

MODUL III

SORTING AND SEARCHING

3.1 PERMASALAHAN

3.1.1 LOL Playlist

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.

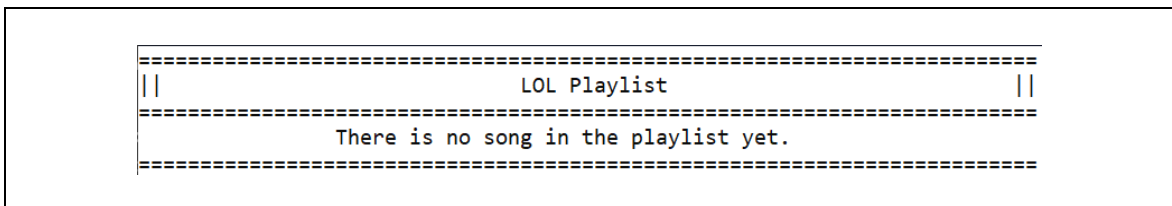
“Kenapa Sar? Biasanya langsung nyari stick, kok sekarang malah bengong?” tanya Rijal, penasaran.

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 sambil tertawa.



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

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

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

```

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

```

*** 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 urutan 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.”

=====	
	LOL Playlist
=====	
Judul lagu yang ingin dicari : We are	
Lagu yang sesuai dengan judul 'We are' ditemukan pada posisi ke - 7	
=====	
7	One Ok Rock - We are (2017)
4:15	Pop Rock
=====	

*** 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

3.2.1 LOL Playlist

1. Algoritma

- a. Buat data yang akan digunakan.
- b. Buat *double linked list* dengan *method add, print, sort* dan *search*.
- c. Tambahkan data dengan *method add*.
- d. Tampilkan data.
- e. Urutkan data dengan *bubble sort*, kemudian tampilkan data.
- f. Urutkan lagi data dengan aturan lain dengan *selection sort* dan tampilkan data.
- g. Coba cari data dengan *method search*.

2. Source Code

```
public class Musik {
    String lagu, artis, genre;
    int tahun, menit, detik;
    Musik next, prev;

    Musik(String lagu, String artis, String genre, int tahun, int
menit, int detik){
        this.lagu = lagu;
        this.artis = artis;
        this.genre = genre;
        this.tahun = tahun;
        this.menit = menit;
        this.detik = detik;
        this.prev = null; this.next = null;
    }
}
public class Playlist {
    Musik head, tail;

    public void add(String lagu, String artis,String genre, int
tahun, int menit, int detik){
        Musik NewMusik = new Musik(lagu, artis, genre, tahun,
menit, detik);
        if (head == null){
            head = NewMusik;
            tail = NewMusik;
        }else{
            tail.next = NewMusik;
            NewMusik.prev = tail;
            tail = NewMusik;
        }
    }

    public void print(){
```

```

        Musik current = head;
        int i = 1;

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

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

        if(head == null){
            System.out.println("\t\t\t\t\t There is no song in the
playlist yet.");
            return;
        }

        while(current != null){
            System.out.println("\t " + (i++) + ". " + current.artis
+ " - " + current.lagu + "(" + current.tahun + ")");
            System.out.println("\t " + current.menit + ":" +
current.detik + "\t" + current.genre);

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

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

        long startTime;
        long elapsedTime;
        startTime = System.nanoTime();

        Musik temp1, temp2;
        boolean swap;

        do {
            swap = false;
            temp1 = head;
            while (temp1.next != null) {
                temp2 = temp1.next;
                int waktu1detik = temp1.menit * 60 + temp1.detik,
waktu2detik = temp2.menit * 60 + temp2.detik;

                if (waktu1detik > waktu2detik) {
                    String tempLagu = temp1.lagu, tempArtis =
temp1.artis, tempGenre = temp1.genre;
                    int tempTahun = temp1.tahun, tempMenit =
temp1.menit, tempDetik = temp1.detik;

                    temp1.lagu = temp2.lagu;
                    temp1.artis = temp2.artis;
                    temp1.genre = temp2.genre;
                    temp1.tahun = temp2.tahun;
                    temp1.menit = temp2.menit;
                    temp1.detik = temp2.detik;

