

# **LAPORAN TUGAS BESAR**

## **IF2111 Algoritma dan Struktur Data STI**

### **Wayang Wave**


Dipersiapkan oleh:

Kelompok 07

1. Muhammad Rafly 18222067
2. Dzulfaqor Ali D 18222017
3. Willhelmina Rachel S 18222049
4. Yoga Putra Pratama 18222073
5. Habib Akhmad Al F 18222029

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	<b>Sekolah Elektro Informatika ITB</b>	<b>Teknik dan</b>	<b>Nomor Dokumen</b>		<b>Halaman</b>
					<b>Pertama</b>
			<i>IF2111-TB-01-07</i>		98
			<i>Revisi</i>	<i>01</i>	24 - 11 - 2023

# Daftar Isi

<b>Ringkasan.....</b>	<b>5</b>
1 Penjelasan Tambahan Spesifikasi Tugas.....	7
1.1 Tampilan Banner.....	7
1.2 Enhance.....	8
2 Struktur Data (ADT).....	9
2.1 ADT List.....	9
2.2 ADT Mesin Karakter dan Mesin Kata.....	11
2.3 ADT Queue.....	15
2.4 ADT Stack.....	17
2.5 ADT Set dan Map.....	18
2.6 ADT List dengan Struktur Berkait.....	20
2.7 ADT Mesin File dan Mesin Input.....	22
3 Program Utama.....	25
4 Algoritma-Algoritma Menarik.....	37
4.1 ADT List Dinamik x Struktur Berkait.....	37
4.2 Enhance.....	41
5 Data Test.....	45
5.1 Data Test START.....	46
5.2 Data Test LOAD.....	46
5.3 Data Test LIST.....	47
6.3.1 Data Test List Default.....	47
6.3.2 Data Test List Playlist.....	48
5.4 Data Test PLAY.....	49
6.4.1 Data Test Play Song.....	49
6.4.2 Data Test Play Playlist.....	50
5.5 Data Test QUEUE.....	51
6.5.1 Data Test Queue Song.....	51
6.5.2 Data Test Queue Playlist.....	52
6.5.3 Data Test Queue Swap.....	53
6.5.4 Data Test Queue Remove.....	54
6.5.5 Data Test Queue Clear.....	54
5.6 Data Test SONG.....	55
6.6.1 Data Test Song Next.....	55
6.6.2 Data Test Song Previous.....	56

5.7 Data Test PLAYLIST.....	58
6.7.1 Data Test Playlist Create.....	58
6.7.2 Data Test Playlist Add Song.....	58
6.7.3 Data Test Playlist Add Album.....	60
6.7.4 Data Test Playlist Swap.....	60
6.7.5 Data Test Playlist Remove.....	61
6.7.6 Data Test Playlist Delete.....	63
5.8 Data Test STATUS.....	63
5.9 Data Test SAVE.....	64
5.10 Data Test QUIT.....	65
5.11 Data Test HELP.....	66
6.11.1 Data Test Help Sebelum Masuk.....	66
6.11.2 Data Test Help Setelah Masuk.....	66
6 Test Script.....	67
7 Pembagian Kerja dalam Kelompok.....	75
Tabel 1 Daftar Pembagian Kerja Program.....	75
<b>8 Lampiran.....</b>	<b>77</b>
8.1 Deskripsi Tugas Besar.....	77
<b>Spesifikasi Umum.....</b>	<b>77</b>
<b>System Mechanic.....</b>	<b>77</b>
1. About the System.....	77
2. Main Menu.....	77
3. Command.....	77
a. START.....	77
b. LOAD <filename>.....	78
c. LIST.....	78
d. PLAY.....	80
e. QUEUE.....	81
f. SONG.....	84
g. PLAYLIST.....	84
h. STATUS.....	90
i. SAVE <filename>.....	92
j. QUIT.....	92
k. HELP.....	92
l. <INVALID COMMAND>.....	93
8.2 Notulen Rapat.....	94

8.3 Log Activity Anggota Kelompok.....	98
--	----

## Ringkasan

Wayang Wave adalah sebuah aplikasi yang digunakan untuk melakukan streaming musik, memberikan akses tanpa batas kepada beragam lagu dari berbagai genre dan artis. Wayang Wave dilengkapi dengan beberapa fitur utama, termasuk:

1. Memutar lagu
2. Menampilkan daftar lagu
3. Membuat dan menghapus playlist
4. Mengatur urutan pemutaran lagu
5. Menampilkan status dari aplikasi

Berdasarkan deskripsi tersebut, kami mengembangkan program streaming musik dengan berbagai fitur seperti start, load, list, play, queue, song, playlist, status, save, quit, help, dan invalid command.

Dalam tugas besar ini, Kelompok 07 – K01 telah membuat aplikasi simulasi berbasis antarmuka baris perintah (CLI) dalam bahasa C dengan memanfaatkan struktur data yang telah dipelajari dalam mata kuliah IF2111 – Algoritma dan Struktur Data STI.

Aplikasi ini akan memunculkan menu utama yang berisi opsi start, load dan juga help. Di aplikasi ini, pengguna dapat memilih beberapa perintah, antara lain list default untuk melihat daftar penyanyi yang tersedia dimana nantinya dapat memilih untuk melihat album dari penyanyi yang akan dipilih, list playlist untuk melihat daftar playlist yang ada pada pengguna, play song untuk memutar lagu dengan memasukkan nama penyanyi, nama album serta ID lagunya, play playlist untuk memutar lagu berdasarkan ID playlist, queue song untuk menambahkan lagu ke dalam antrian, queue playlist untuk menambahkan lagu dari playlist ke dalam antrian, queue swap untuk menukar lagu pada urutan ke-x dan juga urutan ke-y, queue remove untuk menghapus lagu dari antrian dan queue clear untuk mengosongkan antrian

Dalam aplikasi ini, pengguna juga dapat menggunakan navigasi pada lagu yang sedang berjalan, yaitu song next untuk memutar lagu berikutnya dalam antrian dan song previous untuk memutar lagu yang terakhir kali dimainkan,

Pengguna juga diberikan kebebasan untuk melakukan basic command pada playlistnya, seperti playlist create untuk membuat playlist baru dan menambahkannya

ke daftar playlist pengguna, playlist add untuk menambahkan lagu ke playlist yang sudah ada dalam daftar playlist pengguna, playlist swap untuk menukar lagu pada urutan x dan urutan y dalam playlist dengan ID tertentu, playlist remove untuk menghapus lagu pada urutan n dalam playlist dengan indeks ID tertentu, playlist delete untuk menghapus playlist yang sudah ada dalam daftar playlist pengguna,

Selain itu, ada beberapa fitur lainnya di dalam aplikasi WayangWave ini yaitu status untuk menampilkan lagu yang sedang dimainkan beserta daftar lagu dalam antrian dan playlist mana lagu itu dimainkan, save untuk menyimpan state aplikasi terbaru ke file, quit untuk keluar dari aplikasi WayangWave, dan yang terakhir help untuk menampilkan daftar perintah yang dapat dieksekusi dengan deskripsi.

Pengerjaan aplikasi ini juga dituang dalam bentuk laporan yang melibatkan penguraian umum mengenai tantangan yang dihadapi oleh aplikasi Wayang Wave, penjelasan tambahan mengenai spesifikasi tugas, penjelasan mengenai struktur data (ADT), struktur program-program dasar, algoritma-algoritma menarik yang ditemukan selama pelaksanaan tugas besar ini, uraian saat melaksanakan uji coba data, skenario uji yang mencakup seluruh fitur yang ada, alokasi tugas, dan penambahan dokumen seperti deskripsi tugas besar, catatan rapat, log aktivitas, dan sejenisnya.

# 1 Penjelasan Tambahan Spesifikasi Tugas

## 1.1 *Tampilan Banner*



Pada halaman awal saat membuka WayangWave, kami memasukkan tampilan sambutan "Welcome to WayangWave." Meskipun bukan suatu fitur, namun tampilan ini bertujuan merepresentasikan aplikasi WayangWave. Di bagian bawahnya, terdapat menu bantuan yang memberikan informasi kepada pengguna mengenai perintah START dan LOAD.

## 1.2 Enhance

```
>> ENHANCE;

Playlist Anda:
  1. BLACKPINK My Top Three
  2. Arctic Monkeys My Top 5
  3. Mixed Playlist

Pilih ID playlist yang ingin di-enhance: 2;

Berikut adalah lagu-lagu rekomendasi dari penyanyi Arctic Monkeys dalam album Humbug :
  1. Arctic Monkeys - Humbug - My Propeller
  2. Arctic Monkeys - Humbug - Crying Lightning
  3. Arctic Monkeys - Humbug - Secret Door

Apakah Anda ingin memasuki lagu rekomendasi dalam playlist? (Y/N): Y;
ID lagu berapa yang ingin ditambahkan?: 1;

Lagu My Propeller sudah dimasukkan ke dalam playlist.

Berikut adalah lagu-lagu rekomendasi dari penyanyi Arctic Monkeys dalam album Humbug :
  1. Arctic Monkeys - Humbug - My Propeller
  2. Arctic Monkeys - Humbug - Crying Lightning
  3. Arctic Monkeys - Humbug - Secret Door

Apakah Anda ingin memasuki lagu rekomendasi dalam playlist? (Y/N): Y;
ID lagu berapa yang ingin ditambahkan?: 3;

Lagu Secret Door sudah dimasukkan ke dalam playlist.

Berikut adalah lagu-lagu rekomendasi dari penyanyi Arctic Monkeys dalam album Humbug :
  1. Arctic Monkeys - Humbug - My Propeller
  2. Arctic Monkeys - Humbug - Crying Lightning
  3. Arctic Monkeys - Humbug - Secret Door

Apakah Anda ingin memasuki lagu rekomendasi dalam playlist? (Y/N): N;
```

Fungsi Enhance adalah bagian dari bonus. Enhance akan meminta playlist yang ingin di-enhance terlebih dahulu. Jika id playlist input user tidak valid, maka akan keluar dari enhance. Jika valid, program akan menampilkan album random. Kemudian user dapat memilih lagu hasil rekomendasi yang ingin ditambahkan kepada playlist. Jumlah lagu minimal pada rekomendasi adalah 2, jika tidak ditemukan setelah 10 kali random attempt, enhance akan gagal.



## 2 Struktur Data (ADT)

Di bawah ini akan dijelaskan ADT apa saja yang digunakan dan peruntukannya. Selain itu akan dijelaskan juga sketsanya.

### 2.1 ADT List

ADT ini digunakan dengan implementasi array yang terbagi berdasarkan tipe pengalokasiannya, yaitu statis dan dinamis. List statis digunakan untuk merepresentasikan daftar penyanyi yang terdapat dalam aplikasi. Sedangkan list dinamis digunakan untuk merepresentasikan daftar playlist yang mungkin ada dalam aplikasi. Untuk adt ini, pada bagian header diberi nama `listdinamik.h` dan `liststatik.h`. Pada bagian source code nya mengikuti penamaan header

**Struktur data yang Kita gunakan adalah List Statik**, yang merupakan struktur data linier dengan ukuran tetap. Struktur data ini digunakan untuk merepresentasikan daftar penyanyi dalam aplikasi Kita.

**Sketsa Struktur Data:** List Statik Kita terdiri dari array of Word dan Neff (jumlah elemen efektif). Array ini memungkinkan Kita untuk menyimpan banyak Word (yang mungkin mewakili penyanyi) dalam satu list.

**Persoalan yang Diselesaikan:** List Statik ini memungkinkan Kita untuk menyimpan, mengakses, dan memanipulasi daftar penyanyi dalam aplikasi Kita. Kita dapat dengan mudah menambahkan penyanyi ke list atau mencari penyanyi dalam list.

**Alasan Pemilihan:** List Statik dipilih karena sederhana dan efisien untuk kasus di mana jumlah penyanyi tidak akan berubah secara signifikan, sehingga tidak perlu alokasi dan dealokasi memori secara dinamis.

**Implementasi sebagai ADT:** List Statik ini diimplementasikan sebagai Abstract Data Type (ADT) karena menyembunyikan detail implementasi dari pengguna. Pengguna hanya perlu tahu tentang operasi yang dapat dilakukan pada List Statik (seperti `CreateLS`, `InsertLS`, `SearchLS`, dll.) tanpa harus tahu bagaimana operasi tersebut diimplementasikan. Ini membuat List Statik lebih mudah digunakan dan dipahami, serta memungkinkan Kita untuk mengubah implementasi di belakang layar tanpa mengganggu kode yang menggunakan ADT ini.

**Pengimplementasian ADT ini terhadap fitur yang digunakan:**

**Start:** untuk menginisialisasi daftar penyanyi saat aplikasi pertama kali dijalankan.

**Load:** untuk memuat daftar penyanyi dari file saat aplikasi dijalankan.

**Play Song:** untuk mencari penyanyi dari lagu yang diputar.

**Queue Song:** untuk mencari penyanyi dari lagu yang ada dalam antrian.

**Playlist Add Album/Song:** untuk menambahkan penyanyi dari album/lagu yang ditambahkan ke playlist.

**Save:** untuk menyimpan daftar penyanyi ke file saat aplikasi ditutup.

Selain itu, struktur data yang kita gunakan untuk merepresentasikan daftar playlist yang mungkin ada dalam aplikasi adalah **List Dinamik** yang merupakan struktur data linier yang dapat berubah ukurannya secara dinamis. Struktur data ini digunakan untuk merepresentasikan daftar playlist dalam aplikasi kita.

**Sketsa Struktur Data:** List Dinamik kita terdiri dari array of pointer ke StrukBerkait.h, kapasitas, dan Neff (jumlah elemen efektif). Array of pointer ini memungkinkan Kita untuk menyimpan banyak StrukBerkait (yang mungkin mewakili playlist) dalam satu list.

**Persoalan yang Diselesaikan:** List Dinamik ini memungkinkan kita untuk menyimpan, mengakses, dan memanipulasi daftar playlist dalam aplikasi kita. Kita dapat dengan mudah menambahkan, menghapus, atau mengubah urutan playlist. Selain itu, Kita juga dapat memuat dan menyimpan data playlist ke dalam file.

**Alasan Pemilihan:** List Dinamik dipilih karena fleksibilitasnya. Berbeda dengan array biasa, List Dinamik dapat berubah ukurannya sesuai kebutuhan, sehingga meminimalkan pemborosan memori. Selain itu, List Dinamik memungkinkan Kita untuk melakukan operasi seperti insert dan delete di tengah list tanpa harus memindahkan banyak elemen, seperti yang mungkin terjadi pada array biasa.

**Implementasi sebagai ADT:** List Dinamik ini diimplementasikan sebagai Abstract Data Type (ADT) karena menyembunyikan detail implementasi dari pengguna. Pengguna hanya perlu tahu tentang operasi yang dapat dilakukan pada List Dinamik (seperti CreateLD, InsertLD, DeleteLD, dll.) tanpa harus tahu bagaimana operasi tersebut diimplementasikan. Ini membuat List Dinamik lebih mudah digunakan dan dipahami, serta memungkinkan Kita untuk mengubah implementasi di belakang layar tanpa mengganggu kode yang menggunakan ADT ini.

**Pengimplementasian ADT ini terhadap fitur yang digunakan:**

**Load:** untuk memuat daftar playlist dari file saat aplikasi dijalankan.

**Queue Playlist:** untuk menambahkan playlist ke antrian.

**Playlist Add Album/Song:** untuk menambahkan album/lagu ke playlist tertentu.

**Playlist Swap:** untuk mengubah urutan playlist dalam list.

**Playlist Remove:** untuk menghapus lagu dari playlist tertentu.

**Playlist Delete:** untuk menghapus playlist dari list.

**Save:** untuk menyimpan daftar playlist ke file saat aplikasi ditutup.

## 2.2 ADT Mesin Karakter dan Mesin Kata

ADT ini digunakan untuk melakukan parsing command dalam aplikasi dan membaca ataupun membuat file konfigurasi dari aplikasi.

**Berikut adalah detil penjelasan untuk adt Mesin Kata yang dalam filenya diberi penamaan Kata.c**

**Sketsa Struktur Data:** Word Kita terdiri dari array karakter dan panjang kata. Array karakter ini memungkinkan Kita untuk menyimpan banyak karakter dalam satu kata.

**Persoalan yang Diselesaikan:** Word ini memungkinkan Kita untuk menyimpan, mengakses, dan memanipulasi kata atau string dalam aplikasi Kita. Kita dapat dengan mudah membuat kata, membandingkan dua kata, menampilkan kata, dan mengubah kata menjadi integer atau sebaliknya.

**Alasan Pemilihan:** Word dipilih karena sederhana dan efisien untuk kasus di mana Kita perlu bekerja dengan kata atau string. Dengan Word, Kita dapat dengan mudah melakukan operasi pada kata atau string.

**Implementasi sebagai ADT:** Word ini diimplementasikan sebagai Abstract Data Type (ADT) karena menyembunyikan detail implementasi dari pengguna. Pengguna hanya perlu tahu tentang operasi yang dapat dilakukan pada Word (seperti CreateWord, IsWordSame, PasteWord, dll.) tanpa harus tahu bagaimana operasi tersebut diimplementasikan. Ini membuat Word lebih mudah digunakan dan dipahami, serta memungkinkan Kita untuk mengubah implementasi di belakang layar tanpa mengganggu kode yang menggunakan ADT ini.

**Berikut Adalah penjelasan** lebih detail tentang bagaimana ADT Word digunakan dalam fitur-fitur aplikasi Kita:

**Start:** Fitur ini digunakan untuk menginisialisasi data aplikasi, termasuk membuat Word untuk menyimpan informasi seperti nama playlist atau lagu. Misalnya, jika pengguna memulai aplikasi dan membuat playlist baru, nama playlist tersebut dapat disimpan dalam Word.

**Save:** Fitur ini digunakan untuk menyimpan data aplikasi ke dalam file. Data yang disimpan mungkin dalam bentuk Word dan kemudian dikonversi menjadi string sebelum

disimpan. Misalnya, jika pengguna menyimpan sesi mereka, semua data yang relevan (seperti nama playlist, lagu, dll.) dapat disimpan dalam Word dan kemudian ditulis ke file.

