

**LAPORAN PROJEK UAS**  
**PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK**



**Nama Anggota:**

<b>Sholihat Briliana Marshush</b>	<b>2110511139</b>
<b>Annisa Hadyana Fadhilah</b>	<b>2110511142</b>
<b>Desi Ratnasari</b>	<b>2110511152</b>
<b>Amanda Najwa Perak Azizah</b>	<b>2110511158</b>

**Informatika**  
**Fakultas Ilmu Komputer**  
**Universitas Pembangunan Nasional Veteran Jakarta**  
**2022**

## **ABSTRAK**

Laporan ini disusun untuk memenuhi syarat penilaian Ujian Akhir Semester pada mata kuliah Program Berorientasi Objek. Pada kesempatan kali ini, kami membuat program pemutar musik sederhana yang kita beri nama “DENDANGIN”. Nama “DENDANGIN” kita pilih agar sesuai dengan fungsinya untuk mendengar musik yang berdendang, nama “DENDANGIN” menurut kelompok kami cukup unik dan menonjolkan arti bahwa program ini dibuat oleh Indonesia. Program ini menyediakan layanan kepada user untuk dapat memutar musik yang mereka inginkan. Di dalam program ini kami memasukan berbagai fitur kepada user agar menyempurnakan mereka dalam menggunakan program streaming musik ini. Dalam membangun program ini kami menggunakan bahasa java serta terdapat 495 baris code program yang sudah kita buat seefektif mungkin untuk menjalankan program “DENDANGIN” ini.

## **ABSTRACT**

*This report was prepared to meet the requirements for the Final Semester Examination in the Object Oriented Program course. On this occasion, we created a simple music player program which we named "DENDANGIN". We chose the name "DENDANGIN" to suit its function for listening to singing music, according to our group the name "DENDANGIN" is quite unique and emphasizes the meaning that this program was created by Indonesia. This program provides services to users to be able to play the music they want. In this program we include various features for users to perfect them in using this music streaming program. In building this program we use the Java language and there are 495 lines of program code that we have made as effective as possible to run this "DENDANGIN" program.*

## **PENDAHULUAN**

### **A. SKENARIO LATAR BELAKANG**

Brili adalah seorang gadis pecinta musik jazz yang berkuliah di UPN Veteran Jakarta. Ia sedang menikmati secangkir americano di tangan kirinya dan smartphone dengan program streaming service di tangan kanannya. Jarinya bergulir menelusuri playlist favoritnya yang berjudul “Harmony di Pagi Hari”. Namun hal tak terduga terjadi, Dia tidak percaya bahwa musisi favoritnya sudah tidak tersedia di platform streaming music! Namun, Brili melihat hal ini bukanlah sebagai akhir zaman, tetapi sebagai sebuah awal baru, suatu revolusi kaum jazz! Dia langsung mengendarai vespa kuningnya untuk bertemu nisa, manda dan desi untuk mendiskusikan suatu platform streaming service baru yang akan menjawab keresahannya.

### **B. TUJUAN**

Program pemutar musik sederhana ini kami buat untuk mengimplementasikan apa yang sudah kami pelajari selama 1 semester mengenai Bahasa Pemrogramman Java. Namun ada beberapa tujuan, diantaranya:

1. Menambah minat belajar mahasiswa tentang pembuatan program aplikasi bahasa pemrogramman java.
2. Memberikan gambaran kepada mahasiswa lain, bagaimana program pemutar musik jika dibuat dengan bahasa pemrogramman java.

## RANCANGAN

### 1. RANCANGAN CLASS

Rancangan class merupakan sebuah diagram yang menggambarkan hubungan antar class. Class dapat diartikan juga sebagai suatu blueprint atau cetakan untuk menciptakan suatu instant dari object. class juga merupakan grup suatu object dengan kemiripan attributes/properties, behaviour dan relasi ke object lain. Dalam project ini kami membuat class pada program dibawah ini

```
340     class Music implements Comparable<Music>{
341
342         3 usages
343         private String judul;
344         3 usages
345         private String artis;
346         3 usages
347         private String album;
348         3 usages
349         private int tahun;
350         5 usages
351         private boolean favorit;
352         2 usages
353         private static int howMuchMusic = 0;
354         2 usages
355         private int added;
356
357         Music(){
358
359     }
```

```
414     class SortByJudul implements Comparator<Music>{
415
416     @Override
417     public int compare(Music a, Music b){
418         if(a.getJudul().compareTo(b.getJudul()) > 0){
419             return 1;
420         }else if(a.getJudul().compareTo(b.getJudul()) < 0){
421             return -1;
422         }else{
423             if(a.getAdded() < b.getAdded()){
424                 return 1;
425             }else{
426                 return -1;
427             }
428         }
429     }
430 }
```

```
450     class SortByFavorit implements Comparator<Music>{
451     @Override
452     public int compare(Music a, Music b){
453         if(a.getFavorit() == true && b.getFavorit() == false){
454             return -1;
455         }else if(a.getFavorit() == false && b.getFavorit() == true){
456             return 1;
457         }else{
458             if(a.getAdded() < b.getAdded()){
459                 return -1;
460             }else{
461                 return 1;
462             }
463         }
464     }
465 }
```

```
466 class SortByAlbum implements Comparator<Music>{
467     public int compare(Music a, Music b){
468         if(a.getAlbum().compareTo(b.getAlbum()) > 0){
469             return 1;
470         }else if(a.getAlbum().compareTo(b.getAlbum()) < 0){
471             return -1;
472         }else{
473             if(a.getAdded() < b.getAdded()){
474                 return 1;
475             }else{
476                 return -1;
477             }
478         }
479     }
480 }
```

```
482 class SortByTahun implements Comparator<Music>{
483     public int compare(Music a, Music b){
484         if(a.getTahun() > b.getTahun()){
485             return 1;
486         }else if(a.getTahun() < b.getTahun()){
487             return -1;
488         }else{
489             if(a.getAdded() < b.getAdded()){
490                 return 1;
491             }else{
492                 return -1;
493             }
494         }
495     }
496 }
```