```

```

        temp2.lagu = tempLagu;
        temp2.artis = tempArtis;
        temp2.genre = tempGenre;
        temp2.tahun = tempTahun;
        temp2.menit = tempMenit;
        temp2.detik = tempDetik;

        swap = true;
    }
    temp1 = temp1.next;
}
} while (swap);
elapsedTime = System.nanoTime() - startTime;
System.out.println("Elapsed Time is " + (elapsedTime /
1000000.0)+ "msec");
}

public void Selection() {
    if (head == null || head.next == null) {
        return;
    }
    long startTime, elapsedTime;
    startTime = System.nanoTime();

    Musik temp1, temp2;
    int Year;
    String tempLagu, tempArtis, tempGenre;
    int tempTahun, tempMenit, tempDetik;

    for (temp1 = head; temp1 != null; temp1 = temp1.next) {
        temp2 = temp1.next;
        Year = temp1.tahun;

        while (temp2 != null) {
            if (temp2.tahun > Year) {
                Year = temp2.tahun;

                tempLagu = temp1.lagu;
                tempArtis = temp1.artis;
                tempGenre = temp1.genre;
                tempTahun = temp1.tahun;
                tempMenit = temp1.menit;
                tempDetik = temp1.detik;

                temp1.lagu = temp2.lagu;
                temp1.artis = temp2.artis;
                temp1.genre = temp2.genre;
                temp1.tahun = temp2.tahun;
                temp1.menit = temp2.menit;
                temp1.detik = temp2.detik;

                temp2.lagu = tempLagu;
                temp2.artis = tempArtis;
                temp2.genre = tempGenre;
                temp2.tahun = tempTahun;
                temp2.menit = tempMenit;
                temp2.detik = tempDetik;
            }
            temp2 = temp2.next;
        }
    }
}

```

```

    }
    elapsedTime = System.nanoTime() - startTime;
    System.out.println("Elapsed Time is " + (elapsedTime /
1000000.0)+ "msec");
}

public void search(String lagu){
    Musik current = head;
    int i = 1;
    long startTime, elapsedTime;
    startTime = System.nanoTime();
    if (head == null){
        System.out.println("\t\t\t\t\t There is no song in the
playlist yet.");
    }

    while (current != null){
        elapsedTime = System.nanoTime() - startTime;
        if(current.lagu.equalsIgnoreCase(lagu)){
            System.out.println("Judul lagu yang ingin dicari :
" + lagu);
            System.out.println("Lagu yang sesuai dengan judul
'" + lagu + "' ditemukan pada posisi ke - " + i);
            System.out.println("\t " + i + ". " + current.artis
+ " - " + current.lagu + "(" + current.tahun + ")");
            System.out.println("\t " + current.menit + ":" +
current.detik + "\t" + current.genre);

System.out.println("=====
=====");
            System.out.println("Elapsed Time is " +
(elapsedTime / 1000000.0)+ "msec");
            return;
        }
        current = current.next;
        ++i;
    }
}

}

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

        LOL.add("I Want to Write You a Song", "One Direction",
"Pop", 2015, 3, 22);
        LOL.add("Piano Man", "Billy Joel", "Folk", 1973, 5, 39);
        LOL.add("Die With A Smile", "Bruno Mars dan Lady Gaga",
"Pop", 2024, 4, 11);
        LOL.add("Nina", ".Feast", "Indonesian Rock", 2024, 4, 37);
        LOL.add("Bad", "Wave to Earth", "Korean Rock, Thai Indie",
2023, 4, 23);
        LOL.add("8 Letters", "Why Don't We", "Pop", 2018, 3, 10);
        LOL.add("There Is a Light That Never Goes Out", "The
Smiths", "Indie", 1986, 4, 4);
        LOL.add("To the Bone", "Pamungkas", "Indonesian Indie",
2020, 5, 44);
        LOL.add("We are", "One Ok Rock", "Pop Rock", 2017, 4, 15);
        LOL.add("Small girl", "Lee Young Ji", "R&B", 2024, 3, 9);
        LOL.add("21 Guns", "Green Day", "Punk Pop", 2009, 5, 21);
    }
}