**Load:** Fitur ini digunakan untuk memuat data dari file ke dalam aplikasi Kita . Data dalam file mungkin disimpan dalam bentuk string dan kemudian dibaca dan disimpan dalam ADT Word. Misalnya, jika pengguna memuat sesi yang sebelumnya disimpan, data dari file dapat dibaca dan disimpan dalam Word.

**List:** Fitur ini digunakan untuk menampilkan daftar playlist atau lagu. Nama-nama ini mungkin disimpan dalam Word dan ditampilkan kepada pengguna. Misalnya, jika pengguna meminta daftar semua playlist, setiap nama playlist (yang disimpan dalam Word) dapat dicetak ke layar.

**Play:** Fitur ini digunakan untuk memutar lagu atau playlist. Nama lagu atau playlist mungkin disimpan dalam Word. Misalnya, jika pengguna memilih untuk memutar lagu, nama lagu tersebut (yang disimpan dalam Word) dapat digunakan untuk mencari dan memutar lagu tersebut.

**QUEUE:** Fitur ini digunakan untuk memanipulasi antrian lagu. Nama lagu mungkin disimpan dalam Word. Misalnya, jika pengguna menambahkan lagu ke antrian, nama lagu tersebut (yang disimpan dalam Word) dapat digunakan untuk mencari dan menambahkan lagu tersebut ke antrian.

**SONG:** Fitur ini digunakan untuk navigasi lagu yang ada pada antrian lagu saat ini. Nama lagu mungkin disimpan dalam Word. Misalnya, jika pengguna memilih untuk melanjutkan ke lagu berikutnya dalam antrian, nama lagu tersebut (yang disimpan dalam Word) dapat digunakan untuk mencari dan memutar lagu tersebut.

**Playlist:** Fitur ini digunakan untuk melakukan operasi pada playlist seperti CREATE, ADD, SWAP, REMOVE dan DELETE. Nama playlist mungkin disimpan dalam Word. Misalnya, jika pengguna membuat playlist baru, nama playlist tersebut (yang disimpan dalam Word) dapat digunakan untuk membuat playlist baru.

**STATUS:** Fitur ini digunakan untuk menampilkan lagu yang sedang dimainkan beserta antrian lagu yang ada dan dari playlist mana lagu itu diputar. Nama lagu dan playlist mungkin disimpan dalam Word. Misalnya, jika pengguna meminta status aplikasi, nama lagu yang sedang dimainkan dan playlist tempat lagu tersebut berasal (keduanya disimpan dalam Word) dapat ditampilkan.

**QUIT:** digunakan untuk keluar dari sesi aplikasi. Tidak ada penggunaan khusus Word untuk fitur ini.

**HELP:** digunakan untuk menampilkan daftar perintah yang mungkin untuk dieksekusi dengan deskripsi mereka. Deskripsi perintah mungkin disimpan dalam Word.

**Selain itu, ada juga adt mesin karakter yang akan dijelaskan lebih lanjut di bawah ini:**

**Sketsa Struktur Data:** Mesin Karakter Kita terdiri dari karakter saat ini (CurrentChar) dan penKita akhir file atau akhir proses (EOP). CurrentChar adalah karakter yang saat ini sedang dibaca atau ditulis, sedangkan EOP adalah penKita yang menunjukkan apakah telah mencapai akhir file atau akhir proses.

**Persoalan yang Diselesaikan:** Mesin Karakter ini memungkinkan Kita untuk membaca dan menulis karakter dari dan ke file atau stdin dalam aplikasi Kita. Kita dapat dengan mudah membaca karakter satu per satu dari file atau stdin, menulis karakter ke file, dan mengecek apakah telah mencapai akhir file atau akhir proses.

**Alasan Pemilihan:** Mesin Karakter dipilih karena memudahkan pembacaan dan penulisan karakter dari dan ke file atau stdin. Dengan Mesin Karakter, Kita dapat membaca dan menulis karakter satu per satu, yang sangat berguna untuk parsing command atau membaca/membuat file konfigurasi.

**Implementasi sebagai ADT:** Mesin Karakter ini diimplementasikan sebagai Abstract Data Type (ADT) karena menyembunyikan detail implementasi dari pengguna. Pengguna hanya perlu tahu tentang operasi yang dapat dilakukan pada Mesin Karakter (seperti StartRead, StartWrite, Adv, AdvWrite, dll.) tanpa harus tahu bagaimana operasi tersebut diimplementasikan. Ini membuat Mesin Karakter lebih mudah digunakan dan dipahami, serta memungkinkan Kita untuk mengubah implementasi di belakang layar tanpa mengganggu kode yang menggunakan ADT ini.

**Pengimplementasian ADT ini terhadap fitur yang digunakan:** Mesin Karakter Kita digunakan secara luas di seluruh aplikasi Kita untuk melakukan parsing command dan membaca/membuat file konfigurasi. Misalnya, saat pengguna memasukkan command, Mesin Karakter dapat digunakan untuk membaca command tersebut satu karakter per satu karakter, mem-parsing command tersebut, dan kemudian melakukan aksi yang sesuai berdasarkan command tersebut. Selain itu, saat aplikasi Kita memuat atau

menyimpan konfigurasi, Mesin Karakter dapat digunakan untuk membaca atau menulis file konfigurasi tersebut satu karakter per satu karakter.

Lebih lanjut berikut adalah penjelasan lebih detail tentang bagaimana ADT Mesin Karakter digunakan dalam fitur-fitur aplikasi Kita:

- **Start:** Fitur ini digunakan untuk menginisialisasi data aplikasi. Mesin Karakter dapat digunakan untuk membaca input dari pengguna dan mem-parsing command yang diberikan pengguna.
- **Save:** Fitur ini digunakan untuk menyimpan data aplikasi ke dalam file. Mesin Karakter dapat digunakan untuk menulis data ke file satu karakter pada satu waktu.
- **Load:** Fitur ini digunakan untuk memuat data dari file ke dalam aplikasi Kita. Mesin Karakter dapat digunakan untuk membaca data dari file satu karakter pada satu waktu.
- **List:** Fitur ini digunakan untuk menampilkan daftar playlist atau lagu. Mesin Karakter dapat digunakan untuk membaca dan menulis nama-nama ini.
- **Play:** Fitur ini digunakan untuk memutar lagu atau playlist. Mesin Karakter dapat digunakan untuk membaca nama lagu atau playlist yang ingin diputar oleh pengguna.
- **QUEUE:** Fitur ini digunakan untuk memanipulasi antrian lagu. Mesin Karakter dapat digunakan untuk membaca dan menulis nama lagu yang ditambahkan atau dihapus dari antrian.
- **SONG:** Fitur ini digunakan untuk navigasi lagu yang ada pada antrian lagu saat ini. Mesin Karakter dapat digunakan untuk membaca dan menulis nama lagu saat pengguna berpindah ke lagu berikutnya atau sebelumnya.
- **Playlist:** Fitur ini digunakan untuk melakukan operasi pada playlist seperti CREATE, ADD, SWAP, REMOVE dan DELETE. Mesin Karakter dapat digunakan untuk membaca dan menulis nama playlist saat pengguna membuat, menambahkan, menukar, menghapus, atau menghapus playlist.

- **STATUS:** Fitur ini digunakan untuk menampilkan lagu yang sedang dimainkan beserta antrian lagu yang ada dan dari playlist mana lagu itu diputar. Mesin Karakter dapat digunakan untuk membaca dan menulis informasi ini.
- **QUIT:** digunakan untuk keluar dari sesi aplikasi. Tidak ada penggunaan khusus Mesin Karakter untuk fitur ini.
- **HELP:** digunakan untuk menampilkan daftar perintah yang mungkin untuk dieksekusi dengan deskripsi mereka. Mesin Karakter dapat digunakan untuk membaca dan menulis informasi ini.

Dengan demikian, ADT Mesin Karakter memainkan peran penting dalam aplikasi Kita dan digunakan dalam berbagai fitur untuk membaca dan menulis karakter dari dan ke file atau stdin. Ini menunjukkan fleksibilitas dan kegunaan ADT Mesin Karakter dalam pengembangan aplikasi Kita.

## 2.3 ADT Queue

ADT ini digunakan untuk merepresentasikan urutan lagu yang akan dimainkan dalam aplikasi.

Struktur data yang Kita gunakan adalah Queue (antrian), yang merupakan struktur data linier yang mengikuti aturan First-In-First-Out (FIFO). Struktur data ini digunakan untuk merepresentasikan urutan lagu yang akan dimainkan dalam aplikasi Kita.

**Sketsa Struktur Data:** Queue Kita terdiri dari array of Detail (yang mewakili lagu), `IdxHead` (indeks elemen pertama dalam antrian), dan `IdxTail` (indeks elemen terakhir dalam antrian).

**Persoalan yang Diselesaikan:** Queue ini memungkinkan Kita untuk menyimpan, mengakses, dan memanipulasi urutan lagu dalam aplikasi Kita. Kita dapat dengan mudah menambahkan lagu ke antrian, memutar lagu pertama dalam antrian, dan menghapus lagu dari antrian.

**Alasan Pemilihan:** Queue dipilih karena sifatnya yang FIFO, yang cocok untuk situasi di mana Kita perlu memutar lagu dalam urutan tertentu. Dengan Queue, Kita dapat menambahkan lagu ke akhir antrian dan selalu memutar lagu dari awal antrian, yang memastikan bahwa semua lagu akan dimainkan dalam urutan yang benar.

**Implementasi sebagai ADT:** Queue ini diimplementasikan sebagai Abstract Data Type (ADT) karena menyembunyikan detail implementasi dari pengguna. Pengguna hanya perlu tahu tentang operasi yang dapat dilakukan pada Queue (seperti CreateQ, EnqueueQ, DequeueQ, dll.) tanpa harus tahu bagaimana operasi tersebut diimplementasikan. Ini membuat Queue lebih mudah digunakan dan dipahami, serta memungkinkan Kita untuk mengubah implementasi di belakang layar tanpa mengganggu kode yang menggunakan ADT ini.

**Pengimplementasian ADT** ini terhadap fitur yang digunakan:

- **Save:** Saat menyimpan data aplikasi, Kita mungkin perlu menyimpan urutan lagu yang akan dimainkan. Kita dapat melakukannya dengan menulis isi Queue ke file. Kita mungkin mulai dari Head Queue dan bergerak ke Tail, menulis setiap Detail (lagu) ke file satu per satu.
- **Load:** Saat memuat data aplikasi, Kita mungkin perlu memuat urutan lagu yang akan dimainkan dari file. Kita dapat melakukannya dengan membaca file dan menambahkan setiap Detail (lagu) ke Queue satu per satu. Kita mungkin mulai dengan Queue kosong dan menggunakan operasi EnqueueQ untuk menambahkan setiap Detail ke Queue.
- **Queue Song:** digunakan untuk menambahkan lagu ke antrian. Lagu yang ditambahkan akan disimpan dalam Queue.
- **Queue Playlist:** digunakan untuk menambahkan semua lagu dalam playlist ke antrian. Lagu-lagu ini akan disimpan dalam Queue.
- **Play Playlist:** digunakan untuk memutar semua lagu dalam playlist. Lagu-lagu ini akan diambil dari Queue dan dimainkan satu per satu.
- **Queue Swap:** digunakan untuk menukar posisi dua lagu dalam antrian. Operasi ini akan memodifikasi Queue.
- **Queue Remove:** digunakan untuk menghapus lagu dari antrian. Operasi ini akan memodifikasi Queue.
- **Queue Clear:** digunakan untuk menghapus semua lagu dari antrian. Operasi ini akan memodifikasi Queue.



- **Song Next:** digunakan untuk memutar lagu berikutnya dalam antrian. Lagu berikutnya akan diambil dari Queue.
- **Song Previous:** digunakan untuk memutar lagu sebelumnya dalam antrian. Meskipun ini tidak biasa untuk Queue, Kita memiliki implementasi khusus Queue yang mendukung operasi ini.

## 2.4 ADT Stack

ADT ini digunakan untuk merepresentasikan urutan lagu yang telah dimainkan dalam aplikasi, dimana TOP merupakan lagu yang telah dimainkan.

Struktur data yang Kita gunakan adalah Stack, yang merupakan struktur data linier yang mengikuti aturan Last-In-First-Out (LIFO). Struktur data ini digunakan untuk merepresentasikan urutan lagu yang telah dimainkan dalam aplikasi Kita.

**Sketsa Struktur Data:** Stack Kita terdiri dari array of Detail (yang mungkin mewakili lagu) dan Top (indeks elemen puncak dalam stack).

**Persoalan yang Diselesaikan:** Stack ini memungkinkan Kita untuk menyimpan, mengakses, dan memanipulasi urutan lagu yang telah dimainkan dalam aplikasi Kita. Kita dapat dengan mudah menambahkan lagu ke stack saat lagu tersebut dimainkan, dan mengambil lagu dari stack saat pengguna memilih untuk memutar lagu sebelumnya.

**Alasan Pemilihan:** Stack dipilih karena sifatnya yang LIFO, yang cocok untuk situasi di mana Kita perlu melacak urutan lagu yang telah dimainkan. Dengan Stack, Kita dapat menambahkan lagu ke puncak stack saat lagu tersebut dimainkan, dan selalu mengambil lagu dari puncak stack, yang memastikan bahwa lagu yang paling baru dimainkan selalu ada di puncak stack.

**Implementasi sebagai ADT:** Stack ini diimplementasikan sebagai Abstract Data Type (ADT) karena menyembunyikan detail implementasi dari pengguna. Pengguna hanya perlu tahu tentang operasi yang dapat dilakukan pada Stack (seperti CreateS, PushS, PopS, dll.) tanpa harus tahu bagaimana operasi tersebut diimplementasikan. Ini membuat Stack lebih mudah digunakan dan dipahami, serta memungkinkan Kita untuk mengubah implementasi di belakang layar tanpa mengganggu kode yang menggunakan ADT ini.

**Pengimplementasian** ADT ini terhadap fitur yang digunakan:

- **Load:** digunakan untuk memuat urutan lagu yang telah dimainkan dari file ke dalam aplikasi Kita. Lagu-lagu ini akan disimpan dalam Stack.
- **Save:** digunakan untuk menyimpan urutan lagu yang telah dimainkan ke dalam file. Lagu-lagu ini akan diambil dari Stack dan ditulis ke file.
- **Play Song:** digunakan untuk memutar lagu. Saat lagu dimainkan, lagu tersebut akan ditambahkan ke Stack.
- **Play Playlist:** digunakan untuk memutar semua lagu dalam playlist. Saat setiap lagu dalam playlist dimainkan, lagu tersebut akan ditambahkan ke Stack.
- **Song Next:** digunakan untuk memutar lagu berikutnya dalam playlist. Saat lagu berikutnya dimainkan, lagu sebelumnya (yang sekarang telah selesai dimainkan) akan ditambahkan ke Stack.
- **Song Previous:** digunakan untuk memutar lagu sebelumnya. Lagu sebelumnya akan diambil dari Stack dan dimainkan.

## 2.5 ADT Set dan Map

ADT Set digunakan untuk menyimpan lagu dari suatu album. Dengan begitu, lagu yang ada dipastikan tidak memiliki duplikat.

Struktur data yang Kita gunakan adalah Set, yang merupakan struktur data yang digunakan untuk menyimpan elemen unik dalam koleksi. Dalam hal ini, Set digunakan untuk menyimpan lagu dari suatu album, dengan setiap lagu dipastikan tidak memiliki duplikat.

**Sketsa Struktur Data:** Set Kita terdiri dari array of Word (yang mewakili lagu) dan Length (jumlah lagu dalam set).

**Persoalan yang Diselesaikan:** Set ini memungkinkan Kita untuk menyimpan, mengakses, dan memanipulasi lagu dalam suatu album dalam aplikasi Kita. Kita dapat dengan mudah menambahkan lagu ke set, menghapus lagu dari set, dan memeriksa apakah suatu lagu ada dalam set.

**Alasan Pemilihan:** Set dipilih karena sifatnya yang memastikan bahwa setiap elemen dalam koleksi unik. Dengan Set, Kita dapat memastikan bahwa tidak ada lagu yang sama dalam satu album.

**Implementasi sebagai ADT:** Set ini diimplementasikan sebagai Abstract Data Type (ADT) karena menyembunyikan detail implementasi dari pengguna. Pengguna hanya perlu tahu tentang operasi yang dapat dilakukan pada Set (seperti CreateSet, AddSet,

DeleteSet, dll.) tanpa harus tahu bagaimana operasi tersebut diimplementasikan. Ini membuat Set lebih mudah digunakan dan dipahami, serta memungkinkan Kita untuk mengubah implementasi di belakang layar tanpa mengganggu kode yang menggunakan ADT ini.

### **Pengimplementasian ADT ini terhadap fitur yang digunakan:**

- **Load:** digunakan untuk memuat lagu dari file ke dalam aplikasi Kita. Lagu-lagu ini akan disimpan dalam Set.
- **Save:** digunakan untuk menyimpan lagu ke dalam file. Lagu-lagu ini akan diambil dari Set dan ditulis ke file.
- **Start:** digunakan untuk menginisialisasi data aplikasi, termasuk membuat Set untuk menyimpan lagu.
- **Play Song:** digunakan untuk memutar lagu. Lagu yang dimainkan mungkin dicari dalam Set.
- **Queue Song:** digunakan untuk menambahkan lagu ke antrian. Lagu yang ditambahkan mungkin dicari dalam Set.
- **Playlist Add Album:** digunakan untuk menambahkan semua lagu dalam album ke playlist. Lagu-lagu ini mungkin dicari dalam Set.
- **Playlist Add Song:** digunakan untuk menambahkan lagu ke playlist. Lagu yang ditambahkan mungkin dicari dalam Set.