## 2. RANCANGAN POLYMORPHISME

Polimorfisme (bahasa inggris *polymorphism*) adalah sebuah prinsip dalam biologi di mana oraganisme atau spesies dapat memiliki banyak bentuk atau tahapan (*stages*). Polimorfisme dalam OOP adalah sebuah prinsip di mana class dapat memiliki banyak “bentuk” method yang berbeda-beda meskipun namanya sama. “Bentuk” di sini dapat kita artikan: isinya berbeda, parameternya berbeda, dan tipe datanya berbeda. Didalam program DENDANGIN ini kami menerapkan polymorphisme pada:

```
364     public void makeFav() { favorit = true; }
367
368     2 usages
369     public void deleteFav() { favorit = false; }
371
372     6 usages
373     public String getAlbum() { return album; }
375
376     9 usages
377     public boolean getFavorit() { return favorit; }
379
380     10 usages
381     public String getArtis() { return artis; }
383
384     10 usages
385     public String getJudul() { return judul; }
387
388     6 usages
389     public int getTahun() { return tahun; }
391
392     12 usages
393     public int getAdded() { return added; }
```

### 3. RANCANGAN INHERITANCE

Inheritance adalah suatu teknik yang menyatakan bahwa anak dari objek akan mewarisi data/atribut dan metode dari induknya langsung. Atribut dan metode dari objek induk diturunkan kepada anak objek, demikian seterusnya. Inheritance mempunyai arti bahwa atribut dan operasi akan dimiliki bersama di antarakeslas yang mempunyai hubungan secara hirarki. Suatu kelas dapat ditentukan secara umum, kemudian ditentukan spesifik menjadi subkelas. Setiap subkelas mempunyai hubungan atau mewarisi semua sifat yang dimiliki oleh kelas induknya, dan ditambah dengan sifat unik yang dimilikinya.

### 4. RANCANGAN ENKAPSULASI

Kami menerapkan rancangan enkapsulasi dengan cara menggunakan access modifier private pada attribute seluruh class. Untuk melindungi property (atribut) / method tertentu dari sebuah kelas agar tidak sembarang diakses dan dimodifikasi oleh suatu bagian program kami mengimplementasikan beberapa getter dan setter

Dibawah ini merupakan salah satu contoh rancangan enkapsulasi pada Program Dendangin, Perhatikan keyword “this” di bawah ini (lihat pada class Enkapsulasi). Untuk membedakan variabel alas pada parameter dan variabel judul pada atribut class Enkapsulasi, digunakanlah keyword “this”. Sehingga untuk menggunakan atribut judul pada class Enkapsulasi menggunakan this.judul. Sehingga hal ini berlaku pada atribut artis, album, serta artis.

```
354     □ Music(String judul, String artis, String album, int tahun){  
355         this.judul = judul;  
356         this.artis = artis;  
357         this.album = album;  
358         this.tahun = tahun;  
359         howMuchMusic++;  
360         added = howMuchMusic;  
361         favorit = false;  
362     }  
    □
```

## **Fitur - Fitur**

Di dalam program streaming music “DENDANGIN” ini terdapat beberapa fitur yaitu:

1. Next Music

Fitur ini digunakan untuk memutar musik selanjutnya dari musik yang diputar pada playlist.

2. Previous Music

Fitur ini digunakan untuk memutar musik sebelumnya dari musik yang diputar pada playlist.

3. Make Favorite

Fitur ini digunakan untuk menambahkan musik ke dalam daftar favorit.

4. Details Current Music

Fitur ini digunakan untuk menampilkan informasi tentang musik yang sedang diputar seperti artis, album, dan tahun.

5. Delete Current Music

Fitur ini digunakan untuk menghapus musik yang sedang diputar.

6. Search Music

Fitur ini digunakan untuk mencari musik yang diinginkan.

7. Add Music

Fitur ini digunakan untuk menambahkan musik ke dalam playlist.

8. Sort Playlist

Fitur ini digunakan untuk mengurutkan musik berdasarkan favorit, judul, artis, album, tahun dan musik yang baru saja ditambahkan.

9. Show All Music

Fitur ini digunakan untuk menampilkan seluruh musik.

Saat program dijalankan, user akan diminta untuk menambahkan minimal satu musik.

Oleh karena itu, program meminta detail sebuah musik dengan:

1. Judul

2. Artis

3. Album

4. Tahun rilis

## Penjelasan Code Program

```
1 | package Playlist;
```

Dibuat package bernama “Playlist”

```
3 | import java.util.*;
```

Mengimport “java.util.\*” untuk memasukan suatu method atau perintah dalam bahasa Pemrograman Java sehingga perintah tersebut sanggup dipakai atau dapat berfungsi.

```
5 ►  public class Playlist {  
6     static int pointer = 0;  
7     static Scanner in = new Scanner(System.in);  
8  
9     static ArrayList<Music> playlist = new ArrayList<>();  
10    static String sort = "Recently added";  
11    public static final String ANSI_CYAN = "\u001B[36m";  
12    public static final String ANSI_RESET = "\u001B[0m";  
13    public static final String ANSI_GREEN = "\u001B[32m";  
14    public static final String ANSI_YELLOW = "\u001B[0;33m";
```

Dibuat public class bernama “Playlist” dan pada line selanjutnya terdapat static yang merupakan perintah khusus sebuah method diakses langsung tanpa melalui object. Terdapat “static int pointer” yang menyimpan pointer di dalam program ini, kemudian ada “static Scanner in” untuk menyimpan fungsi scan/meminta inputan apabila dibutuhkan program, lalu ada “static ArrayList<Music> playlist” untuk menyimpan array dari music yang dimasukkan, lalu “static String sort” untuk menyimpan bahan sortingan di dalam playlist program DENDANGIN. Sedangkan public static final merupakan prosedur yang menghasilkan nilai tetap dan bersifat public yang dapat diakses dari prosedur atau fungsi manapun. Dari line 11-14 menyimpan variabel untuk mewarnai tulisan di dalam program.

Dibuat public static void sebuah method dengan nama main, yang mana method ini adalah method utama untuk menjalankan sebuah program Program DENDANGIN. Pada line 17-26 dibuat sebuah statement yang dapat di cetak/output program untuk menyambut user. Lalu pada line 20 akan masuk ke bagian perulangan apabila diinputkan "Y" dan akan masuk ke fungsi "addMusic" untuk menambahkan musik.