```



```

        LOL.add("Little White Lies", "One Direction", "Pop", 2013,
3, 36);
        LOL.add("Uptown Funk", "Mark Ronson ft. Bruno Mars",
"Funk",2014, 4, 29);
        LOL.add("Over Again", "One Direction", "Pop", 2013, 3, 50);
        LOL.add("Imagine", "John Lennon", "Rock",1971, 3, 4);
        LOL.add("Walking in the Wind", "One Direction", "Pop",
2015, 3, 30);
        LOL.add("End of the Day", "One Direction", "Pop", 2015, 3,
22);
        LOL.add("Heroes", "David Bowie", "Rock", 1977, 3, 35 );
        LOL.add("Still the One", "One Direction", "Pop", 2014, 3,
5);
        LOL.add("I Would", "One Direction", "Pop", 2011, 3, 10);
        LOL.add("Fireflies", "Owl City", "Synth-pop", 2009, 3, 48);
        LOL.add("I Want to Break Free", "Queen", "Rock",1984, 4,
21);
        LOL.add("Girl Almighty", "One Direction", "Pop", 2014, 3,
34);
        LOL.add("Change Your Ticket", "One Direction", "Pop", 2014,
3, 40);
        LOL.add("Love You Goodbye", "One Direction", "Pop", 2015,
3, 34);
        LOL.add("Sweet Child O' Mine", "Guns N' Roses",
"Rock",1987, 5, 56);
        LOL.add("Through the Dark", "One Direction", "Pop", 2013,
3, 35);
        LOL.add("Little Things (Acoustic)", "One Direction", "Pop",
2013, 3, 37);
        LOL.add("Back for You", "One Direction", "Pop", 2013, 3,
20);
        LOL.add("Girls Just Want to Have Fun", "Cyndi Lauper",
"Pop", 1983, 3, 55);
        LOL.add("Gotta Be You", "One Direction", "Pop", 2011, 3,
5);
        LOL.add("I Want", "One Direction", "Pop", 2011, 2, 55);
        LOL.add("No Control (Acoustic)", "One Direction", "Pop",
2014, 3, 15);
        LOL.add("The Man Who Can't Be Moved", "The Script", "Pop
Rock", 2009, 3, 59);
        LOL.add("Up All Night", "One Direction", "Pop", 2011, 3,
17);
        LOL.add("The Story of My Life (Acoustic)", "One Direction",
"Pop", 2013, 4, 5);
        LOL.add("Nobody Else", "One Direction", "Pop", 2014, 3,
22);
        LOL.add("Temporary Fix", "One Direction", "Pop", 2015, 3,
14);
        LOL.add("She's Not Afraid", "One Direction", "Pop", 2012,
2, 55);
        LOL.add("Little White Lies (Acoustic)", "One Direction",
"Pop", 2013, 3, 36);
        LOL.add("No Control (Live)", "One Direction", "Pop", 2014,
3, 20);
        LOL.add("Where Do Broken Hearts Go", "One Direction",
"Pop", 2015, 3, 35);
        LOL.add("Creep", "Radiohead", "Rock", 1992, 3, 58);
        LOL.add("Love Me Like You Do", "One Direction", "Pop",
2015, 3, 22);
        LOL.add("Rehab", "Amy Winehouse", "Soul", 2006, 3, 34);