### **ADT Map digunakan untuk menyimpan album untuk tiap penyanyi dan juga menyimpan lagu untuk tiap album.**

Struktur data yang Kita gunakan adalah Map, yang merupakan struktur data yang digunakan untuk menyimpan pasangan kunci-nilai. Dalam hal ini, Map digunakan untuk menyimpan album untuk setiap penyanyi dan juga menyimpan lagu untuk setiap album.

**Sketsa Struktur Data:** Map Kita terdiri dari array of MapEntry (yang mewakili pasangan kunci-nilai) dan Length (jumlah pasangan kunci-nilai dalam map). Setiap MapEntry terdiri dari Key (kunci, yang mungkin mewakili penyanyi atau album) dan Value (nilai, yang mungkin mewakili set lagu).

**Persoalan yang Diselesaikan:** Map ini memungkinkan Kita untuk menyimpan, mengakses, dan memanipulasi album dan lagu dalam aplikasi Kita. Kita dapat dengan mudah menambahkan album atau lagu ke map, menghapus album atau lagu dari map, dan memeriksa apakah suatu album atau lagu ada dalam map.

**Alasan Pemilihan:** Map dipilih karena memungkinkan Kita untuk menyimpan pasangan kunci-nilai, yang sangat berguna untuk situasi di mana Kita perlu mengasosiasikan nilai (seperti set lagu) dengan kunci tertentu (seperti penyanyi atau album).

**Implementasi sebagai ADT:** Map ini diimplementasikan sebagai Abstract Data Type (ADT) karena menyembunyikan detail implementasi dari pengguna. Pengguna hanya perlu tahu tentang operasi yang dapat dilakukan pada Map (seperti CreateM, InsertM, DeleteM, dll.) tanpa harus tahu bagaimana operasi tersebut diimplementasikan. Ini membuat Map lebih mudah digunakan dan dipahami, serta memungkinkan Kita untuk mengubah implementasi di belakang layar tanpa mengganggu kode yang menggunakan ADT ini.

Pengimplementasian ADT ini terhadap fitur yang digunakan:

- **Load:** digunakan untuk memuat album dan lagu dari file ke dalam aplikasi Kita. Album dan lagu ini akan disimpan dalam Map.
- **Save:** digunakan untuk menyimpan album dan lagu ke dalam file. Album dan lagu ini akan diambil dari Map dan ditulis ke file.
- **Start:** digunakan untuk menginisialisasi data aplikasi, termasuk membuat Map untuk menyimpan album dan lagu.
- **Play Song:** digunakan untuk memutar lagu. Lagu yang dimainkan mungkin dicari dalam Map berdasarkan album atau penyanyi.
- **Queue Song:** digunakan untuk menambahkan lagu ke antrian. Lagu yang ditambahkan mungkin dicari dalam Map berdasarkan album atau penyanyi.
- **Playlist Add Album:** digunakan untuk menambahkan lagu ke playlist. Lagu yang ditambahkan mungkin dicari dalam Map berdasarkan album atau penyanyi.
- **Playlist Add Song :** digunakan untuk menambahkan semua lagu dalam album ke playlist. Lagu-lagu ini mungkin dicari dalam Map berdasarkan album.

## 2.6 ADT List dengan Struktur Berkait

ADT ini digunakan untuk merepresentasikan sebuah playlist dimana playlist merupakan suatu kumpulan lagu yang kontigu.

Struktur data yang Kita gunakan adalah Struktur Berkait (Linked List), yang merupakan struktur data yang digunakan untuk menyimpan sekumpulan elemen (dalam hal ini, lagu) dalam urutan tertentu.

**Sketsa Struktur Data:** Struktur Berkait Kita terdiri dari node-node yang masing-masing berisi Detail (yang mungkin mewakili lagu) dan pointer ke node berikutnya. Setiap Struktur Berkait juga memiliki judul dan pointer ke node pertama.

**Persoalan yang Diselesaikan:** Struktur Berkait ini memungkinkan Kita untuk menyimpan, mengakses, dan memanipulasi lagu dalam suatu playlist dalam aplikasi Kita. Kita dapat dengan mudah menambahkan lagu ke Struktur Berkait, menghapus lagu dari Struktur Berkait, dan memeriksa apakah suatu lagu ada dalam Struktur Berkait.

**Alasan Pemilihan:** Struktur Berkait dipilih karena fleksibilitasnya. Dengan Struktur Berkait, Kita dapat dengan mudah menambahkan atau menghapus elemen di mana saja dalam urutan, yang mungkin sulit dilakukan dengan struktur data lain seperti array.

**Implementasi sebagai ADT:** Struktur Berkait ini diimplementasikan sebagai Abstract Data Type (ADT) karena menyembunyikan detail implementasi dari pengguna. Pengguna hanya perlu tahu tentang operasi yang dapat dilakukan pada Struktur Berkait (seperti CreateSB, InsertSB, DeleteSB, dll.) tanpa harus tahu bagaimana operasi tersebut diimplementasikan. Ini membuat Struktur Berkait lebih mudah digunakan dan dipahami, serta memungkinkan Kita untuk mengubah implementasi di belakang layar tanpa mengganggu kode yang menggunakan ADT ini.

**Pengimplementasian ADT ini terhadap fitur yang digunakan:**

- **Load:** digunakan untuk memuat lagu dari file ke dalam aplikasi Kita. Lagu-lagu ini akan disimpan dalam Struktur Berkait.
- **Save:** digunakan untuk menyimpan lagu ke dalam file. Lagu-lagu ini akan diambil dari Struktur Berkait dan ditulis ke file.
- **Play Playlist:** digunakan untuk memutar semua lagu dalam playlist. Lagu-lagu ini akan diambil dari Struktur Berkait dan dimainkan satu per satu.
- **Queue Playlist:** digunakan untuk menambahkan semua lagu dalam playlist ke antrian. Lagu-lagu ini akan diambil dari Struktur Berkait dan ditambahkan ke antrian.
- **Playlist Create:** digunakan untuk membuat playlist baru. Playlist baru ini akan disimpan sebagai Struktur Berkait baru.
- **Playlist Add Song:** digunakan untuk menambahkan lagu ke playlist. Lagu yang ditambahkan akan ditambahkan ke Struktur Berkait yang mewakili playlist.

- **Playlist Add Album:** digunakan untuk menambahkan semua lagu dalam album ke playlist. Lagu-lagu ini akan diambil dari Struktur Berkait yang mewakili album dan ditambahkan ke Struktur Berkait yang mewakili playlist.
- **Playlist Swap:** digunakan untuk menukar posisi dua lagu dalam playlist. Operasi ini akan memodifikasi Struktur Berkait yang mewakili playlist.
- **Playlist Remove:** digunakan untuk menghapus lagu dari playlist. Operasi ini akan memodifikasi Struktur Berkait yang mewakili playlist.
- **Playlist Delete:** digunakan untuk menghapus playlist. Operasi ini akan menghapus Struktur Berkait yang mewakili playlist.

## 2.7 ADT Mesin File dan Mesin Input

Struktur data yang Kita gunakan adalah Mesin File, yang merupakan struktur data yang digunakan untuk membaca dan menulis file teks.

**Sketsa Struktur Data:** Mesin File kita tidak memiliki struktur data khusus. Sebaliknya, ia menggunakan variabel global seperti `currentChar` dan `currentWord` yang didefinisikan di tempat lain dalam program kita.

**Persoalan yang Diselesaikan:** Mesin File ini memungkinkan Anda untuk membaca dan menulis file teks dalam aplikasi kita. Kita dapat dengan mudah membaca kata-kata dari file, mengabaikan spasi dan baris baru, dan menulis kata-kata ke file.

**Alasan Pemilihan:** Mesin File dipilih karena kemampuannya untuk membaca dan menulis file teks, yang sangat berguna untuk aplikasi yang perlu menyimpan dan memuat data.

**Implementasi sebagai ADT:** Mesin File ini diimplementasikan sebagai Abstract Data Type (ADT) karena menyembunyikan detail implementasi dari pengguna. Pengguna hanya perlu tahu tentang operasi yang dapat dilakukan pada Mesin File (seperti `StartFileF`, `AdvWordF`, `AdvLineF`, dll.) tanpa harus tahu bagaimana operasi tersebut diimplementasikan. Ini membuat Mesin File lebih mudah digunakan dan dipahami, serta memungkinkan kita untuk mengubah implementasi di belakang layar tanpa mengganggu kode yang menggunakan ADT ini.

**Pengimplementasian ADT ini terhadap fitur yang digunakan:**

STEI- ITB	<nomor dokumen>	Halaman 22 dari 99 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

**Start:** digunakan untuk memulai aplikasi kita. Mesin File mungkin digunakan di sini untuk membuka file konfigurasi atau file data awal.

**Load:** digunakan untuk memuat data dari file ke dalam aplikasi kita. Mesin File akan digunakan di sini untuk membaca data dari file.

**Save:** digunakan untuk menyimpan data aplikasi kita ke file. Mesin File akan digunakan di sini untuk menulis data ke file.

Selain itu, kita menggunakan struktur data Mesin Input yang digunakan untuk menerima inputan

**Sketsa Struktur Data:** Mesin Input kita tidak memiliki struktur data khusus. Sebaliknya, ia menggunakan variabel global seperti `currentChar` dan `currentWord` yang didefinisikan di tempat lain dalam program Kita.

**Persoalan yang Diselesaikan:** Mesin Input ini memungkinkan kita untuk membaca dan memproses input dari pengguna dalam aplikasi kita. Kita dapat dengan mudah membaca kata-kata dari input pengguna, mengabaikan spasi dan baris baru, dan memproses perintah yang dimasukkan oleh pengguna.

**Alasan Pemilihan:** Mesin Input dipilih karena kemampuannya untuk membaca dan memproses input teks, yang sangat berguna untuk aplikasi yang perlu menerima dan memproses perintah dari pengguna.

**Implementasi sebagai ADT:** Mesin Input ini diimplementasikan sebagai Abstract Data Type (ADT) karena menyembunyikan detail implementasi dari pengguna. Pengguna hanya perlu tahu tentang operasi yang dapat dilakukan pada Mesin Input (seperti `StartWordI`, `AdvWordI`, `CopyWordI`, dll.) tanpa harus tahu bagaimana operasi tersebut diimplementasikan. Ini membuat Mesin Input lebih mudah digunakan dan dipahami, serta memungkinkan kita untuk mengubah implementasi di belakang layar tanpa mengganggu kode yang menggunakan ADT ini.

**Pengimplementasian ADT ini terhadap fitur yang digunakan:**

**Load:** digunakan untuk memuat data dari file ke dalam aplikasi kita. Mesin Input akan digunakan di sini untuk membaca data dari input pengguna.

**Save:** digunakan untuk menyimpan data aplikasi kita ke file. Mesin Input akan digunakan di sini untuk membaca nama file dari input pengguna.

**Queue Remove, Queue Clear, Queue Swap, Playlist Create, Playlist Add Album, Playlist Add Song, Playlist Swap, Playlist Remove, Playlist Delete, dan Quit:**

Semua fitur ini memerlukan input dari pengguna, dan Mesin Input akan digunakan untuk membaca dan memproses input tersebut.



### 3 Program Utama

```
#include "data.h"
#include "console.h"

int main() {
    Initiate();
    PRINTWELCOME();
    printf("\n");
    HELP();
    printf("\n");

    while (!EndProgram) {

        printf(">> ");
        StartWordI();
        printf("\n");

        if (IsWordSame(CurrentWord, WordSTART)) {
            if (IsEOP()) {
                if (!LoggedIn) {
                    START();
                }
            }
            else {
                CorrectSession = false;
            }
        }
        else {
            KnownCommand = false;
        }
    }

    else if (IsWordSame(CurrentWord, WordLOAD)) {
        AdvLinI();
        if (IsEOP()) {
            if (!LoggedIn) {
                LOAD();
            }
        }
    }
}
```

```

    }
    else {
        CorrectSession = false;
    }
}
else {
    KnownCommand = false;
}
}

else if (IsWordSame(CurrentWord, WordLIST)) {
    AdvWordl();
    if (IsEOP()) {
        if (IsWordSame(CurrentWord, WordDEFAULT)) {
            if (LoggedIn) {
                LISTDEFAULT();
            }
            else {
                CorrectSession = false;
            }
        }
        else if (IsWordSame(CurrentWord, WordPLAYLIST)) {
            if (LoggedIn) {
                LISTPLAYLIST();
            }
            else {
                CorrectSession = false;
            }
        }
        else {
            KnownCommand = false;
        }
    }
    else {
        KnownCommand = false;
    }
}
}

```

```

else if (IsWordSame(CurrentWord, WordPLAY)) {
    AdvWordI();
    if (IsEOP()) {
        if (IsWordSame(CurrentWord, WordSONG)) {
            if (LoggedIn) {
                PLAYSONG();
            }
            else {
                CorrectSession = false;
            }
        }
        else if (IsWordSame(CurrentWord, WordPLAYLIST)) {
            if (LoggedIn) {
                PLAYPLAYLIST();
            }
            else {
                CorrectSession = false;
            }
        }
        else {
            KnownCommand = false;
        }
    }
    else {
        KnownCommand = false;
    }
}

else if (IsWordSame(CurrentWord, WordQUEUE)) {
    AdvWordI();
    if (IsWordSame(CurrentWord, WordSWAP)) {
        if (LoggedIn) {
            QUEUESWAP();
        }
        else {
            CorrectSession = false;
        }
    }
}

```

```

else if (IsWordSame(CurrentWord, WordREMOVE)) {
    if (LoggedIn) {
        QUEUEREMOVE();
    }
    else {
        CorrectSession = false;
    }
}
else {
    if (IsEOP()) {
        if (IsWordSame(CurrentWord, WordSONG)) {
            if (LoggedIn) {
                QUEUESONG();
            }
            else {
                CorrectSession = false;
            }
        }
        else if (IsWordSame(CurrentWord, WordPLAYLIST)) {
            if (LoggedIn) {
                QUEUEPLAYLIST();
            }
            else {
                CorrectSession = false;
            }
        }
        else if (IsWordSame(CurrentWord, WordCLEAR)) {
            if (LoggedIn) {
                QUEUECLEAR();
            }
            else {
                CorrectSession = false;
            }
        }
        else {
            KnownCommand = false;
        }
    }
}

```

```

        else {
            KnownCommand = false;
        }
    }
}

else if (IsWordSame(CurrentWord, WordSONG)) {
    AdvWordI();
    if (IsEOP()) {
        if (IsWordSame(CurrentWord, WordNEXT)) {
            if (LoggedIn) {
                SONGNEXT();
            }
            else {
                CorrectSession = false;
            }
        }
        else if (IsWordSame(CurrentWord, WordPREVIOUS)) {
            if (LoggedIn) {
                SONGPREVIOUS();
            }
            else {
                CorrectSession = false;
            }
        }
        else {
            KnownCommand = false;
        }
    }
    else {
        KnownCommand = false;
    }
}

else if (IsWordSame(CurrentWord, WordPLAYLIST)) {
    AdvWordI();
    if (IsWordSame(CurrentWord, WordADD)) {
        AdvWordI();
    }
}

```

```

if (IsEOP()) {
    if (IsWordSame(CurrentWord, WordSONG)) {
        if (LoggedIn) {
            PLAYLISTADDSONG();
        }
        else {
            CorrectSession = false;
        }
    }
    else if (IsWordSame(CurrentWord, WordALBUM)) {
        if (LoggedIn) {
            PLAYLISTADDALBUM();
        }
        else {
            CorrectSession = false;
        }
    }
    else {
        KnownCommand = false;
    }
}
else {
    KnownCommand = false;
}
}
else {
    if (IsWordSame(CurrentWord, WordSWAP)) {
        if (LoggedIn) {
            PLAYLISTSWAP();
        }
        else {
            CorrectSession = false;
        }
    }
    else if (IsWordSame(CurrentWord, WordREMOVE)) {
        if (LoggedIn) {
            PLAYLISTREMOVE();
        }
    }
}

```

```

        else {
            CorrectSession = false;
        }
    }
    else {
        if (IsEOP()) {
            if (IsWordSame(CurrentWord, WordCREATE)) {
                if (LoggedIn) {
                    PLAYLISTCREATE();
                }
                else {
                    CorrectSession = false;
                }
            }
            else if (IsWordSame(CurrentWord, WordDELETE)) {
                if (LoggedIn) {
                    PLAYLISTDELETE();
                }
                else {
                    CorrectSession = false;
                }
            }
            else {
                KnownCommand = false;
            }
        }
        else {
            KnownCommand = false;
        }
    }
}

else if (IsWordSame(CurrentWord, WordSTATUS)) {
    if (IsEOP()) {
        if (LoggedIn) {
            STATUS();
        }
    }
}

```

```

        else {
            CorrectSession = false;
        }
    }
    else {
        KnownCommand = false;
    }
}

else if (IsWordSame(CurrentWord, WordSAVE)) {
    if (!IsEOP()) {
        AdvLine1();
        if (LoggedIn) {
            SAVE();
        }
        else {
            CorrectSession = false;
        }
    }
    else {
        KnownCommand = false;
    }
}

else if (IsWordSame(CurrentWord, WordQUIT)) {
    if (IsEOP()) {
        if (LoggedIn) {
            QUIT();
        }
        else {
            CorrectSession = false;
        }
    }
    else {
        KnownCommand = false;
    }
}
}

```



```

else if (IsWordSame(CurrentWord, WordHELP)) {
    if (IsEOP()) {
        HELP();
    }
    else {
        KnownCommand = false;
    }
}
// UNTUK TESTING
else if (IsWordSame(CurrentWord, WordDISPLAY)) {
    DISPLAY();
}
// SAMPE SINI
else if (IsWordSame(CurrentWord, WordENHANCE)) {
    ENHANCE();
}

else {
    KnownCommand = false;
}

if (!KnownCommand || !CorrectSession) {
    INVALIDCOMMAND();
}

if (!IsEOF()) {
    AdvLinel();
}

printf("\n");
}

PRINTKELUAR();

return 0;
}

