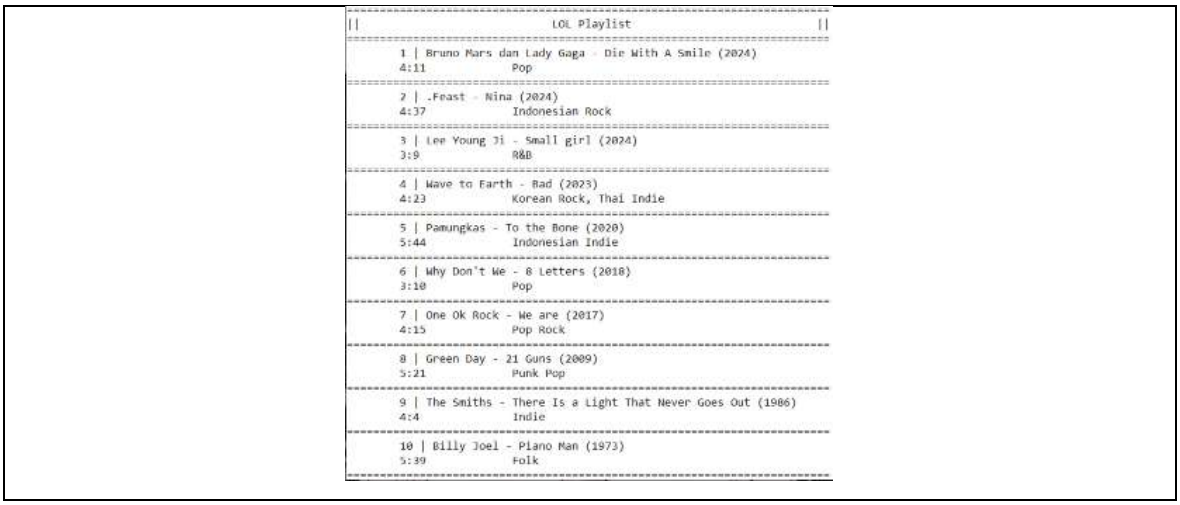
```
||                          LOL Playlist                          ||
=================================================================
       1 | Lee Young Ji - Small girl (1973)
       3:9              R&B
=================================================================
       2 | Why Don't We - 8 Letters (2024)
       3:10             Pop
=================================================================
       3 | The Smiths - There Is a Light That Never Goes Out (2024)
       4:4              Indie
=================================================================
       4 | Bruno Mars dan Lady Gaga - Die With A Smile (2023)
       4:11             Pop
=================================================================
       5 | One Ok Rock - We are (2018)
       4:15             Pop Rock
=================================================================
       6 | Wave to Earth - Bad (1986)
       4:23             Korean Rock, Thai Indie
=================================================================
       7 | .Feast - Nina (2020)
       4:37             Indonesian Rock
=================================================================
       8 | Green Day - 21 Guns (2017)
       5:21             Punk Pop
=================================================================
       9 | Billy Joel - Piano Man (2024)
       5:39             Folk
=================================================================
       10 | Pamungkas - To the Bone (2009)
       5:44             Indonesian Indie
=================================================================
```

* tampilan sesudah lagu di urutkan berdasarkan menit dan detik

Namun, Eysar tiba-tiba punya ide lain. "Jal, kalau saya urutinnya berdasarkan lagu terbaru yang rilis". Rijal tersenyum mengerti. "Jadi urutannya berdasarkan tahun rilis terbaru kan Sar?".

Eysar mengangguk lagi. Ia pun mengaktifkan opsi pengurutan otomatis berdasarkan tahun rilis, sehingga lagu-lagu terbaru akan berada di uturan teratas playlist.

*Note : Mengurutkan menggunakan algoritma Selection Sort (Desc) berdasarkan tahun rilis.



```
||                          LOL Playlist                          ||
=================================================================
       1 | Bruno Mars dan Lady Gaga - Die With A Smile (2024)
       4:11             Pop
=================================================================
       2 | .Feast - Nina (2024)
       4:37             Indonesian Rock
=================================================================
       3 | Lee Young Ji - Small girl (2024)
       3:9              R&B
=================================================================
       4 | Wave to Earth - Bad (2023)
       4:23             Korean Rock, Thai Indie
=================================================================
       5 | Pamungkas - To the Bone (2020)
       5:44             Indonesian Indie
=================================================================
       6 | Why Don't We - 8 Letters (2018)
       3:10             Pop
=================================================================
       7 | One Ok Rock - We are (2017)
       4:15             Pop Rock
=================================================================
       8 | Green Day - 21 Guns (2009)
       5:21             Punk Pop
=================================================================
       9 | The Smiths - There Is a Light That Never Goes Out (1986)
       4:4              Indie
=================================================================
       10 | Billy Joel - Piano Man (1973)
       5:39             Folk
=================================================================
```

* tampilan sesudah lagu di urutkan berdasarkan tahun

Akhirnya setelah cukup lama menyusun dan menata, Eysar menatap playlist-nya dengan senyum lega. Playlist itu kini rapi, terurut dari durasi, dan lagu-lagu terbaru.