Pada line 45-53 terdapat sebuah statement untuk menampilkan bahwa musik di Program DENDANGIN siap diputar dan line 55 berisi deklarasi di awal variabel agar langsung masuk ke kondisi “1”. Selanjutnya akan masuk ke perulangan while dengan tambahan try dan catch yang berfungsi untuk mengetahui jenis error atau exception apa yang muncul.

```
71      else if(command == 3){
72          if(playlist.get(pointer).getFavorit() == false){
73              playlist.get(pointer).makeFav();
74          }else{
75              playlist.get(pointer).deleteFav();
76          }
77      }
78      else if(command == 4){
79          System.out.println("");
80          detailsCurrentMusic();
81      }
82      else if(command == 5){
83          System.out.println("");
84          deleteCurrentMusic();
85      }
86      else if(command == 6){
87          System.out.println("");
88          searchMusic();
89      }
90      else if(command == 7){
91          System.out.println("");
92          addMusic();
93      }
94      else if(command == 8){
95          sortMusic();
96      }
```

Pada line 71-77 akan masuk ke fungsi “makeFavorite” yang berfungsi untuk memasukkan suatu musik menjadi Favorite. Selanjutnya pada line 78-81 akan masuk ke fungsi “detailsCurrentMusic” yang berfungsi untuk menampilkan informasi music yang baru didengar, seperti tahun, album, dan penyanyi. Di line 82-85 terdapat fungsi “deleteCurrentMusic” berfungsi untuk menghapus music yang baru didengar. Pada line 86-89 terdapat fungsi “searchMusic” berfungsi untuk mencari suatu musik di Program DENDANGIN. Di line 90-93 terdapat fungsi “addMusic” berfungsi untuk menambahkan music yang anda inginkan. Dan pada line 94-96 terdapat fungsi “sortMusic” berfungsi untuk mengurutkan music berdasarkan judul, artis, ataupun tahun.

```

97     } else if(command == 9){
98         System.out.println("");
99         for(int i = 0; i < playlist.size(); i++){
100            System.out.println("Music "+ (i+1));
101            System.out.println(playlist.get(i));
102            System.out.println("");
103        }
104    }
105    else if(command == 0){
106        break;
107    }
108    else {
109        System.out.println("Sorry, the command you entered is wrong!");
110    }
111}
112
113 System.out.println("Thank you for using Dendangin!");
114

```

Di line 97-104 terdapat fungsi “playlist” berfungsi untuk menampilkan seluruh music. Lalu pada line 105-111 terdapat perintah break, jika memilih perintah 0 maka sistem akan berhenti dan akan menampilkan “sorry, the command you entered is wrong”. Selanjutnya jika perulangan telah selesai maka sistem akan menampilkan “thank you for using DENDANGIN”.

```

` 115 private static void searchMusic(){
116
117     System.out.print("Search Music\n");
118     System.out.print("Title : ");
119     String judul = in.nextLine();
120     System.out.print("Artist : ");
121     String artist = in.nextLine();
122
123     Music details = findMusic(judul, artist);
124
125     if(details != null){
126         System.out.println("\n-----");
127         System.out.println("Title : " + details.getJudul());
128         System.out.println("Artist: " + details.getArtis());
129         System.out.println("Album : " + details.getAlbum());
130         System.out.println("Year : " + details.getTahun());
131         System.out.println("-----");
132     } else{
133         System.out.println("Music not found.");
134     }
135 }
136

```

Pada line 115 dibuat sebuah fungsi bernama searchMusic. Di dalamnya berisi output untuk melakukan pencarian musik. Pada fungsi ini diminta untuk menginputkan judul musik dan nama

artis. Selanjutnya apabila judul dan artis yang dicari sesuai dengan apa yang ada di dalam sistem maka program akan menjalankan perintah untuk menampilkan detail musik yang sudah dicari, sedangkan jika tidak sesuai maka program akan menampilkan kata “Music not found.”

```
137     System.out.println("\n[1] Delete Music");
138     System.out.println("[2] Play Music");
139     if(details.getFavorit() == false) System.out.println("[3] Make Favorite");
140     else System.out.println("[3] Unfavorite");
141     System.out.println("[0] Back To Main Menu");
```

Pada line 137 - 141 diberikan beberapa pilihan untuk musik yang baru saja dicari. Di antaranya adalah delete music, play music, make favorite (jika musik dalam kondisi tidak favorit) atau unfavorite (jika musik sudah dalam kondisi favorit), dan yang terakhir yaitu back to main menu.

```
143     System.out.print("Command: ");
144     String command = in.nextLine();
145
146     if(command.equals("1")){
147         if(playlist.size() == 1){
148             System.out.println("Oops! You haven't added a song yet.");
149         }else{
150             System.out.println("Music deleted successfully!");
151             playlist.remove(details);
152         }
153         if(pointer == playlist.size()) pointer = 0;
154     }else if(command.equals("2")){
155         pointer = findMusicINT(judul, artist);
156     }else if(command.equals("3")){
157         if(details.getFavorit() == false) {
158             System.out.println("The music successfully added as a favorite!");
159             details.makeFav();
160         } else {
161             System.out.println("The music successfully deleted from favorite");
162             details.deleteFav();
163         }
164     }else if(command.equals("0")){
165         return;
166     }else{
167         System.out.println("Wrong command :(");
168         return;
169     }
170 }
```

Masuk ke line 143 akan mencetak kata “Command: “ , di line 144 program meminta inputan dari user. Kemudian line 146 masuk ke program kondisi if else, jika user menginputkan “1” akan masuk ke pilihan menghapus musik yang dipilih. Kemudian program akan masuk lagi ke kondisi if, jika didalam playlist hanya terdapat 1 musik saja maka program akan mencetak kata “Oops! You haven't added a song yet.” karena di dalam playlist minimal harus ada 1 musik. Dan apabila didalam playlist terdapat lebih dari 1 musik maka akan mencetak kata “Music deleted successfully!” maka musik yang dipilih berhasil dihapus dari playlist program.

```
172     private static void nextMusic() {
173
174         if(pointer+1 == playlist.size()){
175             pointer = 0;
176         }
177         else{
178             pointer++;
179         }
180
181     }
```