```

1. Program dimulai dengan memanggil fungsi Initiate() dan PRINTWELCOME(). Fungsi Initiate() mungkin digunakan untuk menginisialisasi beberapa variabel atau struktur data yang diperlukan oleh program, sementara PRINTWELCOME() mungkin digunakan untuk mencetak pesan selamat datang ke pengguna.
2. Program kemudian memasuki loop utama, yang berjalan selama variabel EndProgram bernilai false. Dalam setiap iterasi loop, program membaca kata dari input pengguna dan memeriksa kata tersebut untuk menentukan perintah apa yang harus dijalankan.
3. Jika kata yang dibaca sama dengan "START", "LOAD", "LIST", atau "PLAY", program akan memanggil fungsi yang sesuai (START(), LOAD(), LISTDEFAULT(), PLAYSONG(), dll.) untuk menjalankan perintah tersebut. Jika kata yang dibaca tidak cocok dengan salah satu perintah ini, program akan menandai perintah tersebut sebagai tidak dikenal dengan mengatur KnownCommand menjadi false.
4. Setelah menjalankan perintah, program kembali ke awal loop dan membaca kata berikutnya dari input pengguna. Proses ini berlanjut sampai EndProgram menjadi true, yang mungkin terjadi jika pengguna memasukkan perintah khusus untuk keluar dari program.
5. Beberapa perintah memerlukan argumen tambahan, yang dibaca dengan memanggil AdvWord() atau AdvLine(). Misalnya, perintah "PLAY" memerlukan argumen "SONG" atau "PLAYLIST" untuk menentukan apa yang harus diputar.
6. Beberapa fungsi, seperti START() dan LOAD(), hanya dapat dipanggil jika pengguna belum masuk (LoggedIn adalah false). Jika pengguna sudah masuk, program akan menandai sesi sebagai tidak benar dengan mengatur CorrectSession menjadi false.
7. Beberapa fungsi lainnya, seperti LISTDEFAULT() dan PLAYSONG(), hanya dapat dipanggil jika pengguna sudah masuk (LoggedIn adalah true). Jika pengguna belum masuk, program juga akan menandai sesi sebagai tidak benar.
8. Program memeriksa kata yang dibaca dari input pengguna dan membandingkannya dengan string "QUEUE" dan "SONG".
9. Jika kata yang dibaca sama dengan "QUEUE", program akan membaca kata berikutnya dan memeriksa apakah itu sama dengan "SWAP", "REMOVE", "SONG", "PLAYLIST", atau "CLEAR". Jika ya, program akan memanggil fungsi yang sesuai (QUEUESWAP(), QUEUEREMOVE(), QUEUESONG(), QUEUEPLAYLIST(), atau QUEUECLEAR()) untuk menjalankan perintah tersebut. Jika kata yang dibaca tidak cocok dengan salah satu perintah ini, program akan menandai perintah tersebut sebagai tidak dikenal dengan mengatur KnownCommand menjadi false.

10. Jika kata yang dibaca sama dengan “SONG”, program akan membaca kata berikutnya dan memeriksa apakah itu sama dengan “NEXT” atau “PREVIOUS”. Jika ya, program akan memanggil fungsi yang sesuai (SONGNEXT() atau SONGPREVIOUS()) untuk menjalankan perintah tersebut. Jika kata yang dibaca tidak cocok dengan salah satu perintah ini, program akan menandai perintah tersebut sebagai tidak dikenal dengan mengatur KnownCommand menjadi false.
11. Beberapa fungsi, seperti QUEUESWAP(), QUEUEREMOVE(), QUEUESONG(), QUEUEPLAYLIST(), QUEUECLEAR(), SONGNEXT(), dan SONGPREVIOUS(), hanya dapat dipanggil jika pengguna sudah masuk (LoggedIn adalah true). Jika pengguna belum masuk, program akan menandai sesi sebagai tidak benar dengan mengatur CorrectSession menjadi false.
12. Program memeriksa kata yang dibaca dari input pengguna dan membandingkannya dengan string “PLAYLIST”, “STATUS”, “SAVE”, dan “QUIT”.
13. Jika kata yang dibaca sama dengan “PLAYLIST”, program akan membaca kata berikutnya dan memeriksa apakah itu sama dengan “ADD”, “SWAP”, “REMOVE”, “CREATE”, atau “DELETE”. Jika ya, program akan memanggil fungsi yang sesuai (PLAYLISTADDSONG(), PLAYLISTADDALBUM(), PLAYLISTSWAP(), PLAYLISTREMOVE(), PLAYLISTCREATE(), atau PLAYLISTDELETE()) untuk menjalankan perintah tersebut. Jika kata yang dibaca tidak cocok dengan salah satu perintah ini, program akan menandai perintah tersebut sebagai tidak dikenal dengan mengatur KnownCommand menjadi false.
14. Jika kata yang dibaca sama dengan “STATUS”, “SAVE”, atau “QUIT”, program akan memanggil fungsi yang sesuai (STATUS(), SAVE(), atau QUIT()) jika pengguna sudah masuk (LoggedIn adalah true). Jika pengguna belum masuk, program akan menandai sesi sebagai tidak benar dengan mengatur CorrectSession menjadi false.
15. Program memeriksa kata yang dibaca dari input pengguna dan membandingkannya dengan string “HELP” dan “DISPLAY”.
16. Jika kata yang dibaca sama dengan “HELP”, program akan memanggil fungsi HELP() jika tidak ada kata lain setelah “HELP”. Jika ada kata lain setelah “HELP”, program akan menandai perintah tersebut sebagai tidak dikenal dengan mengatur KnownCommand menjadi false.
17. Jika kata yang dibaca sama dengan “DISPLAY”, program akan memanggil fungsi DISPLAY(). Fungsi ini tampaknya digunakan untuk tujuan pengujian dan mungkin mencetak beberapa informasi untuk debugging atau pengujian.
18. Jika kata yang dibaca tidak cocok dengan salah satu perintah yang dikenal, program akan menandai perintah tersebut sebagai tidak dikenal dengan mengatur KnownCommand menjadi false.

19. Setelah memeriksa perintah, program memeriksa apakah perintah tersebut dikenal (KnownCommand adalah true) dan apakah sesi pengguna benar (CorrectSession adalah true). Jika salah satu kondisi ini tidak terpenuhi, program akan memanggil fungsi INVALIDCOMMAND() untuk menangani perintah yang tidak valid atau sesi yang salah.
20. Program kemudian memeriksa apakah telah mencapai akhir file input dengan memanggil IsEOF(). Jika belum, program akan membaca baris berikutnya dari input dengan memanggil AdvLine().
21. Proses ini berlanjut sampai program mencapai akhir file input. Setelah itu, program mencetak pesan keluar dengan memanggil PRINTKELUAR() dan mengakhiri eksekusi dengan mengembalikan 0.

## 4 Algoritma-Algoritma Menarik

### 4.1 ADT List Dinamik x Struktur Berkait

```
#define Nil NULL

typedef struct node* Address;

typedef struct node {
    Detail Info;
    Address Next;
} Node;

typedef struct {
    Word Title;
    Address First;
} StrukBerkait;

#define Next(q) (q)->Next
#define Info(q) (q)->Info
#define First(sb) ((sb).First)
#define Title(sb) ((sb).Title)
#define InitialCapacity 16

typedef int IdxType;

typedef struct {
    StrukBerkait *Content;
    int Capacity;
    int Neff;
} ListDinamik;

ListDinamik CreateLD() {
    ListDinamik A;
    A.Content = (StrukBerkait *)malloc(sizeof(StrukBerkait) * InitialCapacity);
    A.Capacity = InitialCapacity;
    A.Neff = 0;
```

```

        return A;
    }

void DeallocateLD(ListDinamik *l) {
    free((*l).Content);
}

boolean IsEmptyLD(ListDinamik l) {
    return l.Neff == 0;
}

boolean IsFullLD(ListDinamik l) {
    return l.Neff == l.Capacity;
}

boolean IsIdxValidLD(ListDinamik l, IdxType i) {
    return i < l.Neff && i >= 0;
}

EType GetLD(ListDinamik l, IdxType i) {
    return Title(l.Content[i]);
}

void InsertLD(ListDinamik *l, EType x, IdxType i) {
    StrukBerkait xsb;

    CreateSB(&xsb);
    Title(xsb) = x;

    int j;

    if (IsFullLD(*l)) {
        StrukBerkait *temp = (StrukBerkait *)malloc(sizeof(StrukBerkait) *
(*l).Capacity);
        for (j = 0; j < (*l).Neff; j++)
        {
            PasteWord(Title((*l).Content[j]), &Title(temp[j]));
            First(temp[j]) = First((*l).Content[j]);
        }
    }
}

```

```

    }

    DeallocateLD(l);
    (*l).Capacity *= 2;
    (*l).Content = (StrukBerkait *)malloc(sizeof(StrukBerkait) * (*l).Capacity);

    for (j = 0; j < (*l).Neff; j++)
    {
        PasteWord(Title(temp[j]), &Title((*l).Content[j]));
        First((*l).Content[j]) = First(temp[j]);
    }
    free(temp);
}

for (j = (*l).Neff; j > i; j--)
{
    PasteWord(Title((*l).Content[j - 1]), &Title((*l).Content[j]));
    First((*l).Content[j]) = First((*l).Content[j - 1]);
}

PasteWord(Title(xsb), &Title((*l).Content[i]));
(*l).Neff++;
}

void DeleteLD(ListDinamik *l, IdxType i) {
    int j;
    for (j = i; j < (*l).Neff; j++)
    {
        PasteWord(Title((*l).Content[j + 1]), &Title((*l).Content[j]));
        First((*l).Content[j]) = First((*l).Content[j + 1]);
    }
    (*l).Neff--;

    if ((*l).Neff <= ((*l).Capacity / 4)) {
        StrukBerkait *temp = (StrukBerkait *)malloc(sizeof(StrukBerkait) *
(*l).Capacity);
        for (j = 0; j < (*l).Neff; j++)
        {

```

```

        PasteWord(Title((*l).Content[j]), &Title(temp[j]));
        First(temp[j]) = First((*l).Content[j]);
    }

    DeallocateLD(l);
    (*l).Capacity /= 2;
    (*l).Content = (StrukBerkait *)malloc(sizeof(StrukBerkait) * (*l).Capacity);

    for (j = 0; j < (*l).Neff; j++)
    {
        PasteWord(Title(temp[j]), &Title((*l).Content[j]));
        First((*l).Content[j]) = First(temp[j]);
    }
    free(temp);
}
}

void DisplayLD(ListDinamik l) {
    if (IsEmptyLD(l)) {
        printf("Kosong\n");
    }
    else {
        for (int i = 0; i < l.Neff; i++) {
            printf("  %d. ", i+1);
            DisplayWordNewLine(Title(l.Content[i]));
        }
    }
}

void DisplayIsiLD(ListDinamik l, ElType x) {
    boolean found = false;
    int i = 0;
    while (i < l.Neff && !found) {
        if (IsWordSame(Title(l.Content[i]), x)) {
            DisplaySB(l.Content[i]);
            found = true;
        }
        i++;
    }
}

```



```

    }
    if (!found) {
        printf("Playlist tidak ada\n");
    }
}

void DisplaySemuaLD(ListDinamik l) {
    if (IsEmptyLD(l)) {
        printf("Kosong\n");
    }

    for (int i = 0; i < l.Neff; i++) {
        printf("Judul %d: ", i+1);
        DisplayWordNewLine(Title(l.Content[i]));
        DisplaySB(l.Content[i]);
    }
}

```

Primitif list dinamik digunakan dalam semua fungsi yang berhubungan dengan pengaksesan, penambahan, pengurangan, dan penggantian playlist. Ini menarik karena cara penggabungan list dinamik dan struktur berkait. Struktur berkait memiliki struktur Title dan First. Title merupakan judul dari suatu playlist dan First adalah pointer terhadap node pertama struktur berkait playlist tertentu.

Dalam list dinamik, hanya diakses Title dan penukaran first. Ketika ingin mengakses isi dari playlist, maka digunakan primitif struktur berkait.

## 4.2 Enhance

```

printf("Playlist Anda:\n");
DisplayLD(DaftarPlaylist);
printf("\n");
printf("Pilih ID playlist yang ingin di-enhance: ");
StartLine();
printf("\n");
int id_playlist = WordToInt(CurrentWord) - 1;

if (IsValidLD(DaftarPlaylist, id_playlist)) {

```

```

int n_list_lagu = 0;
int max_iteration = 0;
int id_penyanyi;
int id_album;
int id_lagu;
Word penyanyi;
Word album;
Word inputYN;
StrukBerkait list_lagu;
Set list_album;
Detail temp;

while (max_iteration < 10 && n_list_lagu < 2) {

    id_penyanyi = gacha(DaftarPenyanyi.Neff);
    PasteWord(DaftarPenyanyi.Content[id_penyanyi], &penyanyi);
    list_album = ValueM(AlbumPenyanyi, penyanyi);

    id_album = gacha(list_album.Length);
    PasteWord(list_album.Content[id_album], &album);

    CreateSB(&list_lagu);

    for (int i = 0; i < ValueM(LaguAlbum, album).Length; i++) {
        Created(&temp, penyanyi, album, ValueM(LaguAlbum, album).Content[i]);
        if (!IsMemberSB(DaftarPlaylist.Content[id_playlist], temp)) {
            InsertSB(&list_lagu, temp, LengthSB(list_lagu));
        }
    }

    n_list_lagu = LengthSB(list_lagu);
    max_iteration++;
}

if (n_list_lagu >= 2) {
    PasteWord(WordY, &inputYN);

    while (!IsWordSame(inputYN, WordN)) {
        printf("Berikut adalah lagu-lagu rekomendasi dari penyanyi ");
        DisplayWord(penyanyi);
        printf(" dalam album ");
        DisplayWord(album);
        printf(" :\n");
    }
}

```

```

        DisplaySB(list_lagu);
        printf("\n");

        printf("Apakah Anda ingin memasuki lagu rekomendasi dalam playlist?
(Y/N): ");
        StartLine1();
        PasteWord(CurrentWord, &inputYN);

        if (IsWordSame(inputYN, WordY)) {
            printf("ID lagu berapa yang ingin ditambahkan?: ");
            StartLine1();
            id_lagu = WordToInt(CurrentWord) - 1;
            printf("\n");

            if (IsValidSB(list_lagu, id_lagu)) {
                GetSB(&list_lagu, &temp, id_lagu);
                if (!IsMemberSB(DaftarPlaylist.Content[id_playlist], temp)) {
                    printf("Lagu ");
                    DisplayWord(temp.Lagu);
                    printf(" sudah dimasukkan ke dalam playlist.\n\n");
                    InsertSB(&DaftarPlaylist.Content[id_playlist], temp,
LengthSB(DaftarPlaylist.Content[id_playlist]));
                }
                else {
                    printf("Lagu ");
                    DisplayWord(temp.Lagu);
                    printf(" sudah pernah dimasukkan ke dalam playlist.\n\n");
                }
            }
            else {
                printf("ID Lagu %d tidak ada dalam rekomendasi. Silakan coba
lagi.\n\n", id_lagu + 1);
            }
        }
        else if (!IsWordSame(inputYN, WordN)) {
            printf("Input hanya bisa Y/N!\n\n");
        }
    }
}
else {
    printf("ID Playlist %d memiliki terlalu banyak lagu untuk direkomendasi.\n",
id_playlist + 1);
}
}

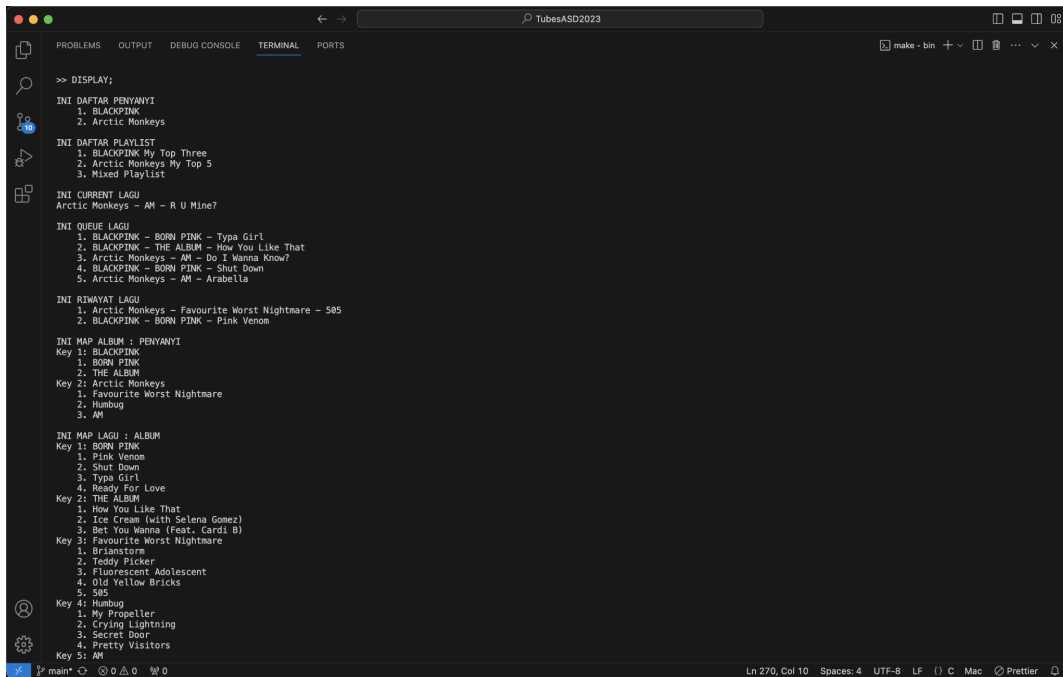
```

```
}  
else {  
    printf("ID Playlist %d tidak ada dalam daftar. Silakan coba lagi.\n", id_playlist +  
1);  
}
```

Enhance adalah fungsi yang digunakan ketika user ingin menambahkan lagu kepada suatu playlist berdasarkan rekomendasi random. Algoritma ini menggunakan fungsi gacha yang berupa random berdasarkan waktu jalannya program. Dengan menggunakan looping pengecekan beberapa lapis, Enhance dapat memastikan bahwa rekomendasi memungkinkan dan menghasilkan rekomendasi lagu baru dengan jumlah minimal 2.