```

```

        LOL.add("Secret Love Song", "Taylor Swift", "Pop", 2015,
3, 35);
        LOL.add("Koiiro", "Musawo", "Pop", 2022, 3, 59);
        LOL.add("Lampu Kuning", "Juicy Luicy", "Pop", 2023, 4, 5);
        LOL.add("They Don't Know About Us", "One Direction", "Pop",
2012, 3, 5);
        LOL.add("Rock Me", "One Direction", "Pop", 2012, 2, 40);
        LOL.add("American Pie", "Don McLean", "Folk Rock", 1971, 8,
33);

        LOL.print();

        System.out.println("\n");

System.out.println("=====
=====");
        System.out.println("||                                BUBBLE
SORT                                ||");

System.out.println("=====
=====");
        LOL.sortbubble();
        LOL.print();

        System.out.println("\n");

System.out.println("=====
=====");
        System.out.println("||                                SELECTION
SORT                                ||");

System.out.println("=====
=====");
        LOL.Selection();
        LOL.print();

        System.out.println("\n");

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

System.out.println("=====
=====");
        LOL.search("We are");
    }
}

```

3.3 ANALISIS DATA

3.3.1 Antrian Pemusing Kepala

```
public class Musik {
    String lagu, artis, genre;
    int tahun, menit, detik;
    Musik next, prev;

    Musik(String lagu, String artis, String genre, int tahun, int menit,
    int detik){
        this.lagu = lagu;
        this.artis = artis;
        this.genre = genre;
        this.tahun = tahun;
        this.menit = menit;
        this.detik = detik;
        this.prev = null; this.next = null;
    }
}
```

Script “public class Musik” adalah *script* yang menyimpan data, yang akan digunakan untuk *double linked list* dan mengatur parameter “Musik” dengan “Musik(String lagu, String artis, String genre, int tahun, int menit, int detik)”.

```
public class Playlist {
    Musik head, tail;

    public void add(String lagu, String artis,String genre, int tahun,
    int menit, int detik){
        Musik NewMusik = new Musik(lagu, artis, genre, tahun, menit,
    detik);
        if (head == null){
            head = NewMusik;
            tail = NewMusik;
        }else{
            tail.next = NewMusik;
            NewMusik.prev = tail;
            tail = NewMusik;
        }
    }
}
```

Script “public class Playlist {” adalah *class* dengan beberapa *method* yang akan digunakan di program seperti “void add(String lagu, String artis,String genre, int tahun, int menit, int detik)” untuk menambah data di *double linked list*.

```
public void print(){
    Musik current = head;
    int i = 1;

    System.out.println("=====
    =====");
    System.out.println("|||                                LOL PLayerlist
    ||");
    System.out.println("=====
    =====")
}
```

```

=====");

        if(head == null){
            System.out.println("\t\t\t\t\t There is no song in the
playlist yet.");
            return;
        }

        while(current != null){
            System.out.println("\t " + (i++) + ". " + current.artis + "
- " + current.lagu + "(" + current.tahun + ")");
            System.out.println("\t " + current.menit + ":" +
current.detik + "\t" + current.genre);

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

```

Script “void print()” adalah *method* untuk menampilkan data yang ada pada *double linked list* seperti nama lagu, penyanyinya, tahun lagu rilis, *genre* lagu dan durasi lagu dalam menit dan detik.

```

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

    long startTime;
    long elapsedTime;
    startTime = System.nanoTime();

    Musik temp1, temp2;
    boolean swap;

    do {
        swap = false;
        temp1 = head;
        while (temp1.next != null) {
            temp2 = temp1.next;
            int waktu1detik = temp1.menit * 60 + temp1.detik,
            waktu2detik = temp2.menit * 60 + temp2.detik;

            if (waktu1detik > waktu2detik) {
                String tempLagu = temp1.lagu, tempArtis =
temp1.artis, tempGenre = temp1.genre;
                int tempTahun = temp1.tahun, tempMenit =
temp1.menit, tempDetik = temp1.detik;

                temp1.lagu = temp2.lagu;
                temp1.artis = temp2.artis;
                temp1.genre = temp2.genre;
                temp1.tahun = temp2.tahun;
                temp1.menit = temp2.menit;
                temp1.detik = temp2.detik;

                temp2.lagu = tempLagu;
                temp2.artis = tempArtis;
                temp2.genre = tempGenre;
            }
            temp1 = temp2;
        }
    } while (swap);
}