Rijal menepuk pundaknya sambil tersenyum. "Playlist siap menemani galaumu tuh Sar" candanya. Eysar tertawa kecil. "LOL, makasi udah bantu ya Jal".

Setelah playlist-nya tertata rapi, Eysar merasa playlist itu sudah cukup sempurna. Namun, suatu hari saat kembali ke rental PS Rijal, ia ingin mencari tahu letak salah satu lagu favoritnya yang ada di playlist.

“Jal, saya lagi pengen dengerin satu lagu di playlist nih, tapi lupa letaknya di urutan ke berapa” kata Eysar sambil membuka Spotify.

Rijal tertawa kecil. “LOL, lagu di playlist-mu kan udah ada sepuluh lebih sekarang. Kalau di-scroll satu-satu bisa lama, Sar”.

*Note: silahkan menambahkan lagu lagi secara bebas sampai 15 lagu.

Eysar mengangguk setuju, lalu mulai mengetik judul lagu yang dia ingat di kolom pencarian dalam playlist-nya. Setelah beberapa detik, lagu yang ia cari muncul, lengkap dengan nomor urutnya.

“Yes, ketemu! Lagunya ada di urutan nomor lima!” seru Eysar dengan senyum puas.

Rijal tersenyum, lalu menggodanya. “Nah, enak kan? Makanya playlist-mu diatur. Biar pas nyari gampang.”

* tampilan hasil pencarian berdasarkan judul lagu menggunakan algoritman linear search.

Eysar tertawa. “Benar juga, Jal. Sekarang nggak ribet lagi kalau mau cari-cari lagu favorit!”

Sejak saat itu, Eysar semakin senang menyusun dan memperbaiki playlist-nya, memastikan lagu-lagu favoritnya mudah ditemukan kapan saja. Rijal, yang melihat semangat temannya, hanya bisa tertawa kecil sambil kembali melayani pelanggan di rental PS.

3.2 HASIL PERCOBAAN

2.2.1 Penjaga Perpustakaan

1. Algoritma
    a. Start
    b. Buat playlist kosong dengan menggunakan struktur data *double linked list*.
    c. Tampilkan tampilan awal playlist yang masih kosong.
    d. **Input** nama playlist sebagai "LOL Playlist".

e. **Loop (i = 1 to 10)** Minta pengguna untuk memasukkan informasi lagu (judul lagu, nama artis, tahun rilis, durasi dalam menit dan detik, serta genre). Dan tambahkan lagu ke playlist.
f. Tampilkan playlist yang berisi 10 lagu yang baru saja ditambahkan.
g. **Sort Playlist by Duration** Gunakan algoritma *Bubble Sort* untuk mengurutkan lagu berdasarkan durasi dalam menit dan detik (ascending).dan tampilkan playlist setelah diurutkan berdasarkan durasi.
h. **Sort Playlist by Year** Gunakan algoritma *Selection Sort* untuk mengurutkan lagu berdasarkan tahun rilis (descending). Tampilkan playlist setelah diurutkan berdasarkan tahun rilis terbaru.
i. **Add Songs to Playlist** Tambahkan 5 lagu tambahan dengan informasi lagu yang berbeda.
j. **Search Song by Title** Minta pengguna untuk memasukkan judul lagu yang ingin dicari. Gunakan algoritma *Linear Search* untuk mencari lagu berdasarkan judul. Jika lagu ditemukan, tampilkan nomor urut lagu dalam playlist.
k. Tampilkan pesan bahwa lagu telah ditemukan beserta posisinya.
l. Finish