## 5 Data Test

Berikut adalah patokan data sebelum dilakukan data test, jadi misal dilakukan data test kembali, maka patokannya tetap yang akan ditampilkan di bawah.



```
>>> DISPLAY;

INI DAFTAR PENYANYI
1. BLACKPINK
2. Arctic Monkeys

INI DAFTAR PLAYLIST
1. BLACKPINK My Top Three
2. Arctic Monkeys My Top 5
3. Mixed Playlist

INI CURRENT LAGU
Arctic Monkeys - AM - R U Mine?

INI QUEUE LAGU
1. BLACKPINK - BORN PINK - Typa Girl
2. BLACKPINK - THE ALBUM - How You Like That
3. Arctic Monkeys - AM - Do I Wanna Know?
4. BLACKPINK - BORN PINK - Shut Down
5. Arctic Monkeys - AM - Arabella

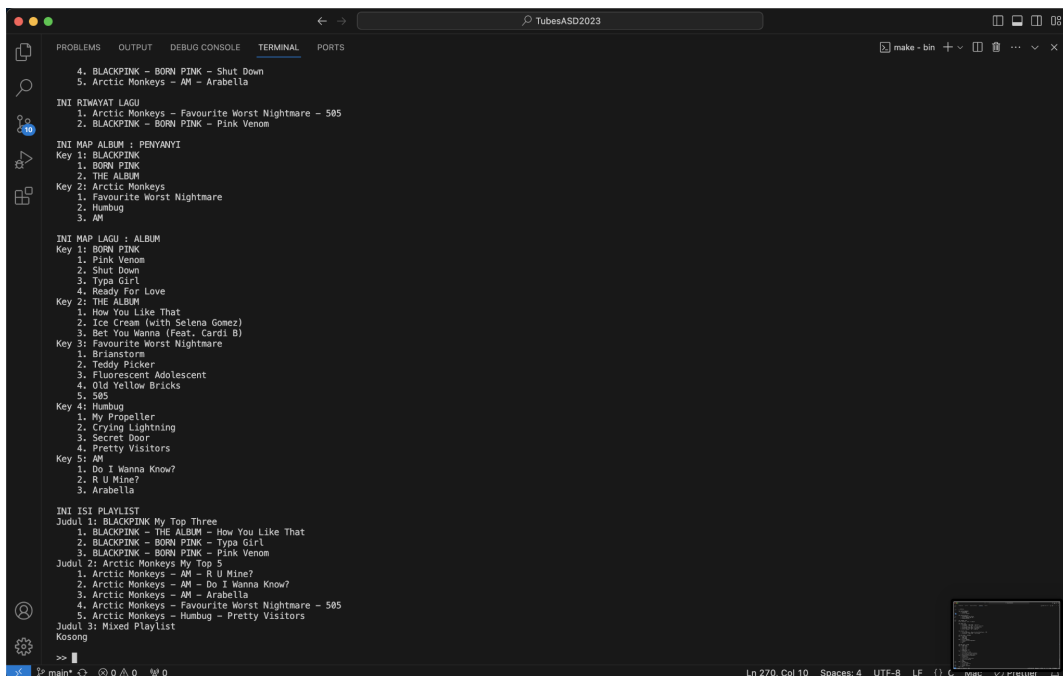
INI RIMAYAT LAGU
1. Arctic Monkeys - Favourite Worst Nightmare - 505
2. BLACKPINK - BORN PINK - Pink Venom

INI MAP ALBUM : PENYANYI
Key 1: BLACKPINK
1. BORN PINK
2. THE ALBUM
Key 2: Arctic Monkeys
1. Favourite Worst Nightmare
2. Humbug
3. AM

INI MAP LAGU : ALBUM
Key 1: BORN PINK
1. Pink Venom
2. Shut Down
3. Typa Girl
4. Ready For Love
Key 2: THE ALBUM
1. How You Like That
2. Ice Cream (with Selena Gomez)
3. Bet You Wanna (Feat. Cardi B)
Key 3: Favourite Worst Nightmare
1. Brianstorm
2. Teddy Picker
3. Fluorescent Adolescent
4. Old Yellow Bricks
5. 505
Key 4: Humbug
1. My Propeller
2. Crying Lightning
3. Secret Door
Key 5: AM
1. Do I Wanna Know?
2. R U Mine?
3. Arabella

INI ISI PLAYLIST
Judul 1: BLACKPINK My Top Three
1. BLACKPINK - THE ALBUM - How You Like That
2. BLACKPINK - BORN PINK - Typa Girl
Judul 2: Arctic Monkeys My Top 5
1. Arctic Monkeys - AM - R U Mine?
2. Arctic Monkeys - AM - Do I Wanna Know?
3. Arctic Monkeys - AM - Arabella
4. Arctic Monkeys - Favourite Worst Nightmare - 505
5. Arctic Monkeys - Humbug - Pretty Visitors
Judul 3: Mixed Playlist
Kosong
```

Gambar 1.1 Display(Kondisi awal)



```
4. BLACKPINK - BORN PINK - Shut Down
5. Arctic Monkeys - AM - Arabella

INI RIMAYAT LAGU
1. Arctic Monkeys - Favourite Worst Nightmare - 505
2. BLACKPINK - BORN PINK - Pink Venom

INI MAP ALBUM : PENYANYI
Key 1: BLACKPINK
1. BORN PINK
2. THE ALBUM
Key 2: Arctic Monkeys
1. Favourite Worst Nightmare
2. Humbug
3. AM

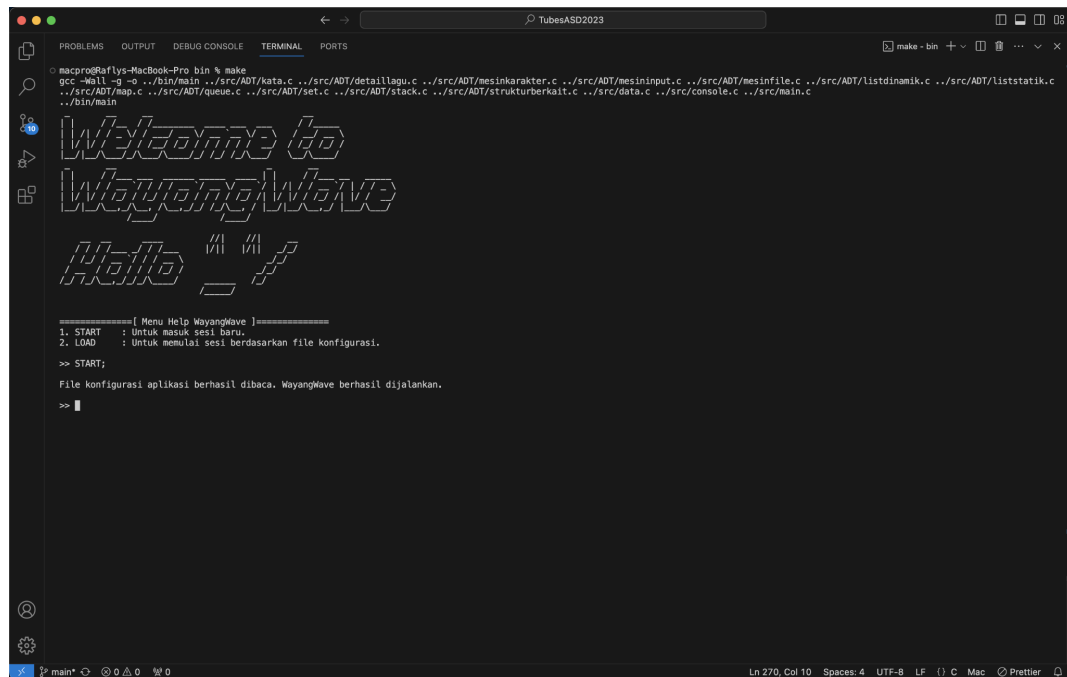
INI MAP LAGU : ALBUM
Key 1: BORN PINK
1. Pink Venom
2. Shut Down
3. Typa Girl
4. Ready For Love
Key 2: THE ALBUM
1. How You Like That
2. Ice Cream (with Selena Gomez)
3. Bet You Wanna (Feat. Cardi B)
Key 3: Favourite Worst Nightmare
1. Brianstorm
2. Teddy Picker
3. Fluorescent Adolescent
4. Old Yellow Bricks
5. 505
Key 4: Humbug
1. My Propeller
2. Crying Lightning
3. Secret Door
Key 5: AM
1. Do I Wanna Know?
2. R U Mine?
3. Arabella

INI ISI PLAYLIST
Judul 1: BLACKPINK My Top Three
1. BLACKPINK - THE ALBUM - How You Like That
2. BLACKPINK - BORN PINK - Typa Girl
Judul 2: Arctic Monkeys My Top 5
1. Arctic Monkeys - AM - R U Mine?
2. Arctic Monkeys - AM - Do I Wanna Know?
3. Arctic Monkeys - AM - Arabella
4. Arctic Monkeys - Favourite Worst Nightmare - 505
5. Arctic Monkeys - Humbug - Pretty Visitors
Judul 3: Mixed Playlist
Kosong
```

Gambar 1.2 Display (Kondisi awal)

## 5.1 Data Test START

Pada menu ini, hal yang dites adalah mengenai menjalankan WayangWave tanpa menggunakan file konfigurasi yang sudah di save, bila berhasil maka akan mencetak pesan file konfigurasi berhasil dibaca dan WayangWave berhasil dijalankan. Berikut adalah data testnya



```
macproRaflys-MacBook-Pro bin % make
gcc -Wall -g -O -D -I../src/ADT/kata.c ../src/ADT/detailagu.c ../src/ADT/mesinkarakter.c ../src/ADT/mesininput.c ../src/ADT/mesinfile.c ../src/ADT/listdinamik.c ../src/ADT/liststatik.c
../src/ADT/map.c ../src/ADT/queue.c ../src/ADT/set.c ../src/ADT/stack.c ../src/ADT/strukturberkait.c ../src/data.c ../src/console.c ../src/main.c
../bin/main

Welcome to
WayangWave

Hello _P

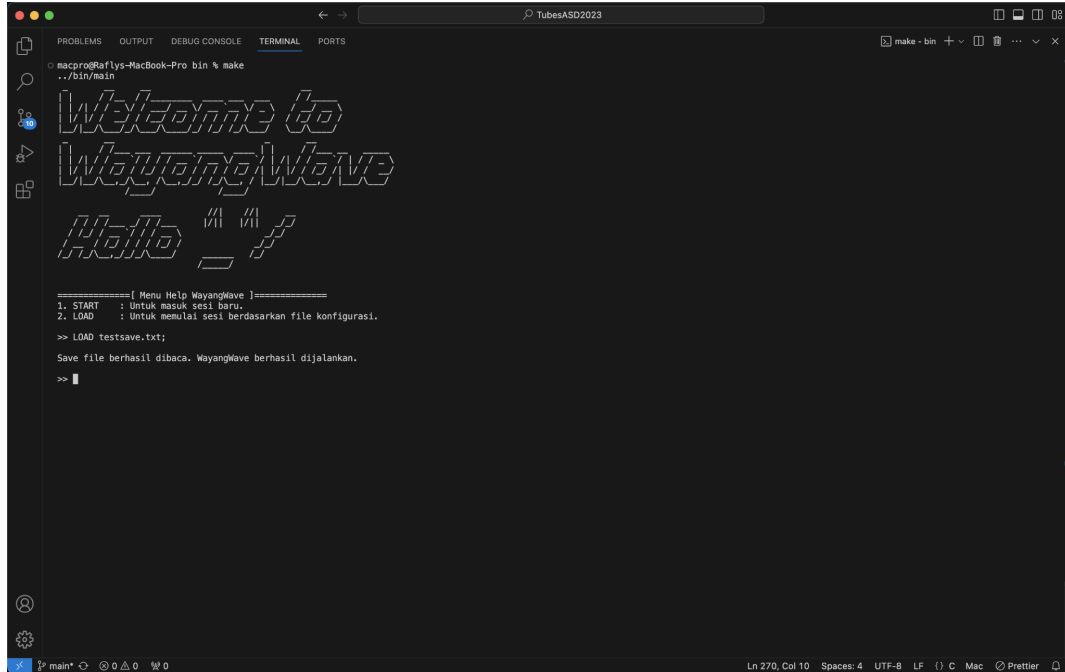
===== [ Menu Help WayangWave ] =====
1. START : Untuk masuk sesi baru.
2. LOAD  : Untuk memulai sesi berdasarkan file konfigurasi.
>> START;

File konfigurasi aplikasi berhasil dibaca. WayangWave berhasil dijalankan.
>>
```

Gambar 2 Data Test START

## 5.2 Data Test LOAD

Hal yang dites untuk ini adalah kemampuan untuk membaca file konfigurasi yang sudah pernah disimpan saat menjalankan WayangWave sebelumnya, Apabila berhasil maka akan mencetak save file berhasil dibaca dan WayangWave berhasil dijalankan. Apabila tidak berhasil, maka akan menampilkan save file tidak ditemukan dan WayangWave gagal dijalankan



Gambar 3 Data Test LOAD

## 5.3 Data Test LIST

### 6.3.1 Data Test List Default

Pada list default bisa dicek mengenai daftar penyanyi yang sudah dimiliki user. Berdasarkan penyanyi yang dipilih user, bisa dicek album album yang dibuat penyanyi yang dipilih dan dimiliki user. Berdasarkan album yang dipilih, user dapat melihat lagu - lagu yang berada di dalam album tersebut. Berikut adalah data testnya.

```
>> LIST DEFAULT;

Daftar Penyanyi:
  1. BLACKPINK
  2. Arctic Monkeys

Ingin melihat album yang ada? (Y/N): Y;
Pilih penyanyi untuk melihat album mereka: BLACKPINK;

Daftar Album oleh BLACKPINK :
  1. BORN PINK
  2. THE ALBUM

Ingin melihat lagu yang ada? (Y/N): Y;
Pilih album untuk melihat lagu yang ada di album: BORN PINK;
Daftar Lagu Album BORN PINK oleh BLACKPINK :
  1. Pink Venom
  2. Shut Down
  3. Typa Girl
  4. Ready For Love

>>
```

Gambar 4.1.1 Data Test LIST DEFAULT

Apabila pada saat menjalankan LIST DEFAULT lalu dipilih tidak ingin melihat album atau lagu, maka WayangWave bisa menjalankan perintah lainnya, berikut adalah data testnya

```
>> LIST DEFAULT;

Daftar Penyanyi:
  1. BLACKPINK
  2. Arctic Monkeys

Ingin melihat album yang ada? (Y/N): N;

>>
```

Gambar 4.1.2 Data Test LIST DEFAULT

Apabila terjadi kesalahan dalam melakukan input pada LIST DEFAULT, berikut adalah data testnya.

```
>> LIST DEFAULT;

Daftar Penyanyi:
  1. BLACKPINK
  2. Arctic Monkeys

Ingin melihat album yang ada? (Y/N): RANDOM;
Input harus Y/N: RANDOM2;
Input harus Y/N: Y;
Pilih penyanyi untuk melihat album mereka: RANDOM;

Penyanyi RANDOM tidak ada dalam daftar. Silakan coba lagi.

>>
```

Gambar 4.1.3 Data Test LIST DEFAULT gagal

### 6.3.2 Data Test List Playlist

Pada bagian ini hanya akan ditampilkan mengenai playlist yang dimiliki user, bisa kosong maupun ada. Apabila user tidak memiliki playlist maka akan dicetak keterangan tidak memiliki playlist, namun bila memiliki playlist maka akan dicetak nama-nama playlist yang tersedia. Berikut adalah data testnya

```
>> LIST PLAYLIST;

Daftar playlist yang kamu miliki:
  1. BLACKPINK My Top Three
  2. Arctic Monkeys My Top 5
  3. Mixed Playlist

>>
```

Gambar 4.2.1 Data Test LIST PLAYLIST



```
>> LIST PLAYLIST;

Daftar playlist yang kamu miliki:
Kamu tidak memiliki playlist.

>> █
```

Gambar 4.2.2 Data Test LIST PLAYLIST (playlist kosong)

## 5.4 Data Test PLAY

### 6.4.1 Data Test Play Song

Pada bagian ini ditampilkan mengenai pemilihan lagu yang diputar, diawali dengan user memilih penyanyi, lalu memilih album, dan diakhiri dengan memilih lagu yang berada di album. Berikut adalah data testnya apabila berhasil hingga memutar lagu.

```
>> PLAY SONG;

Daftar Penyanyi :
1. BLACKPINK
2. Arctic Monkeys

Masukkan Nama Penyanyi yang dipilih : Arctic Monkeys;

Daftar Album oleh Arctic Monkeys :
1. Favourite Worst Nightmare
2. Humbug
3. AM

Masukkan Judul Album yang dipilih : Favourite Worst Nightmare;

Daftar Lagu Album Favourite Worst Nightmare oleh Arctic Monkeys :
1. Brianstorm
2. Teddy Picker
3. Fluorescent Adolescent
4. Old Yellow Bricks
5. 505

Masukkan ID Lagu yang dipilih : 2;

Memutar lagu "Teddy Picker" oleh "Arctic Monkeys".

>> █
```

Gambar 5.1.1 Data Test PLAY SONG

Dikarenakan berhasil memutar lagu, maka status yang ada sebelumnya pun akan berubah, berikut adalah tampilan status dikarenakan pemrosesan Play Song.

```

INI CURRENT LAGU
Arctic Monkeys – Favourite Worst Nightmare – Teddy Picker

INI QUEUE LAGU
Kosong

INI RIWAYAT LAGU
Kosong

```

**Gambar 5.1.2 Kondisi setelah PLAY SONG**

Apabila pada saat proses input terjadi ketidaksesuaian data sehingga gagal memutar lagu, maka akan ditampilkan keterangan bahwa yang diinput tidak tersedia dan diminta untuk mencoba lagi. Berikut adalah hasil dari data testnya.

```

>> PLAY SONG;

Daftar Penyanyi :
1. BLACKPINK
2. Arctic Monkeys

Masukkan Nama Penyanyi yang dipilih : WHITEDOG;

Penyanyi WHITEDOG tidak ada dalam daftar. Silakan coba lagi.

>>

```

**Gambar 5.1.3 Data Test PLAY SONG gagal**

## 6.4.2 Data Test Play Playlist

Pada bagian ini akan ditampilkan ketika menjalankan playlist, user diminta untuk memasukkan ID dari playlist yang dipilih. Bila ID yang dipilih benar maka playlist tersebut akan dimainkan, berikut adalah data testnya.

```

>> PLAY PLAYLIST;

Masukkan ID Playlist : 2;

Memutar playlist "Arctic Monkeys My Top 5".

>>

```

**Gambar 5.2.1 Data Test PLAY PLAYLIST**

Dikarenakan berhasil menjalankan playlist, maka hal ini berdampak pada berubahnya current lagu dan queue lagu sebelumnya, berikut adalah tampilan current lagu dan queue lagu sekarang setelah play playlist berhasil

```

INI CURRENT LAGU
Arctic Monkeys - AM - R U Mine?

INI QUEUE LAGU
1. Arctic Monkeys - AM - Do I Wanna Know?
2. Arctic Monkeys - AM - Arabella
3. Arctic Monkeys - Favourite Worst Nightmare - 505
4. Arctic Monkeys - Humbug - Pretty Visitors

INI RIWAYAT LAGU
1. Arctic Monkeys - Humbug - Pretty Visitors
2. Arctic Monkeys - Favourite Worst Nightmare - 505
3. Arctic Monkeys - AM - Arabella
4. Arctic Monkeys - AM - Do I Wanna Know?
5. Arctic Monkeys - AM - R U Mine?