```

```

        temp2.tahun = tempTahun;
        temp2.menit = tempMenit;
        temp2.detik = tempDetik;

        swap = true;
    }
    temp1 = temp1.next;
}
} while (swap);
elapsedTime = System.nanoTime() - startTime;
System.out.println("Elapsed Time is " + (elapsedTime /
1000000.0)+ "msec");
}

```

Script “void sortbubble()” adalah *method* untuk mengurutkan data berdasarkan durasi lagu terpendek ke terpanjang. Pengurutan lagu dilakukan dengan cara memindahkan musik ke kanan bila durasi musik lebih panjang.

```

public void Selection() {
    if (head == null || head.next == null) {
        return;
    }
    long startTime, elapsedTime;
    startTime = System.nanoTime();

    Musik temp1, temp2;
    int Year;
    String tempLagu, tempArtis, tempGenre;
    int tempTahun, tempMenit, tempDetik;

    for (temp1 = head; temp1 != null; temp1 = temp1.next) {
        temp2 = temp1.next;
        Year = temp1.tahun;

        while (temp2 != null) {
            if (temp2.tahun > Year) {
                Year = temp2.tahun;

                tempLagu = temp1.lagu;
                tempArtis = temp1.artis;
                tempGenre = temp1.genre;
                tempTahun = temp1.tahun;
                tempMenit = temp1.menit;
                tempDetik = temp1.detik;

                temp1.lagu = temp2.lagu;
                temp1.artis = temp2.artis;
                temp1.genre = temp2.genre;
                temp1.tahun = temp2.tahun;
                temp1.menit = temp2.menit;
                temp1.detik = temp2.detik;

                temp2.lagu = tempLagu;
                temp2.artis = tempArtis;
                temp2.genre = tempGenre;
                temp2.tahun = tempTahun;
                temp2.menit = tempMenit;
                temp2.detik = tempDetik;
            }
            temp2 = temp2.next;
        }
    }
}

```

```

    }
}
elapsedTime = System.nanoTime() - startTime;
System.out.println("Elapsed Time is " + (elapsedTime /
1000000.0)+ "msec");
}

```

Script “void Selection()” adalah *method* untuk mengurutkan data berdasarkan tahun rilis terbaru. Pengurutan lagu dilakukan dengan cara pencarian lagu dengan tahun rilis terbaru, bila ditemukan maka pindah ke paling awal.

```

public void Selection() {
    if (head == null || head.next == null) {
        return;
    }
    long startTime, elapsedTime;
    startTime = System.nanoTime();

    Musik temp1, temp2;
    int Year;
    String tempLagu, tempArtis, tempGenre;
    int tempTahun, tempMenit, tempDetik;

    for (temp1 = head; temp1 != null; temp1 = temp1.next) {
        temp2 = temp1.next;
        Year = temp1.tahun;

        while (temp2 != null) {
            if (temp2.tahun > Year) {
                Year = temp2.tahun;

                tempLagu = temp1.lagu;
                tempArtis = temp1.artis;
                tempGenre = temp1.genre;
                tempTahun = temp1.tahun;
                tempMenit = temp1.menit;
                tempDetik = temp1.detik;

                temp1.lagu = temp2.lagu;
                temp1.artis = temp2.artis;
                temp1.genre = temp2.genre;
                temp1.tahun = temp2.tahun;
                temp1.menit = temp2.menit;
                temp1.detik = temp2.detik;

                temp2.lagu = tempLagu;
                temp2.artis = tempArtis;
                temp2.genre = tempGenre;
                temp2.tahun = tempTahun;
                temp2.menit = tempMenit;
                temp2.detik = tempDetik;
            }
            temp2 = temp2.next;
        }
    }
    elapsedTime = System.nanoTime() - startTime;
    System.out.println("Elapsed Time is " + (elapsedTime /
1000000.0)+ "msec");
}