3. *Source Code*

```
public class NodeLagu {
    String JudulLagu;
    String NamaArtis;
    int TahunRilis;
    int Durasi;
    String Genre;
    NodeLagu next;
    NodeLagu prev;

    public NodeLagu(String JudulLagu, String NamaArtis, int
TahunRilis, int Durasi, String Genre) {
        this.JudulLagu = JudulLagu;
        this.NamaArtis = NamaArtis;
        this.TahunRilis = TahunRilis;
        this.Durasi = Durasi;
        this.Genre = Genre;
        this.next = null;
        this.prev = null;
    }
}

public class MainRand {
    public static void main(String[] args) {
        StackLagu Playlist = new StackLagu();
```

```
        //Lagu Random
        Playlist.addLagu("All Star", "Smash Mouth", 1999, 200,
"Alternative Rock");
        Playlist.addLagu("Halo", "Beyoncé", 2008, 261, "Pop");
        Playlist.addLagu("Uptown Girl", "Billy Joel", 1983, 198,
"Pop");
        Playlist.addLagu("Africa", "Toto", 1982, 295, "Rock");
        Playlist.addLagu("Born to Run", "Bruce Springsteen", 1975,
271, "Rock");
        Playlist.addLagu("Heroes", "David Bowie", 1977, 215,
"Rock");
        Playlist.addLagu("Dancing Queen", "ABBA", 1976, 232,
"Disco");
        Playlist.addLagu("Losing My Religion", "R.E.M.", 1991, 268,
"Alternative Rock");
        Playlist.addLagu("Stayin' Alive", "Bee Gees", 1977, 285,
"Disco");
        Playlist.addLagu("Call Me Maybe", "Carly Rae Jepsen", 2011,
193, "Pop");
        Playlist.addLagu("Radioactive", "Imagine Dragons", 2012,
187, "Alternative Rock");
        Playlist.addLagu("American Pie", "Don McLean", 1971, 513,
"Folk Rock");
        Playlist.addLagu("I Will Survive", "Gloria Gaynor", 1978,
199, "Disco");
        Playlist.addLagu("Smells Like Teen Spirit", "Nirvana",
1991, 301, "Grunge");
        Playlist.addLagu("Uptown Funk", "Mark Ronson ft. Bruno
Mars", 2014, 269, "Funk");
        Playlist.addLagu("Girls Just Want to Have Fun", "Cyndi
Lauper", 1983, 235, "Pop");
        Playlist.addLagu("Eye of the Tiger", "Survivor", 1982, 245,
"Rock");
        Playlist.addLagu("I Want to Break Free", "Queen", 1984,
261, "Rock");
        Playlist.addLagu("Every Breath You Take", "The Police",
1983, 253, "Rock");
        Playlist.addLagu("Purple Rain", "Prince", 1984, 521,
"Rock");

        Playlist.showPlaylistSortingDurasi();
        Playlist.showPlaylistSortingTahunRilis();
    }
}

package Easy;

public class StackLagu {
    NodeLagu head;
    NodeLagu tail;

    public void addLagu(String JudulLagu, String NamaArtis, int
TahunRilis, int Durasi, String Genre) {
        NodeLagu newLagu = new NodeLagu(JudulLagu, NamaArtis,
TahunRilis, Durasi, Genre);
        if (head == null) {
            head = newLagu;
            tail = newLagu;
        } else {
            tail.next = newLagu;
            newLagu.prev = tail;
```

```
            tail = newLagu;
        }
    }

    public void showPlaylist() {
        int count = 1;

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

System.out.println("=============================================");
        NodeLagu current = head;
        if (head == null) {
            System.out.println("       There is no song in the
Playlist yet.");

System.out.println("=============================================
=========");
        }
        while (current != null) {
            System.out.println(count + " | " + current.NamaArtis +
" - " + current.JudulLagu + " (" + current.TahunRilis + ")");
            System.out.println("\t" + current.Durasi / 60 + ":" +
current.Durasi % 60 + "\t" + current.Genre);

System.out.println("=============================================
=========");
            count++;
            current = current.next;
        }
    }


    public void showPlaylistSortingDurasi() {
        if (head == null || head.next == null) {
            return;
        }

        long waktuMulai = System.nanoTime();

        boolean tukar;

        do {
            tukar = false;
            NodeLagu current = head;

            while (current.next != null) {
                if (current.Durasi > current.next.Durasi) {
                    TukarDurasi(current, current.next);
                    tukar = true;
                }
                current = current.next;
            }
        } while (tukar);

        long waktuBerhenti = System.nanoTime();
        long durasi = waktuBerhenti - waktuMulai;

        System.out.println("\nPlaylist diurutkan berdasarkan durasi
secara ascending:");
```

```
        showPlaylist();
        System.out.println("Durasi: " + durasi);
    }

    private void TukarDurasi(NodeLagu a, NodeLagu b) {
        String tempJudulLagu = a.JudulLagu;
        String tempNamaArtis = a.NamaArtis;
        int tempTahunRilis = a.TahunRilis;
        int tempDurasi = a.Durasi;
        String tempGenre = a.Genre;

        a.JudulLagu = b.JudulLagu;
        a.NamaArtis = b.NamaArtis;
        a.TahunRilis = b.TahunRilis;
        a.Durasi = b.Durasi;
        a.Genre = b.Genre;

        b.JudulLagu = tempJudulLagu;
        b.NamaArtis = tempNamaArtis;
        b.TahunRilis = tempTahunRilis;
        b.Durasi = tempDurasi;
        b.Genre = tempGenre;
    }

    public void showPlaylistSortingTahunRilis() {
        if (head == null || head.next == null) {
            return;
        }

        long waktuMulai = System.nanoTime();

        NodeLagu current = head;

        while (current != null) {
            NodeLagu maxNode = current;
            NodeLagu nextNode = current.next;

            while (nextNode != null) {
                if (nextNode.TahunRilis > maxNode.TahunRilis) {
                    maxNode = nextNode;
                }
                nextNode = nextNode.next;
            }

            if (maxNode != current) {
                TukarTahunRilis(current, maxNode);
            }

            current = current.next;
        }

        long waktuBerhenti = System.nanoTime();
        long durasi = waktuBerhenti - waktuMulai;

        System.out.println("\nPlaylist diurutkan berdasarkan tahun
rilis secara descending:");
        showPlaylist();
        System.out.println("Durasi: " + durasi);
    }

    private void TukarTahunRilis(NodeLagu a, NodeLagu b) {
```

```
        String tempJudulLagu = a.JudulLagu;
        String tempNamaArtis = a.NamaArtis;
        int tempTahunRilis = a.TahunRilis;
        int tempDurasi = a.Durasi;
        String tempGenre = a.Genre;

        a.JudulLagu = b.JudulLagu;
        a.NamaArtis = b.NamaArtis;
        a.TahunRilis = b.TahunRilis;
        a.Durasi = b.Durasi;
        a.Genre = b.Genre;

        b.JudulLagu = tempJudulLagu;
        b.NamaArtis = tempNamaArtis;
        b.TahunRilis = tempTahunRilis;
        b.Durasi = tempDurasi;
        b.Genre = tempGenre;
    }

    public void cariLagu(String judulLaguDicari) {

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

System.out.println("=============================================");
        System.out.println("Judul lagu yang ingin dicari : " +
judulLaguDicari);

        NodeLagu current = head;
        int posisi = 1;
        boolean ditemukan = false;

        while (current != null) {
            if
(current.JudulLagu.equalsIgnoreCase(judulLaguDicari)) {
                System.out.println("Lagu yang sesuai dengan judul
'" + judulLaguDicari + "' ditemukan pada posisi ke - " + posisi);

System.out.println("================================================
=========");
                System.out.println(posisi + " | " +
current.NamaArtis + " - " + current.JudulLagu + " (" +
current.TahunRilis + ")");
                System.out.println("\t" + current.Durasi / 60 + ":"
+ current.Durasi % 60 + "\t" + current.Genre);

System.out.println("================================================
=========");
                ditemukan = true;
                break;
            }
            current = current.next;
            posisi++;
        }

        if (!ditemukan) {
            System.out.println("Lagu dengan judul '" +
judulLaguDicari + "' tidak ditemukan dalam playlist.");

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