```

Gambar 5.2.2 Kondisi setelah PLAY PLAYLIST

Apabila ID playlist yang dipilih tidak tersedia, maka akan ditampilkan pesan berupa ID playlist tidak ada dalam daftar playlist. Berikut adalah data testnya.

```

>> PLAY PLAYLIST;

Masukkan ID Playlist : 4;

ID Playlist 4 tidak ada dalam daftar. Silakan coba lagi.

>>

```

Gambar 5.2.3 Data Test PLAY PLAYLIST gagal

## 5.5 Data Test QUEUE

### 6.5.1 Data Test Queue Song

Pada bagian ini terjadi proses penambahan lagu ke dalam urutan queue dengan urutan yang paling terakhir. Berikut adalah data testnya.

```

>> QUEUE SONG;

Daftar Penyanyi :
1. BLACKPINK
2. Arctic Monkeys

Masukkan Nama Penyanyi : Arctic Monkeys;

Daftar Album oleh Arctic Monkeys :
1. Favourite Worst Nightmare
2. Humbug
3. AM

Masukkan Judul Album yang dipilih : Humbug;

Daftar Lagu Album Humbug oleh Arctic Monkeys :
1. My Propeller
2. Crying Lightning
3. Secret Door
4. Pretty Visitors

Masukkan ID Lagu yang dipilih : 4;

Berhasil menambahkan lagu "Pretty Visitors" oleh "Arctic Monkeys" ke queue.

>>

```

Gambar 6.1.1 Data Test QUEUE SONG

Dikarenakan berhasil menambahkan lagu baru ke queue, maka akan dihasilkan urutan baru pada queue seperti ditampilkan di bawah ini setelah menambahkan lagu Pretty Visitors ke queue

```
INI QUEUE LAGU
1. BLACKPINK - BORN PINK - Shut Down
2. BLACKPINK - THE ALBUM - How You Like That
3. Arctic Monkeys - AM - Do I Wanna Know?
4. BLACKPINK - BORN PINK - Typa Girl
5. Arctic Monkeys - AM - Arabella
```

Gambar 6.1.2 Kondisi setelah QUEUE SONG

Berikut adalah kejadian gagal dalam proses penginputan sehingga muncul output bahwa input tidak tersedia di dalam daftar

```
>> QUEUE SONG;

Daftar Penyanyi :
1. BLACKPINK
2. Arctic Monkeys

Masukkan Nama Penyanyi : Ada deh;
Penyanyi Ada deh tidak ada dalam daftar. Silakan coba lagi.

>>
```

Gambar 6.1.3 Data Test QUEUE SONG gagal

## 6.5.2 Data Test Queue Playlist

Pada bagian ini terjadi proses penambahan Playlist pada queue yang ada. Prosesnya dilakukan dengan memberikan input ID dari playlist yang dipilih. Bila berhasil maka akan dikeluarkan output berupa nama dari playlist yang dipilih berhasil ditambahkan ke queue. Berikut adalah data testnya.

```
>> QUEUE PLAYLIST;

Masukkan ID Playlist : 2;

Berhasil menambahkan playlist "Arctic Monkeys My Top 5" ke queue.

>>
```

Gambar 6.2.1 Data Test QUEUE PLAYLIST

Setelah proses tersebut terjadi, maka lagu yang terdapat pada playlist akan ditambahkan ke queue. Berikut adalah kondisi queue setelah dilakukan queue playlist seperti diatas.

```

INI QUEUE LAGU
1. BLACKPINK - BORN PINK - Typa Girl
2. BLACKPINK - THE ALBUM - How You Like That
3. Arctic Monkeys - AM - Do I Wanna Know?
4. BLACKPINK - BORN PINK - Shut Down
5. Arctic Monkeys - AM - Arabella
6. Arctic Monkeys - AM - R U Mine?
7. Arctic Monkeys - AM - Do I Wanna Know?
8. Arctic Monkeys - AM - Arabella
9. Arctic Monkeys - Favourite Worst Nightmare - 505
10. Arctic Monkeys - Humbug - Pretty Visitors

```

Gambar 6.2.2 Kondisi setelah QUEUE PLAYLIST

### 6.5.3 Data Test Queue Swap

Pada bagian ini akan dilakukan proses swap urutan pada lagu yang terdapat pada queue, Bila proses berhasil maka akan ditampilkan pernyataan bahwa swap berhasil. Berikut adalah data testnya.

```

>> QUEUE SWAP 1 4;
Lagu "Typa Girl" berhasil ditukar dengan "Shut Down"
>>

```

Gambar 6.3.1 Data Test QUEUE SWAP

Dikarenakan hal tersebut, maka kondisi queue saat ini seperti yang ditampilkan pada gambar berikut.

```

INI QUEUE LAGU
1. BLACKPINK - BORN PINK - Shut Down
2. BLACKPINK - THE ALBUM - How You Like That
3. Arctic Monkeys - AM - Do I Wanna Know?
4. BLACKPINK - BORN PINK - Typa Girl
5. Arctic Monkeys - AM - Arabella

```

Gambar 6.3.2 Kondisi setelah QUEUE SWAP

Bila ternyata urutan queue yang dipilih tidak ada salah satu atau dua-duanya, maka berikut adalah data testnya.

```

>> QUEUE SWAP 80 3;
Lagu dengan urutan ke 80 atau 3 tidak terdapat dalam queue!
>>

```

Gambar 6.3.3 QUEUE SWAP gagal

#### 6.5.4 Data Test Queue Remove

Pada bagian ini akan dipilih urutan queue yang ingin dihapus dari queue yang tersedia. Bila berhasil, maka akan ditampilkan pesan bahwa lagu dari album berhasil dihapus dari queue dan urutan dari queue lagu akan diperbaharui. Berikut adalah data testnya.

```
>> QUEUE REMOVE 3;  
Lagu "Do I Wanna Know?" oleh "Arctic Monkeys" telah dihapus dari queue!  
>> █
```

Gambar 6.4.1 Data Test QUEUE REMOVE

Dikarenakan berhasil menghapus salah satu lagu di queue, maka bisa saja urutan di queue berubah, berikut adalah kondisi urutan queue saat ini.

```
INI QUEUE LAGU  
1. BLACKPINK - BORN PINK - Tyla Girl  
2. BLACKPINK - THE ALBUM - How You Like That  
3. BLACKPINK - BORN PINK - Shut Down  
4. Arctic Monkeys - AM - Arabella
```

Gambar 6.4.2 Kondisi setelah QUEUE REMOVE

Bila diproses ternyata urutan tidak tersedia lagu, maka akan ditampilkan data test di bawah ini.

```
>> QUEUE REMOVE 420;  
Lagu dengan urutan ke 420 tidak ada.  
>> █
```

Gambar 6.4.3 Data Test QUEUE REMOVE gagal

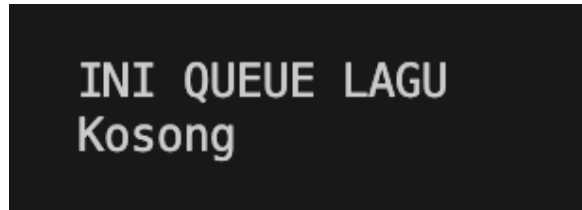
#### 6.5.5 Data Test Queue Clear

Pada bagian ini akan dihapus semua queue yang tersedia, bila berhasil maka akan dihasilkan output queue berhasil dikosongkan. Berikut adalah data testnya.

```
>> QUEUE CLEAR;  
Queue berhasil dikosongkan.  
>> █
```

Gambar 6.5.1 Data Test QUEUE CLEAR

Dikarenakan sudah dihapus semua lagu pada queue, maka queue sekarang harus kosong, berikut adalah isi queue saat ini.



Gambar 6.5.2 Kondisi setelah QUEUE CLEAR

## 5.6 Data Test SONG

### 6.6.1 Data Test Song Next

Pada bagian ini terjadi dua kasus, yang pertama yaitu saat queue sudah kosong, maka akan terjadi proses memainkan kembali lagu yang sedang dimainkan. Berikut adalah data testnya, dilakukan Play Song agar queue kosong

```
>> PLAY SONG;

Daftar Penyanyi :
  1. BLACKPINK
  2. Arctic Monkeys

Masukkan Nama Penyanyi yang dipilih : Arctic Monkeys;

Daftar Album oleh Arctic Monkeys :
  1. Favourite Worst Nightmare
  2. Humbug
  3. AM

Masukkan Judul Album yang dipilih : AM;

Daftar Lagu Album AM oleh Arctic Monkeys :
  1. Do I Wanna Know?
  2. R U Mine?
  3. Arabella

Masukkan ID Lagu yang dipilih : 2;

Memutar lagu "R U Mine?" oleh "Arctic Monkeys".

>> SONG NEXT;

Queue kosong, memutar kembali lagu
"R U Mine?" oleh "Arctic Monkeys"

>> |
```

Gambar 7.1.1 Data Test PLAY SONG kondisi queue kosong

Kasus kedua yaitu bila queue tidak kosong, maka akan dijalankan lagu yang berada di urutan pertama queue dan lagu yang sebelumnya dijalankan akan ditambahkan ke riwayat lagu dengan urutan yang pertama. Berikut adalah data testnya.

```
>> SONG NEXT;  
  
Memutar lagu selanjutnya  
"Typa Girl" oleh "BLACKPINK"  
  
>>
```

Gambar 7.1.2 Data Test PLAY SONG kondisi queue memiliki isi

Berikut adalah kondisi riwayat lagu, queue lagu, dan current lagu saat ini setelah berhasil dilakukan Song Next.

```
INI CURRENT LAGU  
BLACKPINK - BORN PINK - Typa Girl  
  
INI QUEUE LAGU  
2. BLACKPINK - THE ALBUM - How You Like That  
3. Arctic Monkeys - AM - Do I Wanna Know?  
4. BLACKPINK - BORN PINK - Shut Down  
5. Arctic Monkeys - AM - Arabella  
  
INI RIWAYAT LAGU  
1. Arctic Monkeys - AM - R U Mine?  
2. Arctic Monkeys - Favourite Worst Nightmare - 505  
3. BLACKPINK - BORN PINK - Pink Venom
```

Gambar 7.1.3 Kondisi setelah SONG NEXT

### 6.6.2 Data Test Song Previous

Pada bagian ini akan diputar lagu yang berada di urutan pertama riwayat lagu. Bila ternyata riwayat lagu kosong, maka akan kembali memutar lagu yang sedang diputar. Berikut adalah hasil dari data testnya.



```

>> PLAY SONG;

Daftar Penyanyi :
1. BLACKPINK
2. Arctic Monkeys

Masukkan Nama Penyanyi yang dipilih : Arctic Monkeys;

Daftar Album oleh Arctic Monkeys :
1. Favourite Worst Nightmare
2. Humbug
3. AM

Masukkan Judul Album yang dipilih : AM;

Daftar Lagu Album AM oleh Arctic Monkeys :
1. Do I Wanna Know?
2. R U Mine?
3. Arabella

Masukkan ID Lagu yang dipilih : 2;

Memutar lagu "R U Mine?" oleh "Arctic Monkeys".

>> SONG PREVIOUS;

Riwayat lagu kosong, memutar kembali lagu
R U Mine?" oleh "Arctic Monkeys"

>>

```

Gambar 7.2.1 Data Test SONG PREVIOUS kondisi riwayat lagu kosong

Pada kasus terdapat riwayat lagu, maka akan diputar lagu dengan antrian paling atas di riwayat lagu dan lagu yang sebelumnya diputar akan ditaruh ke urutan ke queue sehingga bila dilakukan song next, maka lagu tersebut akan diputar. Berikut adalah data testnya.

```

>> SONG PREVIOUS;

Memutar lagu sebelumnya
505" oleh "Arctic Monkeys"

>>

```

Gambar 7.2.2 Data Test SONG PREVIOUS kondisi riwayat lagu memiliki isi

Dikarenakan hal tersebut, terjadi perubahan pada queue lagu dan riwayat lagu. Berikut adalah kondisinya saat ini.

```

INI CURRENT LAGU
Arctic Monkeys - Favourite Worst Nightmare - 505

INI QUEUE LAGU
1. Arctic Monkeys - AM - R U Mine?
2. BLACKPINK - BORN PINK - Typa Girl
3. BLACKPINK - THE ALBUM - How You Like That
4. Arctic Monkeys - AM - Do I Wanna Know?
5. BLACKPINK - BORN PINK - Shut Down
5. Arctic Monkeys - AM - Arabella

INI RIWAYAT LAGU
1. BLACKPINK - BORN PINK - Pink Venom

```

Gambar 7.2.3 Kondisi setelah SONG PREVIOUS

## 5.7 Data Test PLAYLIST

### 6.7.1 Data Test Playlist Create

Pada bagian ini akan dibuat playlist baru dengan kondisi playlist masihlah kosong. Berikut adalah data testnya.

```

>> PLAYLIST CREATE;
Masukkan nama playlist yang ingin dibuat : Car Jams;
Playlist Car Jams berhasil dibuat! Silakan masukkan lagu - lagu artis terkini kesayangan Anda!
>> █

```

Gambar 8.1.1 Data Test PLAYLIST CREATE

Dikarenakan terciptanya playlist baru, maka di daftar playlist terjadi perubahan. Berikut adalah kondisinya saat ini.

```

INI ISI PLAYLIST
Judul 1: BLACKPINK My Top Three
1. BLACKPINK - THE ALBUM - How You Like That
2. BLACKPINK - BORN PINK - Typa Girl
3. BLACKPINK - BORN PINK - Pink Venom
Judul 2: Arctic Monkeys My Top 5
1. Arctic Monkeys - AM - R U Mine?
2. Arctic Monkeys - AM - Do I Wanna Know?
3. Arctic Monkeys - AM - Arabella
4. Arctic Monkeys - Favourite Worst Nightmare - 505
5. Arctic Monkeys - Humbug - Pretty Visitors
Judul 3: Mixed Playlist
Kosong
Judul 4: Car Jams
Kosong

```

Gambar 8.1.2 Kondisi setelah PLAYLIST CREATE

### 6.7.2 Data Test Playlist Add Song

Pada bagian ini terjadi penambahan lagu pada playlist, bila berhasil maka akan ditampilkan keterangan bahwa lagu dari album berhasil ditambahkan ke playlist. Berikut adalah data testnya.

```

>> PLAYLIST ADD SONG;

Daftar Penyanyi :
1. BLACKPINK
2. Arctic Monkeys

Masukkan Nama Penyanyi yang dipilih : Arctic Monkeys;

Daftar Album oleh Arctic Monkeys :
1. Favourite Worst Nightmare
2. Humbug
3. AM

Masukkan Judul Album yang dipilih : Favourite Worst Nightmare;

Daftar Lagu Album Favourite Worst Nightmare oleh Arctic Monkeys :
1. Brianstorm
2. Teddy Picker
3. Fluorescent Adolescent
4. Old Yellow Bricks
5. 505

Masukkan ID Lagu yang dipilih : 2;

Daftar Playlist Pengguna :
1. BLACKPINK My Top Three
2. Arctic Monkeys My Top 5
3. Mixed Playlist

Masukkan ID Playlist yang dipilih : 2;

Lagu dengan judul "Teddy Picker" pada album Favourite Worst Nightmare oleh penyanyi
Arctic Monkeys berhasil ditambahkan ke dalam playlist Arctic Monkeys My Top 5.

>>

```

**Gambar 8.2.1 Data Test PLAYLIST ADD SONG**

Dikarenakan berhasil menambahkan lagu ke playlist, maka akan terjadi perubahan banyak lagu di playlist tersebut. Berikut adalah kondisi saat ini di playlist.

```

INI ISI PLAYLIST
Judul 1: BLACKPINK My Top Three
1. BLACKPINK - THE ALBUM - How You Like That
2. BLACKPINK - BORN PINK - Typa Girl
3. BLACKPINK - BORN PINK - Pink Venom
Judul 2: Arctic Monkeys My Top 5
1. Arctic Monkeys - AM - R U Mine?
2. Arctic Monkeys - AM - Do I Wanna Know?
3. Arctic Monkeys - AM - Arabella
4. Arctic Monkeys - Favourite Worst Nightmare - 505
5. Arctic Monkeys - Humbug - Pretty Visitors
6. Arctic Monkeys - Favourite Worst Nightmare - Teddy Picker
Judul 3: Mixed Playlist
Kosong