Pada line 172 - 181 menyimpan program untuk melakukan perintah melanjutkan ke musik berikutnya, di line 174 masuk ke kondisi if, apabila nilai pointer + 1 = banyaknya musik di dalam playlist, maka pointer berubah menjadi 0, selain dari kondisi tersebut nilai pointer akan bertambah 1.

```
184     private static void deleteCurrentMusic() {
185
186         if(playlist.size() == 1){
187             System.out.println("Failed to delete music.");
188         }
189
190         else{
191
192             playlist.remove(pointer);
193             if(pointer == playlist.size()) pointer = 0;
194             System.out.println("Music succesfully deleted.");
195         }
196     }
```

Selanjutnya, Public static void ini bernama deleteCurrentMusic yang berguna untuk menghapus musik yang baru saja diputar. Lalu, pada line selanjutnya terdapat kondisi if. Kondisi tersebut digunakan jika program hanya memiliki satu musik maka akan muncul statement “Failed to delete music” yang artinya gagal menghapus musik. Sedangkan, jika musik lebih dari 1 maka musik yang sedang diputar akan berhasil dihapus.

```
208     @
209     private static Music findMusic(String judul, String artist) {
210         for(int i = 0; i < playlist.size(); i++){
211             if(playlist.get(i).getJudul().equals(judul) && playlist.get(i).getArtis().equals(artist)){
212                 return playlist.get(i);
213             }
214         }
215     }
```

Private static yang bernama findMusic akan menggunakan bentuk perulangan java yaitu for. Pada line 209 berisi “int = 0” untuk menyimpan hitungan perulangan, dan “i++” agar dapat memutar musik-musik selanjutnya.

```
217     private static int findMusicINT(String judul, String artist) {
218         for(int i = 0; i < playlist.size(); i++){
219             if(playlist.get(i).getJudul().equals(judul) && playlist.get(i).getArtis().equals(artist)){
220                 return i;
221             }
222         }
223     }
224 }
```

Selanjutnya kami membuat private static yang bernama findMusicINT akan menggunakan perulangan java yaitu for. Private static int ini berfungsi untuk perulangan indexnya. Pada line 218 berisi “int = 0” untuk menyimpan hitungan perulangan, dan “i++” untuk terus memutar musik selanjutnya.

```
226     private static void prevMusic() {
227         if(pointer == 0){
228             pointer = playlist.size()-1;
229         }
230         else{
231             pointer--;
232         }
233     }
234 }
```

Selanjutnya terdapat private static yang bernama prevMusic yang terdapat pointer untuk menyimpan alamat memori. Pada pointer ini menggunakan playlist sebagai alamat, sehingga akan menampilkan musik sebelumnya.

```
236     private static void addMusic() {  
237  
238         System.out.println("Please enter your music!");  
239         System.out.print("Title : ");  
240         String judul = in.nextLine();  
241         System.out.print("Artist: ");  
242         String artist = in.nextLine();  
243         System.out.print("Album : ");  
244         String album = in.nextLine();  
245         System.out.print("Year : ");  
246         String tahun = in.nextLine();  
247  
248         try{  
249             playlist.add(new Music(judul, artist, album, Integer.parseInt(tahun)));  
250         }catch(NumberFormatException e){  
251             System.out.println("Failed to add music!");  
252         }  
253     }  
254 }
```

Pada line 236 - 254 terdapat fungsi addMusic. Fungsi ini digunakan untuk menambahkan musik ke dalam platform streaming musik ini. User diminta untuk menginput judul, artis, album, dan tahun dari musik yang akan ditambahkan. Apabila format yang dimasukan benar, seperti tahun yang diinput oleh data berupa integer, maka musik akan dimasukkan ke playlist. Sedangkan jika salah, maka program akan mengeluarkan output “Failed to add music!”

```
257     private static void display() {  
258         String displayedMusic = (playlist.get(pointer).getFavorit() == true) ? ANSI_CYAN+"* "+playlist.get(pointer).getJudul()+" - "  
259             +playlist.get(pointer).getArtis() + " *"+ANSI_RESET : ANSI_GREEN+playlist.get(pointer).getJudul()+" - "  
260             +playlist.get(pointer).getArtis()+ANSI_RESET;  
261         System.out.println();  
262  
263         int width = 103;  
264         String s = displayedMusic;  
265  
266         int padSize = width - s.length();  
267         int padStart = s.length() + padSize / 2;  
268  
269         s = String.format("%" + padStart + "s", s);  
270         s = String.format("%-" + width + "s", s);
```

Pada line 257 berisi fungsi display. Pada fungsi ini digunakan untuk menampilkan tampilan saat musik sedang diputar.

Pada line 272 - 288 yaitu memuat kode program untuk mencetak display ketika sebuah musik yang dipilih user sedang diputar.

```
290     private static void sortMusic(){
291         System.out.println("");
292         System.out.println("Sort playlist by: ");
293         System.out.println("[1] Favorite");
294         System.out.println("[2] Title");
295         System.out.println("[3] Artist");
296         System.out.println("[4] Album");
297         System.out.println("[5] Year");
298         System.out.println("[6] Recently added");
299         System.out.println("[7] Back to menu");
300         System.out.print("Command : ");
301         String command = in.nextLine();
302         if(command.equals("1")) {
303             Collections.sort(playlist, new SortByFavorit());
304             sort = "Favorite";
305         }
306         else if(command.equals("2")) {
307             Collections.sort(playlist, new SortByJudul());
308             sort = "Title";
309         }
310         else if(command.equals("3")) {
311             Collections.sort(playlist, new SortByArtis());
312             sort = "Artis";
313         }
314     }
```

```

314     }
315     else if(command.equals("4")) {
316         Collections.sort(playlist, new SortByAlbum());
317         sort = "Album";
318     }
319     else if(command.equals("5")) {
320         Collections.sort(playlist, new SortByTahun());
321         sort = "Year";
322     }
323     else if(command.equals("6")) {
324         Collections.sort(playlist);
325         sort = "Recently added";
326     }
327     else if(command.equals("7")) {
328         return;
329     }
330     else {
331         System.out.println("Wrong Command");
332         return;
333     }
334     pointer = 0;
335     System.out.println("Managed to sort by : "+ sort);
336     return;
337 }
338 }
```

Pada line 290 - 334 memuat kode program untuk mengurutkan musik dengan berbagai pilihan.