```
    }
}


public class MainAsc {
    public static void main(String[] args) {
        StackLagu Playlist = new StackLagu();

        //Lagu Ascending
        Playlist.addLagu("Imagine", "John Lennon", 1971, 184,
"Rock");
        Playlist.addLagu("Hotel California", "Eagles", 1977, 390,
"Rock");
        Playlist.addLagu("Billie Jean", "Michael Jackson", 1982,
294, "Pop");
        Playlist.addLagu("Sweet Child O' Mine", "Guns N' Roses",
1987, 356, "Rock");
        Playlist.addLagu("Nothing Compares 2 U", "Sinéad O'Connor",
1990, 310, "Pop");
        Playlist.addLagu("Smells Like Teen Spirit", "Nirvana",
1991, 301, "Rock");
        Playlist.addLagu("Creep", "Radiohead", 1992, 238, "Rock");
        Playlist.addLagu("Wonderwall", "Oasis", 1995, 258,
"Alternative");
        Playlist.addLagu("Torn", "Natalie Imbruglia", 1997, 245,
"Pop");
        Playlist.addLagu("Baby One More Time", "Britney Spears",
1998, 211, "Pop");
        Playlist.addLagu("Stan", "Eminem", 2000, 404, "Hip-Hop");
        Playlist.addLagu("In the End", "Linkin Park", 2000, 216,
"Nu-Metal");
        Playlist.addLagu("Boulevard of Broken Dreams", "Green Day",
2004, 260, "Rock");
        Playlist.addLagu("Rolling in the Deep", "Adele", 2010, 228,
"Soul");
        Playlist.addLagu("Shape of You", "Ed Sheeran", 2017, 233,
"Pop");

        Playlist.showPlaylistSortingDurasi();
        Playlist.showPlaylistSortingTahunRilis();
    }
}

public class MainDesc {
    public static void main(String[] args) {
        StackLagu Playlist = new StackLagu();

        //Lagu Descending
        Playlist.addLagu("Blinding Lights", "The Weeknd", 2020,
200, "Synth-pop");
        Playlist.addLagu("Old Town Road", "Lil Nas X", 2019, 157,
"Country Rap");
        Playlist.addLagu("Shallow", "Lady Gaga & Bradley Cooper",
2018, 217, "Pop");
        Playlist.addLagu("Despacito", "Luis Fonsi ft. Daddy
Yankee", 2017, 227, "Reggaeton");
        Playlist.addLagu("Closer", "The Chainsmokers ft. Halsey",
2016, 244, "Electronic");
        Playlist.addLagu("Uptown Funk", "Mark Ronson ft. Bruno
Mars", 2014, 269, "Funk");
        Playlist.addLagu("Happy", "Pharrell Williams", 2013, 233,
```