```

Script “void Selection()” adalah *method* untuk mengurutkan data berdasarkan tahun rilis terbaru. Pengurutan lagu dilakukan dengan cara pencarian lagu dengan tahun rilis terbaru, bila ditemukan maka pindah ke paling awal.

```
public void search(String lagu){
    Musik current = head;
    int i = 1;
    long startTime, elapsedTime;
    startTime = System.nanoTime();
    if (head == null){
        System.out.println("\t\t\t\t\t There is no song in the
playlist yet.");
    }

    while (current != null){
        elapsedTime = System.nanoTime() - startTime;
        if(current.lagu.equalsIgnoreCase(lagu)){
            System.out.println("Judul lagu yang ingin dicari : " +
lagu);
            System.out.println("Lagu yang sesuai dengan judul '" +
lagu + "' ditemukan pada posisi ke - " + i);
            System.out.println("\t " + i + ". " + current.artis + "
- " + current.lagu + "(" + current.tahun + ")");
            System.out.println("\t " + current.menit + ":" +
current.detik + "\t" + current.genre);

            System.out.println("=====
=====");
            System.out.println("Elapsed Time is " + (elapsedTime /
1000000.0)+ "msec");
            return;
        }
        current = current.next;
        ++i;
    }
}
```

Script “void search()” adalah *method* pencarian data berdasarkan nama lagu, pencarian dilakukan dengan cara pengulangan dan pengecekan dari awal hingga akhir, bila data ditemukan akan ditampilkan data berkaitan lagu tersebut.

```
public class Main {
    public static void main(String[] args) {
        Playlist LOL = new Playlist();
        LOL.print();

        LOL.add("I Want to Write You a Song", "One Direction", "Pop",
2015, 3, 22);
        LOL.add("Piano Man", "Billy Joel", "Folk", 1973, 5, 39);
        LOL.add("Die With A Smile", "Bruno Mars dan Lady Gaga", "Pop",
2024, 4, 11);
        LOL.add("Nina", ".Feast", "Indonesian Rock", 2024, 4, 37);
        LOL.add("Bad", "Wave to Earth", "Korean Rock, Thai Indie", 2023,
4, 23);
        LOL.add("8 Letters", "Why Don't We", "Pop", 2018, 3, 10);
        LOL.add("There Is a Light That Never Goes Out", "The Smiths",
"Indie", 1986, 4, 4);
    }
}
```

```

44);
    LOL.add("To the Bone", "Pamungkas", "Indonesian Indie", 2020, 5,
    LOL.add("We are", "One Ok Rock", "Pop Rock", 2017, 4, 15);
    LOL.add("Small girl", "Lee Young Ji", "R&B", 2024, 3, 9);
    LOL.add("21 Guns", "Green Day", "Punk Pop", 2009, 5, 21);
    LOL.add("Little White Lies", "One Direction", "Pop", 2013, 3,
36);
    LOL.add("Uptown Funk", "Mark Ronson ft. Bruno Mars", "Funk", 2014,
4, 29);
    LOL.add("Over Again", "One Direction", "Pop", 2013, 3, 50);
    LOL.add("Imagine", "John Lennon", "Rock", 1971, 3, 4);
    LOL.add("Walking in the Wind", "One Direction", "Pop", 2015, 3,
30);
    LOL.add("End of the Day", "One Direction", "Pop", 2015, 3, 22);
    LOL.add("Heroes", "David Bowie", "Rock", 1977, 3, 35 );
    LOL.add("Still the One", "One Direction", "Pop", 2014, 3, 5);
    LOL.add("I Would", "One Direction", "Pop", 2011, 3, 10);