Di dalam program ini musik dapat diurutkan berdasarkan:

1. Jika menginputkan “1” maka musik akan diurutkan berdasarkan Favorit
2. Jika menginputkan “2” maka musik akan diurutkan berdasarkan Judul
3. Jika menginputkan “3” maka musik akan diurutkan berdasarkan Artis/Penyanyi
4. Jika menginputkan “4” maka musik akan diurutkan berdasarkan Album
5. Jika menginputkan “5” maka musik akan diurutkan berdasarkan Tahun
6. Jika menginputkan “6” maka musik akan diurutkan berdasarkan musik terakhir yang ditambahkan

Program juga memberikan pilihan kepada user apabila ingin balik ke menu dengan cara menginputkan “7”.

Dan apabila user menginputkan angka selain yang disebutkan dalam pilihan maka program akan mencetak kalimat “Wrong Command” atau yang berarti perintah yang dimasukan salah dan kemudian akan mengembalikan lagi ke bagian menu.

```
340     class Music implements Comparable<Music>{
341
342         3 usages
343             private String judul;
344             3 usages
345                 private String artis;
346                 3 usages
347                     private String album;
348                     3 usages
349                         private int tahun;
350                         5 usages
351                             private boolean favorit;
352                             2 usages
353                                 private static int howMuchMusic = 0;
354                                 2 usages
355                                     private int added;
356
357                                     Music(){
358
359                                         }
```

Pada line diatas terdapat class comparable menyediakan urutan penyortiran tunggal pada Program DENDANGIN dan pada line selanjutnya terdapat berbagai akses private variable yang merupakan kode yang sesuai dengan namanya, akses ini bersifat private. Dengan kata lain, data maupun method hanya dapat diakses oleh kelas.

```
354     class Music implements Comparable<Music>{
355         Music(String judul, String artis, String album, int tahun){
356             this.judul = judul;
357             this.artis = artis;
358             this.album = album;
359             this.tahun = tahun;
360             howMuchMusic++;
361             added = howMuchMusic;
362             favorit = false;
363         }
364     }
```

Pada line 350-362 berguna untuk menyimpan data musik yang sudah dimasukan dan akan muncul default favorite dengan kata false.

```
364     public void makeFav() { favorit = true; }
367
368     2 usages
369     public void deleteFav() { favorit = false; }
371
372     6 usages
373     public String getAlbum() { return album; }
375
376     9 usages
377     public boolean getFavorit() { return favorit; }
379
380     10 usages
381     public String getArtis() { return artis; }
383
384     10 usages
385     public String getJudul() { return judul; }
387
388     6 usages
389     public int getTahun() { return tahun; }
391
392     12 usages
393     public int getAdded() { return added; }
```

Line 364-365 terdapat class makeFav untuk menambahkan musik ke dalam favorite.

Line 367-371 terdapat class deleteFav untuk menghapus musik dari dalam favorite.

Line 372-375 terdapat class getAlbum untuk menyimpan musik di dalam playlist album.

Line 376-379 terdapat class getFavorite untuk meyimpan musik di dalam favorite.

Line 380-383 terdapat class getArtist untuk menyimpan artis di dalam artist.

Line 384-387 terdapat class getJudul untuk menyimpan judul di dalam judul.

Line 388-391 terdapat class getTahun untuk menyimpan tahun di dalam tahun.

Line 392 terdapat class getAdded untuk menambahkan musik di playlist.

```
396 ①@ public int compareTo(Music other){  
397     if(this.getAdded() < other.getAdded()){  
398         return 1;  
399     }else{  
400         return -1;  
401     }  
402 }  
403  
404 ②@ public String toString(){  
405     String temp = "-";  
406     if(favorit == true){  
407         temp = "Favorite";  
408     }  
409     return "Title : " + judul + "\n" + "Artist : " + artis + "\n" + "Album : " + album  
410     + "\n" + "Year : " + tahun + "\n" + "Favorite : " + temp;  
411 }  
412 }
```

Class untuk membandingkan musik yang ada, di line 397-402 masuk ke kondisi if dengan perintah this yang merupakan perintah khusus untuk mengakses musik yang sedang diputar dengan yang lainnya. Pada line 404-410 merupakan kode program untuk menampilkan data musik.

```
414 class SortByJudul implements Comparator<Music>{  
415  
416 ①@ public int compare(Music a, Music b){  
417     if(a.getJudul().compareTo(b.getJudul()) > 0){  
418         return 1;  
419     }else if(a.getJudul().compareTo(b.getJudul()) < 0){  
420         return -1;  
421     }else{  
422         if(a.getAdded() < b.getAdded()){  
423             return 1;  
424         }else{  
425             return -1;  
426         }  
427     }  
428 }  
429  
430 }
```

Pada line 414 - 430 berisi class SortByJudul. Pada class ini digunakan untuk mengurutkan musik berdasarkan judul musik sesuai urutan alphabet, di mana musik dengan judul berawalan A akan ditampilkan terlebih dahulu.

```
432 class SortByArtis implements Comparator<Music>{
433
434     public int compare(Music a, Music b){
435         if(a.getArtis().compareTo(b.getArtis()) > 0){
436             return 1;
437         }else if(a.getArtis().compareTo(b.getArtis()) < 0){
438             return -1;
439         }else{
440             if(a.getAdded() < b.getAdded()){
441                 return 1;
442             }else{
443                 return -1;
444             }
445         }
446     }
447
448 }
```

Pada line 432 - 448 terdapat class SortByArtis. Pada class ini digunakan untuk mengurutkan musik berdasarkan judul artis sesuai urutan alphabet, di mana musik dengan artis berawalan A akan ditampilkan terlebih dahulu.

```
450 class SortByFavorit implements Comparator<Music>{
451     public int compare(Music a, Music b){
452         if(a.getFavorit() == true && b.getFavorit() == false){
453             return -1;
454         }else if(a.getFavorit() == false && b.getFavorit() == true){
455             return 1;
456         }else{
457             if(a.getAdded() < b.getAdded()){
458                 return -1;
459             }else{
460                 return 1;
461             }
462         }
463     }
464 }
```

Di dalam line 450-464 terdapat code program yang berfungsi untuk mengurutkan musik

berdasarkan dengan favorite, jika musik a ada di daftar favorit maka musik a akan ditampilkan di urutan atas, sebaliknya jika musik b yang ada di daftar favorit maka musik b akan ditampilkan di urutan atas, jika kedua musik (a dan b) tidak termasuk ke daftar favorit maka akan diurutkan berdasarkan musik yang di inputkan terlebih dahulu oleh user.