```
"Pop");
        Playlist.addLagu("Radioactive", "Imagine Dragons", 2012,
187, "Alternative Rock");
        Playlist.addLagu("Rolling in the Deep", "Adele", 2010, 228,
"Soul");
        Playlist.addLagu("Fireflies", "Owl City", 2009, 228,
"Synth-pop");
        Playlist.addLagu("Viva La Vida", "Coldplay", 2008, 242,
"Alternative Rock");
        Playlist.addLagu("Bleeding Love", "Leona Lewis", 2007, 262,
"Pop");
        Playlist.addLagu("Rehab", "Amy Winehouse", 2006, 214,
"Soul");
        Playlist.addLagu("Fix You", "Coldplay", 2005, 295,
"Alternative Rock");
        Playlist.addLagu("Hey Ya!", "Outkast", 2003, 235, "Hip-
Hop");

        Playlist.showPlaylistSortingDurasi();
        Playlist.showPlaylistSortingTahunRilis();
    }
}

package Easy;

public class Main {
    public static void main(String[] args) {
        StackLagu Playlist = new StackLagu();

        Playlist.showPlaylist();
        Playlist.addLagu("Piano Man", "Billy Joel", 1973, 339,
"Folk");
        Playlist.addLagu("Die With A Smile", "Bruno Mars dan Lady
Gaga", 2024, 251, "Pop");
        Playlist.addLagu("Nina", ".Feast", 2024, 277, "Indonesian
Rock");
        Playlist.addLagu("Bad", "Wave to Earth", 2023, 263, "Korean
Rock, Thai Indie");
        Playlist.addLagu("8 Letters", "Why Don't We", 2018, 190,
"Pop");
        Playlist.addLagu("There is A Light That Never Goes Out",
"The Smiths", 1986, 244, "Indie");
        Playlist.addLagu("To the Bone", "Pamungkas", 2020, 344,
"Indonesian Indie");
        Playlist.addLagu("We are", "One Ok Rock", 2017, 255, "Pop
Rock");
        Playlist.addLagu("Small girl", "Lee Young Ji", 2024, 189,
"R&B");
        Playlist.addLagu("21 Guns", "Green Day", 2009, 321, "Punk
Pop");
        Playlist.showPlaylist();

        Playlist.showPlaylistSortingDurasi();

        Playlist.showPlaylistSortingTahunRilis();

        Playlist.cariLagu("We are");

        //Lagu Ascending
        Playlist.addLagu("Imagine", "John Lennon", 1971, 184,
"Rock");
```

```
        Playlist.addLagu("Hotel California", "Eagles", 1977, 390,
"Rock");
        Playlist.addLagu("Billie Jean", "Michael Jackson", 1982,
294, "Pop");
        Playlist.addLagu("Sweet Child O' Mine", "Guns N' Roses",
1987, 356, "Rock");
        Playlist.addLagu("Nothing Compares 2 U", "Sinéad O'Connor",
1990, 310, "Pop");
        Playlist.addLagu("Smells Like Teen Spirit", "Nirvana",
1991, 301, "Rock");
        Playlist.addLagu("Creep", "Radiohead", 1992, 238, "Rock");
        Playlist.addLagu("Wonderwall", "Oasis", 1995, 258,
"Alternative");
        Playlist.addLagu("Torn", "Natalie Imbruglia", 1997, 245,
"Pop");
        Playlist.addLagu("Baby One More Time", "Britney Spears",
1998, 211, "Pop");
        Playlist.addLagu("Stan", "Eminem", 2000, 404, "Hip-Hop");
        Playlist.addLagu("In the End", "Linkin Park", 2000, 216,
"Nu-Metal");
        Playlist.addLagu("Boulevard of Broken Dreams", "Green Day",
2004, 260, "Rock");
        Playlist.addLagu("Rolling in the Deep", "Adele", 2010, 228,
"Soul");
        Playlist.addLagu("Shape of You", "Ed Sheeran", 2017, 233,
"Pop");

        //Lagu Descending
        Playlist.addLagu("Blinding Lights", "The Weeknd", 2020,
200, "Synth-pop");
        Playlist.addLagu("Old Town Road", "Lil Nas X", 2019, 157,
"Country Rap");
        Playlist.addLagu("Shallow", "Lady Gaga & Bradley Cooper",
2018, 217, "Pop");
        Playlist.addLagu("Despacito", "Luis Fonsi ft. Daddy
Yankee", 2017, 227, "Reggaeton");
        Playlist.addLagu("Closer", "The Chainsmokers ft. Halsey",
2016, 244, "Electronic");
        Playlist.addLagu("Uptown Funk", "Mark Ronson ft. Bruno
Mars", 2014, 269, "Funk");
        Playlist.addLagu("Happy", "Pharrell Williams", 2013, 233,
"Pop");
        Playlist.addLagu("Radioactive", "Imagine Dragons", 2012,
187, "Alternative Rock");
        Playlist.addLagu("Rolling in the Deep", "Adele", 2010, 228,
"Soul");
        Playlist.addLagu("Fireflies", "Owl City", 2009, 228,
"Synth-pop");
        Playlist.addLagu("Viva La Vida", "Coldplay", 2008, 242,
"Alternative Rock");
        Playlist.addLagu("Bleeding Love", "Leona Lewis", 2007, 262,
"Pop");
        Playlist.addLagu("Rehab", "Amy Winehouse", 2006, 214,
"Soul");
        Playlist.addLagu("Fix You", "Coldplay", 2005, 295,
"Alternative Rock");
        Playlist.addLagu("Hey Ya!", "Outkast", 2003, 235, "Hip-
Hop");