    LOL.add("Fireflies", "Owl City", "Synth-pop", 2009, 3, 48);
    LOL.add("I Want to Break Free", "Queen", "Rock", 1984, 4, 21);
    LOL.add("Girl Almighty", "One Direction", "Pop", 2014, 3, 34);
    LOL.add("Change Your Ticket", "One Direction", "Pop", 2014, 3,
40);
    LOL.add("Love You Goodbye", "One Direction", "Pop", 2015, 3, 34);
    LOL.add("Sweet Child O' Mine", "Guns N' Roses", "Rock", 1987, 5,
56);
    LOL.add("Through the Dark", "One Direction", "Pop", 2013, 3, 35);
    LOL.add("Little Things (Acoustic)", "One Direction", "Pop",
2013, 3, 37);
    LOL.add("Back for You", "One Direction", "Pop", 2013, 3, 20);
    LOL.add("Girls Just Want to Have Fun", "Cyndi Lauper", "Pop",
1983, 3, 55);
    LOL.add("Gotta Be You", "One Direction", "Pop", 2011, 3, 5);
    LOL.add("I Want", "One Direction", "Pop", 2011, 2, 55);
    LOL.add("No Control (Acoustic)", "One Direction", "Pop", 2014,
3, 15);
    LOL.add("The Man Who Can't Be Moved", "The Script", "Pop Rock",
2009, 3, 59);
    LOL.add("Up All Night", "One Direction", "Pop", 2011, 3, 17);
    LOL.add("The Story of My Life (Acoustic)", "One Direction",
"Pop", 2013, 4, 5);
    LOL.add("Nobody Else", "One Direction", "Pop", 2014, 3, 22);
    LOL.add("Temporary Fix", "One Direction", "Pop", 2015, 3, 14);
    LOL.add("She's Not Afraid", "One Direction", "Pop", 2012, 2, 55);
    LOL.add("Little White Lies (Acoustic)", "One Direction", "Pop",
2013, 3, 36);
    LOL.add("No Control (Live)", "One Direction", "Pop", 2014, 3,
20);
    LOL.add("Where Do Broken Hearts Go", "One Direction", "Pop",
2015, 3, 35);
    LOL.add("Creep", "Radiohead", "Rock", 1992, 3, 58);
    LOL.add("Love Me Like You Do", "One Direction", "Pop", 2015, 3,
22);
    LOL.add("Rehab", "Amy Winehouse", "Soul", 2006, 3, 34);
    LOL.add("Secret Love Song", "Taylor Swift", "Pop", 2015, 3, 35);
    LOL.add("Koiro", "Musawo", "Pop", 2022, 3, 59);
    LOL.add("Lampu Kuning", "Juicy Luicy", "Pop", 2023, 4, 5);
    LOL.add("They Don't Know About Us", "One Direction", "Pop", 2012,
3, 5);
    LOL.add("Rock Me", "One Direction", "Pop", 2012, 2, 40);
    LOL.add("American Pie", "Don McLean", "Folk Rock", 1971, 8, 33);
    LOL.print();

```



```

        System.out.println("\n");

System.out.println("=====
=====");
        System.out.println("|||                                BUBBLE SORT
||");

System.out.println("=====
=====");
        LOL.sortbubble();
        LOL.print();

        System.out.println("\n");

System.out.println("=====
=====");
        System.out.println("|||                                SELECTION SORT
||");

System.out.println("=====
=====");
        LOL.Selection();
        LOL.print();

        System.out.println("\n");

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

System.out.println("=====
=====");
        LOL.search("We are");
    }
}

```

Script “public class Main {” adalah *class* yang digunakan untuk menjalankan program dengan menggunakan *method* yang ada pada *class* “Playlist {” yaitu *method* *add*, *print*, *sort* dan *search*.