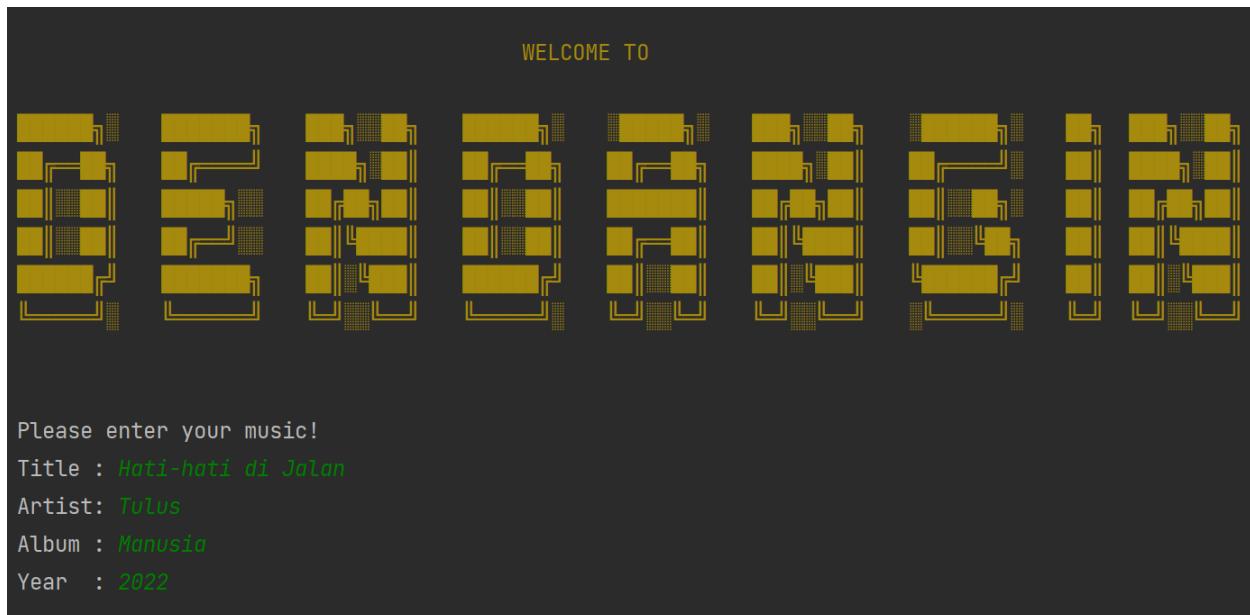
```
466 class SortByAlbum implements Comparator<Music>{
467     @Override
468     public int compare(Music a, Music b){
469         if(a.getAlbum().compareTo(b.getAlbum()) > 0){
470             return 1;
471         }else if(a.getAlbum().compareTo(b.getAlbum()) < 0){
472             return -1;
473         }else{
474             if(a.getAdded() < b.getAdded()){
475                 return 1;
476             }else{
477                 return -1;
478             }
479         }
480     }
}
```

Pada line ini terdapat pengurutan musik dengan class SortByAlbum. Pada class ini digunakan untuk mengurutkan musik berdasarkan judul artis sesuai urutan alphabet, di mana musik dengan album berawalan A akan ditampilkan terlebih dahulu.

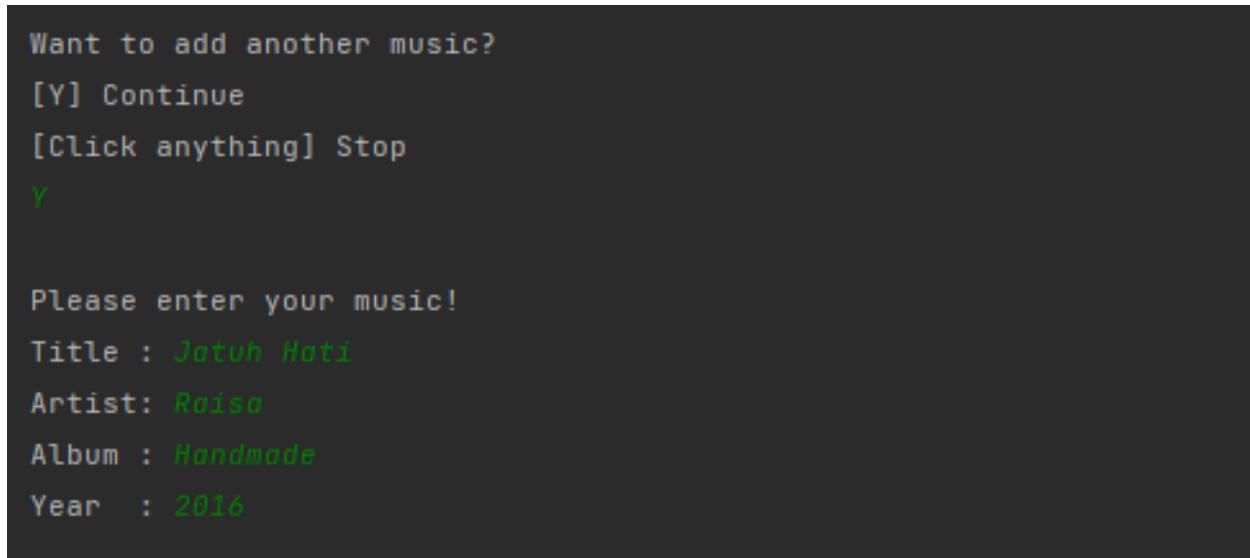
```
482 class SortByTahun implements Comparator<Music>{
483     @Override
484     public int compare(Music a, Music b){
485         if(a.getTahun() > b.getTahun()){
486             return 1;
487         }else if(a.getTahun() < b.getTahun()){
488             return -1;
489         }else{
490             if(a.getAdded() < b.getAdded()){
491                 return 1;
492             }else{
493                 return -1;
494             }
495         }
496     }
}
```