```

**Gambar 8.2.2 Kondisi setelah PLAYLIST ADD SONG**

Namun, bila terjadi ketidaksesuaian antara input dengan data yang ada, maka akan muncul tampilan bahwa input tidak ada, silakan coba lagi. Berikut adalah data testnya.

```

>> PLAYLIST ADD SONG;

Daftar Penyanyi :
1. BLACKPINK
2. Arctic Monkeys

Masukkan Nama Penyanyi yang dipilih : Taylor Swift;

Penyanyi Taylor Swift tidak ada dalam daftar. Silakan coba lagi.

>>

```

**Gambar 8.2.3 Data Test PLAYLIST ADD SONG gagal**

### 6.7.3 Data Test Playlist Add Album

Pada bagian ini terjadi penambahan album pada playlist, sehingga lagu yang ada di album tersebut akan ditambahkan ke playlist. Bila berhasil maka akan ditampilkan bahwa album berhasil ditambahkan ke playlist. Berikut adalah data testnya.

```
>> PLAYLIST ADD ALBUM;

Daftar Penyanyi :
1. BLACKPINK
2. Arctic Monkeys

Masukkan Nama Penyanyi yang dipilih : Arctic Monkeys;

Daftar Album oleh Arctic Monkeys :
1. Favourite Worst Nightmare
2. Humbug
3. AM

Masukkan Judul Album yang dipilih : Favourite Worst Nightmare;

Daftar Playlist Pengguna :
1. BLACKPINK My Top Three
2. Arctic Monkeys My Top 5
3. Mixed Playlist

Masukkan ID Playlist yang dipilih : 3;

Album dengan judul "Favourite Worst Nightmare" berhasil ditambahkan ke dalam playlist pengguna "Mixed Playlist".

>>
```

Gambar 8.3.1 Data Test PLAYLIST ADD ALBUM

Dikarenakan penambahan album ke playlist, maka daftar lagu di playlist yang dituju akan berubah, berikut adalah kondisi terkini.

```
INI ISI PLAYLIST

Judul 1: BLACKPINK My Top Three
1. BLACKPINK - THE ALBUM - How You Like That
2. BLACKPINK - BORN PINK - Typa Girl
3. BLACKPINK - BORN PINK - Pink Venom

Judul 2: Arctic Monkeys My Top 5
1. Arctic Monkeys - AM - R U Mine?
2. Arctic Monkeys - AM - Do I Wanna Know?
3. Arctic Monkeys - AM - Arabella
4. Arctic Monkeys - Favourite Worst Nightmare - 505
5. Arctic Monkeys - Humbug - Pretty Visitors

Judul 3: Mixed Playlist
1. Arctic Monkeys - Favourite Worst Nightmare - Brianstorm
2. Arctic Monkeys - Favourite Worst Nightmare - Teddy Picker
3. Arctic Monkeys - Favourite Worst Nightmare - Fluorescent Adolescent
4. Arctic Monkeys - Favourite Worst Nightmare - Old Yellow Bricks
5. Arctic Monkeys - Favourite Worst Nightmare - 505
```

Gambar 8.3.2 Kondisi setelah PLAYLIST ADD ALBUM

### 6.7.4 Data Test Playlist Swap

Pada bagian ini terjadi penukaran urutan pada satu playlist, hanya bisa secara satu playlist, tidak bisa antar playlist. Caranya adalah dengan memasukan ID playlist dan kedua urutan yang ingin di swap. Bila berhasil, maka akan ditampilkan berhasil menukar lagu dengan lagu lainnya. Berikut adalah data testnya.

```
>> PLAYLIST SWAP 1 1 2;

Berhasil menukar lagu dengan nama "How You Like That" dengan "Typa Girl" di playlist "BLACKPINK My Top Three"

>>
```

Gambar 8.4.1 Data Test PLAYLIST SWAP

Dikarenakan berhasil swap urutan lagu, maka terjadi perubahan pada playlist tersebut. Berikut adalah kondisi playlist setelah swap dilakukan.

```
INI ISI PLAYLIST
Judul 1: BLACKPINK My Top Three
1. BLACKPINK - BORN PINK - Typa Girl
2. BLACKPINK - THE ALBUM - How You Like That
3. BLACKPINK - BORN PINK - Pink Venom
Judul 2: Arctic Monkeys My Top 5
1. Arctic Monkeys - AM - R U Mine?
2. Arctic Monkeys - AM - Do I Wanna Know?
3. Arctic Monkeys - AM - Arabella
4. Arctic Monkeys - Favourite Worst Nightmare - 505
5. Arctic Monkeys - Humbug - Pretty Visitors
Judul 3: Mixed Playlist
Kosong
```

Gambar 8.4.2 Kondisi setelah PLAYLIST SWAP

Namun bisa juga terjadi kesalahan input sehingga tidak cocok dengan semua data. Bila terjadi kesalahan di ID playlist, maka akan ditampilkan playlist dengan id tidak ada. Berikut adalah data testnya.

```
>> PLAYLIST SWAP 500 1 2;

Tidak ada playlist dengan playlist ID 500

>> |
```

Gambar 8.4.3 Data Test PLAYLIST SWAP gagal

Kesalahan bisa juga dikarenakan input urutan lagu yang tidak ada di playlist. Bila terjadi maka akan ditampilkan lagu dengan urutan tidak ada di playlist. Berikut adalah data testnya.

```
>> PLAYLIST SWAP 1 900 2;

Tidak ada lagu dengan urutan 900 atau 2 di playlist

>> |
```

Gambar 8.4.4 Data Test PLAYLIST SWAP gagal

### 6.7.5 Data Test Playlist Remove

Pada bagian ini, terjadi proses menghapus lagu pada urutan playlist. Caranya yaitu dengan melakukan input id playlist dan urutan lagu. Bila berhasil maka akan ditampilkan lagu berhasil dihapus dari playlist. Berikut adalah data testnya.

```
>> PLAYLIST REMOVE 2 4;
Lagu "505" oleh "Arctic Monkeys" telah dihapus dari playlist "Arctic Monkeys My Top 5"!
>> |
```

**Gambar 8.5.1 Data Test PLAYLIST REMOVE**

Dikarenakan hal tersebut, terjadi perubahan urutan pada playlist. Berikut adalah kondisi setelah dilakukan remove.

```
INI ISI PLAYLIST
Judul 1: BLACKPINK My Top Three
  1. BLACKPINK - THE ALBUM - How You Like That
  2. BLACKPINK - BORN PINK - Typa Girl
  3. BLACKPINK - BORN PINK - Pink Venom
Judul 2: Arctic Monkeys My Top 5
  1. Arctic Monkeys - AM - R U Mine?
  2. Arctic Monkeys - AM - Do I Wanna Know?
  3. Arctic Monkeys - AM - Arabella
  4. Arctic Monkeys - Humbug - Pretty Visitors
Judul 3: Mixed Playlist
Kosong
```

**Gambar 8.5.2 Kondisi setelah PLAYLIST REMOVE**

Namun bisa saja remove mengalami kegagalan karena input tidak sesuai dengan data. Bila kesalahan terjadi di input ID playlist, maka akan ditampilkan bahwa playlist dengan ID input tidak tersedia. Berikut adalah data testnya.

```
>> PLAYLIST REMOVE 100 2;
Tidak ada playlist dengan playlist ID 100
>> |
```

**Gambar 8.5.3 Data Test PLAYLIST REMOVE gagal**

Bila kesalahan terjadi karena urutan tidak ada di playlist, maka akan ditampilkan lagu dengan urutan tidak ada di playlist. Berikut adalah data testnya.

```
>> PLAYLIST REMOVE 2 300;
Tidak ada lagu dengan urutan 300 di playlist
>> |
```

**Gambar 8.5.4 Data Test PLAYLIST REMOVE**

### 6.7.6 Data Test Playlist Delete

Pada bagian ini terjadi proses penghapusan playlist yang telah dibuat, bila berhasil maka akan ditampilkan playlist dengan judul telah dihapus. Berikut adalah data testnya.

```
>> PLAYLIST DELETE;

Daftar Playlist Pengguna :
  1. BLACKPINK My Top Three
  2. Arctic Monkeys My Top 5
  3. Mixed Playlist

Masukkan ID Playlist yang dipilih : 2;

Playlist ID 2 dengan judul "Arctic Monkeys My Top 5" berhasil dihapus.

>> █
```

Gambar 8.6.1 Data Test PLAYLIST DELETE

Dikarenakan hal tersebut, terjadi perubahan pada daftar playlist. Berikut adalah kondisi daftar playlist setelah dilakukan delete.

```
INI ISI PLAYLIST
Judul 1: BLACKPINK My Top Three
  1. BLACKPINK - THE ALBUM - How You Like That
  2. BLACKPINK - BORN PINK - Typa Girl
  3. BLACKPINK - BORN PINK - Pink Venom
Judul 2: Mixed Playlist
Kosong

>> █
```

Gambar 8.6.2 Kondisi setelah PLAYLIST DELETE

Bila terjadi kesalahan input ID playlist sehingga tidak ditemukan, maka akan ditampilkan playlist dengan id tidak ada dalam daftar. Berikut adalah data testnya.

```
Daftar Playlist Pengguna :
  1. BLACKPINK My Top Three
  2. Arctic Monkeys My Top 5
  3. Mixed Playlist

Masukkan ID Playlist yang dipilih : 4;

Tidak ada playlist dengan ID 4 dalam daftar playlist pengguna. Silakan coba lagi.

>> █
```

Gambar 8.6.3 Data Test PLAYLIST DELETE gagal

## 5.8 Data Test STATUS

Pada bagian ini digunakan untuk mengecek lagu yang sedang diputar, playlist yang sedang diputar dan queue lagu saat ini. Berikut adalah data testnya bila memutar playlist.

```

>> PLAY PLAYLIST;

Masukkan ID Playlist : 2;

Memutar playlist "Arctic Monkeys My Top 5".

>> STATUS;

Current Playlist: Arctic Monkeys My Top 5

Now Playing:
Arctic Monkeys - AM - R U Mine?

Queue:
1. Arctic Monkeys - AM - Do I Wanna Know?
2. Arctic Monkeys - AM - Arabella
3. Arctic Monkeys - Favourite Worst Nightmare - 505
4. Arctic Monkeys - Humbug - Pretty Visitors

>>

```

Gambar 9.1 Data Test STATUS saat PLAY PLAYLIST

Berikut adalah data testnya bila hanya memutar lagu saja dan memiliki queue lagu.

```

>> STATUS;

Now Playing:
Arctic Monkeys - AM - R U Mine?

Queue:
1. BLACKPINK - BORN PINK - Typa Girl
2. BLACKPINK - THE ALBUM - How You Like That
3. Arctic Monkeys - AM - Do I Wanna Know?
4. BLACKPINK - BORN PINK - Shut Down
5. Arctic Monkeys - AM - Arabella

>>

```

Gambar 9.2 Data Test STATUS saat sedang putar lagu dan punya queue

Berikut adalah data testnya bila ternyata tidak ada lagu yang diputar dan pastinya queue kosong.

```

>> STATUS;

Now Playing:
No songs have been played yet. Please search for a song to begin playback.

Queue:
Your queue is empty.

>>

```

Gambar 9.3 Data Test STATUS lagu diputar kosong dan queue kosong

## 5.9 Data Test SAVE

Pada bagian ini dilakukan proses save atas semua data dari WayangWave yang barusan dilakukan. Bila berhasil, maka akan ditampilkan save file berhasil disimpan. Berikut adalah hasil data testnya.



```
>> SAVE random_name.txt;

Save file berhasil disimpan.

>> █
```

Gambar 10.1 Data Test SAVE

## 5.10 Data Test QUIT

Pada bagian ini user akan keluar dari WayangWave, sebelum betul-betul keluar, user akan diberi pilihan untuk menyimpan data sebelumnya ataupun tidak. Bila user memilih iya maka user akan melakukan input nama file dan file berhasil dibuat. Berikut adalah data testnya.

```
>> QUIT;

Apakah kamu ingin menyimpan data sesi sekarang? Y;

Input nama save file : quit_file.txt;

Save file berhasil disimpan.

Kamu keluar dari
WayangWave

Dede /// /// ///

macpro@Raflys-MacBook-Pro bin % █
```

Gambar 11.1 Data Test QUIT melakukan save

User bisa juga tanpa menyimpan data, Berikut adalah data testnya.

```
>> QUIT;

Apakah kamu ingin menyimpan data sesi sekarang? N;

Keluar tanpa save file.

Kamu keluar dari
WayangWave

Dede /// /// ///

macpro@Raflys-MacBook-Pro bin % █
```

## 5.11 Data Test HELP

### 6.11.1 Data Test Help Sebelum Masuk

Pada bagian ini akan diberikan tampilan 2 pilihan yang tersedia yaitu START dan LOAD. Berikut adalah data testnya.

```
>> HELP;

===== [ Menu Help WayangWave ] =====
1. START   : Untuk masuk sesi baru.
2. LOAD    : Untuk memulai sesi berdasarkan file konfigurasi.

>> █
```

Gambar 12.1.1 Data Test HELP sebelum masuk WayangWave

### 6.11.2 Data Test Help Setelah Masuk

Pada bagian ini diberikan banyak pilihan yang dapat dilakukan user dalam menggunakan WayangWave. Berikut adalah data testnya.

```
>> HELP;

===== [ Menu Help WayangWave ] =====
1. LIST     : Untuk menampilkan playlist, daftar penyanyi, daftar album dan daftar lagu yang ada di album.
2. PLAY     : Untuk memutar lagu atau playlist yang dipilih.
3. QUEUE    : Untuk memanipulasi lagu dengan menambahkan, menukar, menghapus atau mengosongkan lagu.
4. SONG     : Untuk navigasi lagu.
5. PLAYLIST : Basic command untuk playlist yaitu CREATE, ADD, SWAP, REMOVE dan DELETE.
6. STATUS   : Untuk menampilkan lagu yang sedang dimainkan .
7. SAVE     : Untuk menyimpan state aplikasi terbaru ke dalam suatu file.
8. QUIT     : Untuk keluar dari aplikasi WayangWave.

>> █
```

Gambar 12.2.1 Data Test HELP setelah masuk WayangWave

Bila user melakukan perintah permainan yang tidak sesuai / tidak ada di help, maka akan ditampilkan command tidak bisa dieksekusi. Berikut adalah data testnya.

```
>> LOAD testsave.txt;

Command tidak bisa dieksekusi!

>> █
```

Gambar 12.2.2 Data Test kesalahan command

## 6 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Fitur START	Memastikan bahwa aplikasi pemutaran musik WayangWave dapat dijalankan	1. Masukkan command START;	START;	Program membaca file default config.txt dan menampilkan pesan "File konfigurasi sistem berhasil dibaca. WayangWave berhasil dijalankan."	Sesuai yang diharapkan
2	Fitur LOAD	Memastikan apakah file konfigurasi yang sudah di save sebelumnya dapat dibaca	1. Masukkan command LOAD <filename.txt>;	LOAD testsave.txt ;	1. Program membaca file testsave.txt yang ada pada folder save 2. Jika memanggil command LOAD dengan file yang tidak ada pada folder save, akan ditampilkan pesan "Save file tidak ditemukan. WayangWave gagal dijalankan." 3. Jika LOAD dieksekusi setelah memasuki	Sesuai yang diharapkan

					sesi, akan ditampilkan pesan "Command tidak bisa dieksekusi!"	
3	Fitur LIST DEFAULT	Memastikan bahwa program menampilkan semua penyanyi, album, dan lagu dari file.txt yang digunakan	<ol style="list-style-type: none"> <li>1. Masukkan command LIST DEFAULT;</li> <li>2. Masukkan pilihan apakah ingin melihat album atau tidak</li> <li>3. Masukkan nama penyanyi yang tersedia</li> <li>4. Masukkan pilihan apakah ingin melihat lagu atau tidak</li> <li>5. Masukkan nama album yang tersedia</li> </ol>	LIST DEFAULT; Y; BLACKPINK; Y; BORN PINK;	<ol style="list-style-type: none"> <li>1. Program menampilkan list penyanyi, list album dari penyanyi yang dipilih, dan list lagu dari album yang dipilih yang ada di file.txt yang digunakan</li> <li>2. Jika LIST DEFAULT dieksekusi sebelum memasuki sesi, akan ditampilkan pesan "Command tidak bisa dieksekusi!"</li> </ol>	Sesuai yang diharapkan
4	Fitur LIST PLAYLIST	Memastikan bahwa program menampilkan semua playlist yang ada pada pengguna	<ol style="list-style-type: none"> <li>1. Masukkan command LIST PLAYLIST;</li> </ol>	LIST PLAYLIST;	Program menampilkan semua playlist yang dimiliki oleh pengguna	Sesuai yang diharapkan
5	Fitur PLAY SONG	Memastikan bahwa program memainkan lagu berdasarkan	<ol style="list-style-type: none"> <li>1. Masukkan command PLAY SONG;</li> </ol>	PLAY SONG;	Program memainkan lagu berdasarkan	Sesuai yang diharapkan

		masukkan nama penyayi, nama album, dan id lagu	<ol style="list-style-type: none"> <li>Masukkan nama penyayi yang tersedia</li> <li>Masukkan nama album yang tersedia</li> <li>Masukkan id lagu yang tersedia</li> </ol>	Arctic Monkeys;  Favourite Worst Nighmare;  2;	masukkan nama penyayi, nama album, dan id lagu. Status queue dan riwayat lagu menjadi kosong, serta current song adalah lagu yang sedang dimainkan.	
6	Fitur PLAY PLAYLIST	Memastikan bahwa program memainkan lagu berdasarkan id playlist	<ol style="list-style-type: none"> <li>Masukkan command PLAY PLAYLIST;</li> <li>Masukkan id playlist yang tersedia</li> </ol>	PLAY PLAYLIST;  2;	Program memainkan lagu berdasarkan id playlist. Status current song adalah lagu dengan urutan pertama playlist, queue berisi semua lagu dalam playlist dan riwayat lagu berisi urutan lagu dari queue yang di-reverse	Sesuai yang diharapkan
7	Fitur QUEUE SONG	Memastikan bahwa program memasukkan lagu yang dipilih pengguna ke dalam antrian lagu.	<ol style="list-style-type: none"> <li>Masukkan command QUEUE SONG;</li> <li>Masukkan nama penyayi yang tersedia</li> <li>Masukkan nama album yang tersedia</li> <li>Masukkan id lagu yang tersedia</li> </ol>	QUEUE SONG;  Arctic Monkeys;  Humbug;  4;	Program menampilkan pesan "Berhasil menambahkan lagu "nama_lagu" oleh "nama_penyayi" ke queue."	Sesuai yang diharapkan