        //Lagu Random
        Playlist.addLagu("All Star", "Smash Mouth", 1999, 200,
```

```
"Alternative Rock");
        Playlist.addLagu("Halo", "Beyoncé", 2008, 261, "Pop");
        Playlist.addLagu("Uptown Girl", "Billy Joel", 1983, 198,
"Pop");
        Playlist.addLagu("Africa", "Toto", 1982, 295, "Rock");
        Playlist.addLagu("Born to Run", "Bruce Springsteen", 1975,
271, "Rock");
        Playlist.addLagu("Heroes", "David Bowie", 1977, 215,
"Rock");
        Playlist.addLagu("Dancing Queen", "ABBA", 1976, 232,
"Disco");
        Playlist.addLagu("Losing My Religion", "R.E.M.", 1991, 268,
"Alternative Rock");
        Playlist.addLagu("Stayin' Alive", "Bee Gees", 1977, 285,
"Disco");
        Playlist.addLagu("Call Me Maybe", "Carly Rae Jepsen", 2011,
193, "Pop");
        Playlist.addLagu("Radioactive", "Imagine Dragons", 2012,
187, "Alternative Rock");
        Playlist.addLagu("American Pie", "Don McLean", 1971, 513,
"Folk Rock");
        Playlist.addLagu("I Will Survive", "Gloria Gaynor", 1978,
199, "Disco");
        Playlist.addLagu("Smells Like Teen Spirit", "Nirvana",
1991, 301, "Grunge");
        Playlist.addLagu("Uptown Funk", "Mark Ronson ft. Bruno
Mars", 2014, 269, "Funk");
        Playlist.addLagu("Girls Just Want to Have Fun", "Cyndi
Lauper", 1983, 235, "Pop");
        Playlist.addLagu("Eye of the Tiger", "Survivor", 1982, 245,
"Rock");
        Playlist.addLagu("I Want to Break Free", "Queen", 1984,
261, "Rock");
        Playlist.addLagu("Every Breath You Take", "The Police",
1983, 253, "Rock");
        Playlist.addLagu("Purple Rain", "Prince", 1984, 521,
"Rock");

        Playlist.showPlaylistSortingDurasi();
        Playlist.showPlaylistSortingTahunRilis();
    }
}
```

3.3 ANALISIS DATA

3.3.1 Penjaga Perpustakaan

```
public class NodeLagu {
    String JudulLagu;
    String NamaArtis;
    int TahunRilis;
    int Durasi;
    String Genre;
    NodeLagu next;
    NodeLagu prev;

    public NodeLagu(String JudulLagu, String NamaArtis, int TahunRilis,
int Durasi, String Genre) {
        this.JudulLagu = JudulLagu;
        this.NamaArtis = NamaArtis;
        this.TahunRilis = TahunRilis;
        this.Durasi = Durasi;
        this.Genre = Genre;
        this.next = null;
        this.prev = null;
    }
}
```

Script "`public class NodeLagu {`" berfungsi sebagai node dalam struktur data *linked list* untuk menyimpan informasi tentang lagu. Setiap objek "`NodeLagu`" berisi beberapa atribut yang mendeskripsikan informasi tentang sebuah lagu, yaitu `JudulLagu` (judul lagu), `NamaArtis` (nama artis), `TahunRilis` (tahun rilis lagu), `Durasi` (durasi lagu dalam satuan waktu, misalnya detik), dan `Genre` (jenis musik). Kelas ini juga memiliki dua atribut tambahan, `next` dan `prev`, yang merujuk ke node berikutnya dan sebelumnya dalam *linked list*, sehingga memungkinkan *linked list* untuk disusun secara dua arah (*doubly linked list*). Konstruktor dari kelas ini digunakan untuk menginisialisasi nilai atribut-atribut tersebut ketika objek `NodeLagu` dibuat, di mana `next` dan `prev` diatur ke `null` sebagai nilai awal.