Terdapat class SortByTahun, pada class ini digunakan untuk mengurutkan musik berdasarkan judul musik sesuai urutan tahun, di mana musik dengan tahun lebih kecil ditampilkan terlebih dahulu

## OUTPUT PROGRAM



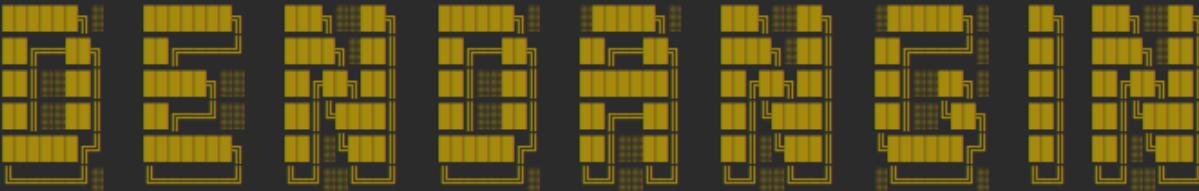
Pada saat pertama kali dijalankan, DENDANGIN akan menyambut user dengan kata “Welcome to DENDANGIN” dan user dipersilahkan mengisi data musik yang ingin disimpan karena minimal harus memiliki 1 musik pada program.



```
Want to add another music?  
[Y] Continue  
[Click anything] Stop  
Y  
  
Please enter your music!  
Title : Evaluasi  
Artist: Hindia  
Album : Menari Dengan Bayangan  
Year : 2019
```

```
Want to add another music?  
[Y] Continue  
[Click anything] Stop  
Y  
  
Please enter your music!  
Title : Ragu  
Artist: Rizky Febian  
Album : Jejak  
Year : 2018  
  
Want to add another music?  
[Y] Continue  
[Click anything] Stop  
Y  
  
Please enter your music!  
Title : Dunia Tipu-Tipu  
Artist: Yura Yunita  
Album : Tutur Botin  
Year : 2021
```

Selanjutnya user diberikan dua pilihan untuk menambahkan musik lainnya. Jika ingin melanjutkan untuk menambahkan musik, maka user diminta untuk mengetik “Y”, sedangkan jika tidak, user diminta untuk mengetik apa saja. Pada tahap ini dicoba untuk menambahkan musik lagi dengan menginput “Y”

```
Want to add another music?  
[Y] Continue  
[Click anything] Stop  
0  
  
DENDANGIN IS READY!  
  
=====  
Currently Playing  
Hati-hati di jalan - Tulus  
00:01 ━━━━━━━━ 04:05  
↶ ||< ─ ─ >|| ⏪  
=====  
Playlist sorted by: Recently added  
  
[1] Next Music [6] Search Music  
[2] Previous Music [7] Add Music  
[3] Make Favorite [8] Sort Playlist  
[4] Details Current Music [9] Show All Music  
[5] Delete Current Music  
Command (0 for exit): 1
```

Tampilan saat musik diputar Program DENDANGIN. User diberikan beberapa fitur yang bisa digunakan pada program.

```
Playlist sorted by: Recently added  
  
[1] Next Music [6] Search Music  
[2] Previous Music [7] Add Music  
[3] Make Favorite [8] Sort Playlist  
[4] Details Current Music [9] Show All Music  
[5] Delete Current Music  
Command (0 for exit): 1  
  
=====  
Currently Playing  
Jatuh Hati - Raisa  
00:01 ━━━━━━━━ 04:05  
↶ ||< ─ ─ >|| ⏪  
=====
```

Ketika proses penambahan musik sudah diberhentikan, program akan otomatis memutar musik berdasarkan musik yg paling pertama ditambahkan, dan apabila menginputkan “1” di program maka akan masuk ke program melanjutkan musik, pada gambar diatas terlihat setelah musik “Hati-hati di Jalan” akan memutar musik “Jatuh Hati” seperti urutan ketika musik tersebut ditambahkan di program awal.

```
[1] Next Music          [6] Search Music
[2] Previous Music     [7] Add Music
[3] Make Favorite      [8] Sort Playlist
[4] Details Current Music [9] Show All Music
[5] Delete Current Music
Command (0 for exit): 2

=====
                                Currently Playing
        Hati-hati di jalan - Tulus
        00:01 ━━━━━━━━━━━━━━━━ 04:05
        ↻ ||◀ ▶|| ⏪
=====
```

Pada bagian ini user dapat memutar musik sebelumnya dengan fitur yang tersedia pada Program DENDANGIN yaitu “Previous Music”. Oleh karena itu, user cukup mengisi perintah dengan angka 2. Maka Program DENDANGIN akan memutar musik sebelumnya.

```
[1] Next Music          [6] Search Music
[2] Previous Music     [7] Add Music
[3] Make Favorite      [8] Sort Playlist
[4] Details Current Music [9] Show All Music
[5] Delete Current Music
Command (0 for exit): 3

=====
                                Currently Playing
        * Hati-hati di jalan - Tulus *
        00:01 ━━━━━━━━━━━━━━━━ 04:05
        ↻ ||◀ ▶|| ⏪
=====
```

Pada bagian ini user juga bisa membuat playlist berdasarkan “Favorite” yang berisi musik-musik favorit Anda. Caranya pun sangat mudah, user hanya mengisi perintah pada Program DENDANGIN dengan angka 3. Ketika musik tersebut sudah favorit maka akan berwarna biru pada saat diputar.

```
=====
                    Currently Playing
* Hati-hati di jalan - Tulus *
00:01 ━━━━━━━━━━━━━━━━ 04:05
← | |< ▶| | ⌂
=====

Playlist sorted by: Recently added

[1] Next Music          [6] Search Music
[2] Previous Music      [7] Add Music
[3] Unfavorite          [8] Sort Playlist
[4] Details Current Music [9] Show All Music
[5] Delete Current Music
Command (0 for exit): 3

=====
                    Currently Playing
Hati-hati di jalan - Tulus
00:01 ━━━━━━━━━━━━━━━━ 04:05
← | |< ▶| | ⌂
=====
```