```
public class MainRand {
    public static void main(String[] args) {
        StackLagu Playlist = new StackLagu();

        //Lagu Random
        Playlist.addLagu("All Star", "Smash Mouth", 1999, 200,
"Alternative Rock");
        Playlist.addLagu("Halo", "Beyoncé", 2008, 261, "Pop");
        Playlist.addLagu("Uptown Girl", "Billy Joel", 1983, 198,
"Pop");
        Playlist.addLagu("Africa", "Toto", 1982, 295, "Rock");
        Playlist.addLagu("Born to Run", "Bruce Springsteen", 1975, 271,
"Rock");
        Playlist.addLagu("Heroes", "David Bowie", 1977, 215, "Rock");
        Playlist.addLagu("Dancing Queen", "ABBA", 1976, 232, "Disco");
        Playlist.addLagu("Losing My Religion", "R.E.M.", 1991, 268,
"Alternative Rock");
        Playlist.addLagu("Stayin' Alive", "Bee Gees", 1977, 285,
"Disco");
```

```
        Playlist.addLagu("Call Me Maybe", "Carly Rae Jepsen", 2011,
193, "Pop");
        Playlist.addLagu("Radioactive", "Imagine Dragons", 2012, 187,
"Alternative Rock");
        Playlist.addLagu("American Pie", "Don McLean", 1971, 513, "Folk
Rock");
        Playlist.addLagu("I Will Survive", "Gloria Gaynor", 1978, 199,
"Disco");
        Playlist.addLagu("Smells Like Teen Spirit", "Nirvana", 1991,
301, "Grunge");
        Playlist.addLagu("Uptown Funk", "Mark Ronson ft. Bruno Mars",
2014, 269, "Funk");
        Playlist.addLagu("Girls Just Want to Have Fun", "Cyndi Lauper",
1983, 235, "Pop");
        Playlist.addLagu("Eye of the Tiger", "Survivor", 1982, 245,
"Rock");
        Playlist.addLagu("I Want to Break Free", "Queen", 1984, 261,
"Rock");
        Playlist.addLagu("Every Breath You Take", "The Police", 1983,
253, "Rock");
        Playlist.addLagu("Purple Rain", "Prince", 1984, 521, "Rock");

        Playlist.showPlaylistSortingDurasi();
        Playlist.showPlaylistSortingTahunRilis();
    }
}
```

Script "`public class MainRand {`" berfungsi sebagai program utama untuk mengelola sebuah daftar putar (playlist) lagu menggunakan objek dari kelas "`StackLagu`". Dalam main method, sebuah objek bernama Playlist dibuat dari kelas StackLagu, yang kemungkinan besar bertindak sebagai wadah untuk menyimpan lagu-lagu dalam struktur stack atau tumpukan. Kode ini memasukkan sejumlah lagu dengan informasi seperti judul lagu, nama artis, tahun rilis, durasi, dan genre menggunakan metode "`addLagu`" pada objek Playlist. Lagu-lagu yang dimasukkan beragam dalam hal genre, tahun rilis, dan durasi.

Setelah semua lagu ditambahkan ke dalam Playlist, metode "`showPlaylistSortingDurasi()`" dan "`showPlaylistSortingTahunRilis()`" dipanggil. Kedua metode ini mungkin berfungsi untuk menampilkan daftar lagu yang diurutkan berdasarkan durasi atau tahun rilis, meskipun implementasi dari metode-metode ini tidak disertakan dalam kode yang diberikan.

```
public class StackLagu {
    NodeLagu head;
    NodeLagu tail;

    public void addLagu(String JudulLagu, String NamaArtis, int
TahunRilis, int Durasi, String Genre) {
        NodeLagu newLagu = new NodeLagu(JudulLagu, NamaArtis,
TahunRilis, Durasi, Genre);
        if (head == null) {
            head = newLagu;
            tail = newLagu;
```

```
        } else {
            tail.next = newLagu;
            newLagu.prev = tail;
            tail = newLagu;
        }
    }

    public void showPlaylist() {
        int count = 1;

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

System.out.println("================================================
=====");
        NodeLagu current = head;
        if (head == null) {
            System.out.println("
 There is no song in the Playlist yet.");

System.out.println("================================================
");
        }
        while (current != null) {
            System.out.println(count + " | " + current.NamaArtis + " -
" + current.JudulLagu + " (" + current.TahunRilis + ")");
            System.out.println("\t" + current.Durasi / 60 + ":" +
current.Durasi % 60 + "\t" + current.Genre);

System.out.println("================================================"
);
            count++;
            current = current.next;
        }
    }
```

Script "`public class StackLagu dan public void showPlaylist()`" yang digunakan untuk mengelola playlist lagu menggunakan struktur data linked list dua arah (doubly linked list). Kelas ini memiliki dua atribut, yaitu head dan tail, yang masing-masing menunjuk ke node pertama dan terakhir dalam playlist. Kelas ini menyediakan metode addLagu untuk menambahkan lagu baru ke dalam playlist, serta metode showPlaylist untuk menampilkan semua lagu yang ada. Kemudian, variabel current diinisialisasi dengan head untuk memulai dari lagu pertama. Jika head bernilai null, pesan bahwa playlist kosong akan ditampilkan. Jika playlist tidak kosong, metode ini menjalankan loop while untuk mencetak detail setiap lagu, termasuk nomor urut (count), nama artis (NamaArtis), judul lagu (JudulLagu), tahun rilis (TahunRilis), durasi dalam format "menit", dan genre (Genre). Setiap lagu dipisahkan dengan garis batas agar lebih rapi dan mudah dibaca. count diincrement setelah setiap lagu ditampilkan, dan current diperbarui ke node berikutnya (current.next) hingga mencapai akhir daftar (null).

```
public class MainAsc {
    public static void main(String[] args) {
        StackLagu Playlist = new StackLagu();

        //Lagu Ascending
        Playlist.addLagu("Imagine", "John Lennon", 1971, 184, "Rock");
        Playlist.addLagu("Hotel California", "Eagles", 1977, 390,
"Rock");
        Playlist.addLagu("Billie Jean", "Michael Jackson", 1982, 294,
"Pop");
        Playlist.addLagu("Sweet Child O' Mine", "Guns N' Roses", 1987,
356, "Rock");
        Playlist.addLagu("Nothing Compares 2 U", "Sinéad O'Connor",
1990, 310, "Pop");
        Playlist.addLagu("Smells Like Teen Spirit", "Nirvana", 1991,
301, "Rock");
        Playlist.addLagu("Creep", "Radiohead", 1992, 238, "Rock");
        Playlist.addLagu("Wonderwall", "Oasis", 1995, 258,
"Alternative");
        Playlist.addLagu("Torn", "Natalie Imbruglia", 1997, 245,
"Pop");
        Playlist.addLagu("Baby One More Time", "Britney Spears", 1998,
211, "Pop");
        Playlist.addLagu("Stan", "Eminem", 2000, 404, "Hip-Hop");
        Playlist.addLagu("In the End", "Linkin Park", 2000, 216, "Nu-
Metal");
        Playlist.addLagu("Boulevard of Broken Dreams", "Green Day",
2004, 260, "Rock");
        Playlist.addLagu("Rolling in the Deep", "Adele", 2010, 228,
"Soul");
        Playlist.addLagu("Shape of You", "Ed Sheeran", 2017, 233,
"Pop");

        Playlist.showPlaylistSortingDurasi();
        Playlist.showPlaylistSortingTahunRilis();
    }
}


public class MainDesc {
    public static void main(String[] args) {
        StackLagu Playlist = new StackLagu();

        //Lagu Descending
        Playlist.addLagu("Blinding Lights", "The Weeknd", 2020, 200,
"Synth-pop");
        Playlist.addLagu("Old Town Road", "Lil Nas X", 2019, 157,
"Country Rap");
        Playlist.addLagu("Shallow", "Lady Gaga & Bradley Cooper", 2018,
217, "Pop");
        Playlist.addLagu("Despacito", "Luis Fonsi ft. Daddy Yankee",
2017, 227, "Reggaeton");
        Playlist.addLagu("Closer", "The Chainsmokers ft. Halsey", 2016,
244, "Electronic");
        Playlist.addLagu("Uptown Funk", "Mark Ronson ft. Bruno Mars",
2014, 269, "Funk");
        Playlist.addLagu("Happy", "Pharrell Williams", 2013, 233,
"Pop");
        Playlist.addLagu("Radioactive", "Imagine Dragons", 2012, 187,
"Alternative Rock");
        Playlist.addLagu("Rolling in the Deep", "Adele", 2010, 228,
```

```
"Soul");
        Playlist.addLagu("Fireflies", "Owl City", 2009, 228, "Synth-
pop");
        Playlist.addLagu("Viva La Vida", "Coldplay", 2008, 242,
"Alternative Rock");
        Playlist.addLagu("Bleeding Love", "Leona Lewis", 2007, 262,
"Pop");
        Playlist.addLagu("Rehab", "Amy Winehouse", 2006, 214, "Soul");
        Playlist.addLagu("Fix You", "Coldplay", 2005, 295, "Alternative
Rock");
        Playlist.addLagu("Hey Ya!", "Outkast", 2003, 235, "Hip-Hop");

        Playlist.showPlaylistSortingDurasi();
        Playlist.showPlaylistSortingTahunRilis();
    }
}
```

Script "`public class MainAsc{ dan public class MainDesc { `" Masing-masing kelas ini berfungsi untuk membuat daftar putar lagu (playlist) menggunakan kelas StackLagu dan menambahkan lagu dalam urutan berbeda, yaitu ascending (menaik) dan descending (menurun). Kedua kelas ini juga memanggil metode showPlaylistSortingDurasi() dan showPlaylistSortingTahunRilis() untuk menampilkan daftar lagu yang diurutkan berdasarkan durasi dan tahun rilis.

Script "`public class MainAsc{ `" mengelola playlist yang berisi lagu-lagu terkenal yang dimasukkan dalam urutan menaik berdasarkan waktu rilis (dari yang lama ke yang baru). Lagu-lagu dalam playlist ini mencakup judul seperti "Imagine" oleh John Lennon (1971), "Billie Jean" oleh Michael Jackson (1982), hingga "Shape of You" oleh Ed Sheeran (2017).