Pada pilihan selanjutnya yaitu “Unfavorite”. Sebagai contoh yaitu pada musik “Hati-hati di jalan - Tulus” yang merupakan musik Favorite, kini diubah menjadi musik dengan kategori tidak favorite.

```
=====
                    Currently Playing
Hati-hati di jalan - Tulus
00:01 ━━━━━━━━━━━━━━━━ 04:05
← | |< ▶| | ⌂
=====

Playlist sorted by: Recently added

[1] Next Music          [6] Search Music
[2] Previous Music      [7] Add Music
[3] Make Favorite        [8] Sort Playlist
[4] Details Current Music [9] Show All Music
[5] Delete Current Music
Command (0 for exit): 4

Music Data:
Title : Hati-hati di jalan
Artist: Tulus
Album : Manusia
Year  : 2022
```

Pada pilihan selanjutnya yaitu “Details Current Music”. Digunakan untuk melihat detail musik dari musik yang baru saja atau terakhir diputar.

```
=====
                    Currently Playing
                    Hati-hati di jalan - Tulus
00:01 ━━━━━━━━━━━━━━━━ 04:05
      ↻ ||< ──▶||   ⏴
=====

Playlist sorted by: Recently added

[1] Next Music          [6] Search Music
[2] Previous Music     [7] Add Music
[3] Make Favorite       [8] Sort Playlist
[4] Details Current Music [9] Show All Music
[5] Delete Current Music
Command (0 for exit): 5

Music successfully deleted.

=====
                    Currently Playing
                    Jatuh Hati - Raisa
00:01 ━━━━━━━━━━━━━━━━ 04:05
      ↻ ||< ──▶||   ⏴
=====
```

Jika menginputkan “5” pada halaman menu, maka akan masuk ke program menghapus musik yang sedang diputar, pada gambar diatas, program sedang memutar musik “Hati-hati di jalan - Tulus” maka musik tersebut akan terhapus dari playlist dan menampilkan kalimat “Music successfully deleted” yang menandakan musik tersebut berhasil dihapus.

```
[1] Next Music          [6] Search Music
[2] Previous Music     [7] Add Music
[3] Make Favorite       [8] Sort Playlist
[4] Details Current Music [9] Show All Music
[5] Delete Current Music
Command (0 for exit): 6

Search Music
Title : Hati-hati di jalan
Artist : Tulus
Music not found.
```

Jika menginputkan “6” maka akan masuk ke program mencari musik, namun karena musik Hati-hati di jalan pada sebelumnya sudah dihapus, maka jika dicari tidak akan ada dan muncul kata “Music not found”.

```
[1] Next Music          [6] Search Music
[2] Previous Music     [7] Add Music
[3] Make Favorite      [8] Sort Playlist
[4] Details Current Music [9] Show All Music
[5] Delete Current Music
Command (0 for exit): 7

Please enter your music!
Title : Monolog
Artist: Pamungkas
Album : Walk The Talk
Year  : 2018
```

Pada bagian ini user memilih Add Music, yang dimana user bisa menambahkan musik dengan menuliskan judul musik, penyanyi, album dan tahun rilisnya.

```
[1] Next Music          [6] Search Music
[2] Previous Music     [7] Add Music
[3] Make Favorite      [8] Sort Playlist
[4] Details Current Music [9] Show All Music
[5] Delete Current Music
Command (0 for exit): 8

Sort playlist by:
[1] Favorite
[2] Title
[3] Artist
[4] Album
[5] Year
[6] Recently added
[7] Back to menu
Command : |
```

Pada bagian ini user memilih option 8, yang dimana user dapat mengurutkan musik berdasarkan Favorite, judul musik, penyanyi, album, tahun, atau terakhir dimainkan.

```
Sort playlist by:  
[1] Favorite  
[2] Title  
[3] Artist  
[4] Album  
[5] Year  
[6] Recently added  
[7] Back to menu  
Command : 1  
Managed to sort by : Favorite  
  
=====  
Currently Playing  
* Ragu - Rizky Febian *  
00:01 ━━━━━━━━ 04:05  
← ||◀ ▶|| ▶  
=====
```

Program DENDANGIN menyediakan fitur pengurutan playlist berdasarkan favorite. Fitur ini tentunya dapat menghemat waktu pengguna untuk menemukan musik favorite yang mereka sedang ingin putarkan.

```
Sort playlist by:  
[1] Favorite  
[2] Title  
[3] Artist  
[4] Album  
[5] Year  
[6] Recently added  
[7] Back to menu  
Command : 5  
Managed to sort by : Year  
  
=====  
Currently Playing  
Jatuh Hati - Raisa  
00:01 ━━━━━━━━ 04:05  
← ||◀ ▶|| ▶  
=====
```

Menu “Sort Playlist” digunakan untuk mengurutkan musik. Diberikan pilihan yaitu favorite, title, artist, album, year, recently added, dan back to menu. Pada contoh ini dipilih sort berdasarkan tahun, di mana musik yang akan diputar yaitu “Jatuh Hati - Raisa” karena musik ini dirilis lebih dahulu daripada musik lain.

```
Playlist sorted by: Year

[1] Next Music           [6] Search Music
[2] Previous Music       [7] Add Music
[3] Make Favorite        [8] Sort Playlist
[4] Details Current Music [9] Show All Music
[5] Delete Current Music
Command (0 for exit): 9
```

```
Music 1
Title : Jatuh Hati
Artist : Raisa
Album : Handmade
Year : 2016
Favorite : -
```

```
Music 2
Title : Monolog
Artist : Pamungkas
Album : Walk The Talk
Year : 2018
Favorite : Favorite
```

```
Music 3
Title : Ragu
Artist : Rizky Febian
Album : Jejak
Year : 2018
Favorite : Favorite
```

```
Music 4
Title : Dunia Tipu-Tipu
Artist : Yura Yunita
Album : Tutur Batin
Year : 2021
Favorite : -
```

Setelah musik di sorting berdasarkan tahun, apabila diinputkan “9” maka akan menampilkan daftar musik yang tahun rilisnya lebih awal, dan menampilkan daftar seperti diatas.

Berikut link: [VIDEO PRESENTASI KELOMPOK 3](#)