8	Fitur QUEUE PLAYLIST	Memastikan bahwa program memasukkan playlist yang dipilih pengguna ke dalam antrian lagu.	1. Masukkan command QUEUE PLAYLIST; 2. Masukkan id playlist yang ingin dimasukkan ke dalam antrian	QUEUE PLAYLIST; 2;	Program menampilkan pesan "Berhasil menambahkan playlist "nama_playlist" ke queue."	Sesuai yang diharapkan
9	Fitur QUEUE SWAP	Memastikan bahwa program dapat menukar urutan antrian lagu ke urutan antrian lagu lainnya yang diinginkan pengguna.	1. Masukkan command QUEUE SWAP <x> <y>;	QUEUE SWAP 1 4;	Program menukar urutan lagu x ke urutan lagu y pada antrian lagu	Sesuai yang diharapkan
10	Fitur QUEUE REMOVE	Memastikan bahwa program menghapus lagu dari antrian lagu dengan menginput id lagu yang ingin dihapus pengguna	1. Masukkan command QUEUE REMOVE; <id>;	QUEUE REMOVE 3;	Program menghapus lagu dari antrian lagu	Sesuai yang diharapkan
12	Fitur SONG NEXT	Memastikan bahwa program memainkan lagu yang ada di dalam urutan lagu. Jika antrian lagu kosong, lagu yang sedang diputar diulang kembali	1. Masukkan command SONG NEXT;	SONG NEXT;	1. Program memutar lagu yang ada di urutan pertama pada queue lagu 2. Kondisi antrian lagu kosong: memutar kembali lagu yang sedang diputar	Sesuai yang diharapkan

13	Fitur SONG PREVIOUS	Memastikan bahwa program memainkan lagu yang terakhir kali diputar oleh pengguna. Jika riwayat lagu kosong, lagu yang sedang diputar diulang kembali.	1. Masukkan command SONG PREVIOUS;	SONG PREVIOUS;	1. Program memutar lagu yang ada di urutan pertama pada riwayat lagu 2. Kondisi riwayat lagu kosong: memutar kembali lagu yang sedang diputar	Sesuai yang diharapkan
14	Fitur PLAYLIST CREATE	Memastikan apakah playlist yang dibuat oleh pengguna ditambahkan ke daftar playlist pengguna	1. Masukkan command PLAYLIST CREATE; 2. Masukkan nama playlist 3. Memanggil command LIST PLAYLIST; untuk memeriksa playlist baru ada di daftar playlist	PLAYLIST CREATE;  Cars Jams;  LIST PLAYLIST;	Program menyimpan nama playlist baru ke dalam daftar playlist pengguna	Sesuai yang diharapkan
15	Fitur PLAYLIST ADD SONG	Memastikan bahwa program menambahkan satu lagu ke playlist yang dipilih pengguna	1. Masukkan command PLAYLIST ADD SONG; 2. Masukkan nama penyanyi yang tersedia 3. Masukkan judul album yang tersedia 4. Masukkan id lagu yang tersedia	PLAYLIST ADD SONG;  Arctic Monkeys;  Favourite Worst Nightmare;  2;	Menambahkan satu spesifik lagu yang ada pada album ke playlist yang dipilih pengguna. Lagu yang ditambahkan hanyalah lagu pada album yang belum	Sesuai yang diharapkan

			5. Masukkan id playlist yang tersedia	2;	ada di playlist yang dipilih	
16	Fitur PLAYLIST ADD ALBUM	Memastikan bahwa program menambahkan semua lagu yang ada pada album ke playlist yang dipilih pengguna	1. Masukkan command PLAYLIST ADD ALBUM; 2. Masukkan nama penyanyi yang tersedia 3. Masukkan judul album yang tersedia 4. Masukkan id playlist yang tersedia	PLAYLIST ADD ALBUM;  Arctic Monkeys;  Favourite Worst Nightmare;  3;	Menambahkan semua lagu yang ada pada album ke playlist yang dipilih pengguna. Lagu yang ditambahkan hanyalah lagu pada album yang belum ada di playlist yang dipilih	Sesuai yang diharapkan
17	Fitur PLAYLIST SWAP	Memastikan bahwa program menukar urutan lagu yang ada pada suatu playlist ke urutan lagu lain dari playlist yang sama	1. Masukkan command PLAYLIST SWAP <id> <x> <y>;	PLAYLIST SWAP 1 1 2;	Program menukar urutan lagu ke-x dari id playlist yang dipilih pengguna ke urutan lagu ke-y pada playlist yg sama	Sesuai yang diharapkan
18	Fitur PLAYLIST REMOVE	Memastikan bahwa program menghapus suatu lagu pada playlist yang ingin dihapus oleh pengguna	1. Masukkan command PLAYLIST REMOVE <id> <x>;	PLAYLIST REMOVE 2 4;	Program menghapus lagu pada urutan ke-x di playlist dengan id yg dipilih pengguna	Sesuai yang diharapkan
19	Fitur PLAYLIST DELETE	Memastikan bahwa program menghapus suatu playlist yang ingin dihapus oleh pengguna	1. Masukkan command PLAYLIST DELETE; 2. Masukkan id playlist yang ingin dihapus;	PLAYLIST DELETE;  2;	Program menghapus playlist berdasarkan id playlist yang dipilih pengguna untuk dihapus	Sesuai yang diharapkan
20	Fitur STATUS	Memastikan bahwa program menampilkan judul lagu yang	1. Masukkan command STATUS;	STATUS;	Program menampilkan status dari lagu yang sedang	Sesuai yang diharapkan



		sedang diputar dan daftar antrian lagu			diputar, daftar antrian lagu beserta nama playlist jika sebelumnya memasukkan command PLAY PLAYLIST;	
21	Fitur SAVE	Memastikan state aplikasi tersimpan pada nama file yang diinginkan pengguna	1. Masukkan command SAVE <filename.txt>;	SAVE random_name.txt;	Program menyimpan state aplikasi ke dalam folder save dengan nama file yang dimasukkan pengguna	Sesuai yang diharapkan
22	Fitur QUIT	Memastikan bahwa program menanyakan pengguna apakah ingin menyimpan state aplikasi saat ini atau tidak, mengeksekusi masukkan pengguna, dan keluar dari program WayangWave	1. Masukkan command QUIT; 2. Masukkan pilihan apakah ingin menyimpan state aplikasi atau tidak	QUIT; Y; quit_file.txt;	1. Program meminta input Y/ N sebagai dasar eksekusi penyimpanan file 2. Jika Y maka program memanggil fungsi SAVE 3. Jika tidak, tidak melakukan apapun 4. Keluar dari aplikasi	Sesuai yang diharapkan
23	Fitur HELP	Memastikan bahwa program menampilkan pesan bantuan yang menjelaskan command yang dapat dimasukkan oleh	1. Masukkan command HELP;	HELP;	1. Command HELP yang dipanggil sebelum memasuki sesi hanya akan menampilkan	Sesuai yang diharapkan

		user beserta kegunaannya			<p>an pesan bantuan dan kegunaan untuk command START dan LOAD</p> <p>2. Dipanggil setelah memasuki sesi, menampilkan pesan bantuan yang menjelaskan seluruh command yang dapat dimasukkan oleh user beserta kegunaannya</p>	
24	INVALID COMMAND	Memastikan bahwa program tidak menerima input diluar dari command-command yang sudah dibuat	1. Mencoba memasukkan input asal	TUBES ALSTRUK DAT;	Program menampilkan pesan "Command tidak diketahui!"	

## 7 Pembagian Kerja dalam Kelompok

Tabel 1 Daftar Pembagian Kerja Program

Nama Lengkap - NIM	Deskripsi Tugas
Muhammad Rafly- 18222067	<ul style="list-style-type: none"><li>- Merancang struktur dan organisasi file program</li><li>- Membuat adt mesin karakter, mesin input, mesin file</li><li>- Membuat adt kata dan detail lagu</li><li>- Membenarkan semua adt</li><li>- Membuat main</li><li>- Membuat function start</li><li>- Membuat function load</li><li>- Membuat funtion save</li><li>- Membuat functions status</li><li>- Membenarkan semua function</li><li>- Laporan log activity dan algoritma menarik</li></ul>
Dzulfaqor Ali Dipanegara - 18222017	<ul style="list-style-type: none"><li>- Mengimplementasikan adt set, stack, dan queue.</li><li>- Membuat function playlist create</li><li>- Membuat function playlist add song</li><li>- Membuat function playlist add album</li><li>- Membuat function playlist swap</li><li>- Membuat function playlist remove</li><li>- Membuat function playlist delete</li><li>- Mengerjakan laporan bagian struktur ADT</li><li>- Mengerjakan laporan bagian Program utama</li><li>- Melakukan unit test terhadap fungsi START, LOAD, DISPLAY, PLAYLIST( termasuk create</li></ul>

	hingga delete)
Yoga Putra Pratama - 18222073	<ul style="list-style-type: none"> <li>- Membuat ADT map</li> <li>- Membuat function queue song</li> <li>- Membuat function queue playlist</li> <li>- Membuat function queue swap</li> <li>- Membuat function queue remove</li> <li>- Membuat function queue clear</li> <li>- Mengerjakan laporan bagian Test Script</li> </ul>
Willhelmina Rachel Silalahi - 18222049	<ul style="list-style-type: none"> <li>- Membuat function Help</li> <li>- Membuat function List</li> <li>- Mengerjakan laporan</li> </ul>
Habib Akhmad Al Farisi - 18222029	<ul style="list-style-type: none"> <li>- Membuat ADT set</li> <li>- Membuat function Play Song</li> <li>- Membuat function Play Playlist</li> <li>- Membuat function Song Next</li> <li>- Membuat function Song Previous</li> <li>- Mengerjakan laporan bagian Data Test</li> </ul>

## 8 Lampiran

### 8.1 Deskripsi Tugas Besar

#### Spesifikasi Umum

Buatlah sebuah aplikasi simulasi berbasis CLI (command-line interface). Sistem ini dibuat dalam **bahasa C** dengan menggunakan **struktur data yang sudah kalian pelajari** di mata kuliah ini. Kalian boleh menggunakan (atau memodifikasi) struktur data yang sudah kalian buat untuk praktikum pada tugas besar ini. Daftar ADT yang wajib digunakan dapat dilihat pada bagian [Daftar ADT](#). Library yang boleh digunakan hanya **stdio.h**, **stdlib.h**, **time.h**, dan **math.h**

#### System Mechanic

##### 1. About the System

WayangWave merupakan sebuah aplikasi yang bisa mensimulasikan *service* pemutaran musik. WayangWave ini memiliki beberapa fitur utama, yaitu:

1. Memutar lagu
2. Menampilkan daftar lagu
3. Membuat dan menghapus *playlist*
4. Mengatur urutan dimainkannya lagu
5. Menampilkan status dari aplikasi

##### 2. Main Menu

Ketika program pertama kali dijalankan, WayangWave akan memperlihatkan main menu yang berisi *welcome page* dan beberapa command yaitu **START**, **LOAD**, dan juga **HELP**. Setelah itu, *main menu* akan menerima masukan berupa command yang akan dijelaskan pada bagian berikutnya.

##### 3. Command

Pemain dapat memasukkan command-command berikut:

###### a. START

START merupakan salah satu command yang dimasukkan pertama kali dalam WayangWave. Setelah menekan Enter, dibaca file konfigurasi default yang berisi daftar penyanyi serta album yang dimiliki.

```
>> START;
```

File konfigurasi aplikasi berhasil dibaca. WayangWave berhasil dijalankan.

**b. LOAD <filename>**

LOAD merupakan salah satu command yang dimasukkan pertama kali dalam WayangWave. Command ini memiliki satu argumen yaitu filename yang merepresentasikan suatu *save file* yang ingin dibuka. *File* didapatkan dari folder tertentu, contohnya *save*. Setelah menekan Enter, akan dibaca *save file* <filename> yang berisi list penyanyi, album, dan lagu yang bisa diputar. Lebih detailnya bisa dilihat pada [Konfigurasi Aplikasi](#).

```
>> LOAD filenihil.txt;
```

```
// Meload /save/filenihil.txt
```

Save file tidak ditemukan. WayangWave gagal dijalankan.

```
>> LOAD savefile.txt;
```

```
// Meload /save/savefile.txt
```

Save file berhasil dibaca. WayangWave berhasil dijalankan.

**c. LIST**

LIST merupakan command yang digunakan untuk menampilkan list playlist yang ada, list penyanyi, list album dari penyanyi, dan list lagu yang ada di album. Terdapat dua jenis list, **DEFAULT** dan **PLAYLIST**.

**1. LIST DEFAULT**

Command LIST DEFAULT digunakan untuk melihat list penyanyi yang ada. Selanjutnya dapat memilih untuk melihat album dari penyanyi yang dipilih. Kemudian melihat lagu yang ada dari album yang dipilih. Terdapat konfirmasi apakah ingin melihat album/lagu.

```
>> LIST DEFAULT;
```

Daftar Penyanyi :

1. New Jeans
2. BLACKPINK
3. Central Cee

4. Rich Brian
5. Denny Caknan
6. Via Vallen

Ingin melihat album yang ada?(Y/N): **Y;**

Pilih penyanyi untuk melihat album mereka: **BLACKPINK;**

Daftar Album oleh BLACKPINK :

1. BORN PINK
2. THE ALBUM
3. SQUARE ONE
4. SQUARE TWO

Ingin melihat lagu yang ada?(Y/N): **Y;**

Pilih album untuk melihat lagu yang ada di album : **SQUARE ONE;**

Daftar Lagu di SQUARE ONE:

1. WHISTLE
2. BOOMBAYAH

**>> LIST DEFAULT;**

Daftar Penyanyi :

1. New Jeans
2. BLACKPINK
3. Central Cee
4. Rich Brian
5. Denny Caknan
6. Via Vallen

Ingin melihat album yang ada?(Y/N): **N;**

## 2. LIST PLAYLIST

Command LIST PLAYLIST digunakan untuk menampilkan playlist yang ada pada pengguna.

**>> LIST PLAYLIST;**

Daftar playlist yang kamu miliki:

1. K-pop Enjoyer

2. Z E R O 4.0
3. Be the main character
4. Hey i liked you.

>> **LIST PLAYLIST;**

Daftar playlist yang kamu miliki:  
Kamu tidak memiliki playlist.

#### d. **PLAY**

PLAY merupakan command yang digunakan untuk memutar lagu atau playlist yang dipilih. Ketika command PLAY dieksekusi, queue yang ada dihapus ketika memainkan lagu atau digantikan oleh lagu dalam playlist ketika memainkan playlist. Terdapat dua jenis play, **SONG** dan **PLAYLIST**.

##### 1. **PLAY SONG**

Command PLAY SONG digunakan untuk memainkan lagu berdasarkan masukan nama penyanyi, nama album, dan id lagu. Ketika command ini berhasil dieksekusi, queue dan riwayat lagu akan menjadi kosong.

>> **PLAY SONG;**

Daftar Penyanyi :

1. New Jeans
2. BLACKPINK
3. Central Cee
4. Rich Brian
5. Denny Caknan
6. Via Vallen

Masukkan Nama Penyanyi yang dipilih : **BLACKPINK;**

Daftar Album oleh BLACKPINK :

1. BORN PINK
2. THE ALBUM
3. SQUARE ONE
4. SQUARE TWO

Masukkan Nama Album yang dipilih : **BORN PINK;**



Daftar Lagu Album BORN PINK oleh BLACKPINK :

1. Pink Venom
2. Shut Down
3. Typa Girl
4. Yeah Yeah Yeah
5. Hard to Love
6. The Happiest Girl
7. Tally
8. Ready for Love

Masukkan ID Lagu yang dipilih : **3**;

Memutar lagu "Type Girl" oleh "BLACKPINK".

## 2. PLAY PLAYLIST

Command **PLAY PLAYLIST** digunakan untuk memainkan lagu berdasarkan id playlist. Ketika command ini berhasil dieksekusi, current song akan menjadi lagu pada urutan pertama playlist dan queue akan berisi semua lagu yang ada dalam playlist yang akan dimainkan dan isi riwayat lagu sama dengan queue, tetapi dengan urutan yang di-*reverse*.

**>> PLAY PLAYLIST;**

Masukkan ID Playlist: **2**;

Memutar playlist "K-pop Enjoyer".

### e. QUEUE

QUEUE merupakan command yang digunakan untuk memanipulasi queue lagu. Command ini memiliki 5 tipe, yaitu **SONG**, **PLAYLIST**, **SWAP**, **REMOVE**, dan **CLEAR**.

#### 1. QUEUE SONG

Command **QUEUE SONG** digunakan untuk menambahkan lagu ke dalam queue. Command ini menerima input lagu berdasarkan nama penyanyi, nama album, dan id dari lagu yang ingin dimasukkan ke dalam queue.

**>> QUEUE SONG;**

Daftar Penyanyi :

1. New Jeans
2. BLACKPINK
3. Central Cee
4. Rich Brian
5. Denny Caknan
6. Via Vallen

Masukkan Nama Penyanyi: **BLACKPINK;**

Daftar Album oleh BLACKPINK :

1. BORN PINK
2. THE ALBUM
3. SQUARE ONE
4. SQUARE TWO

Masukkan Nama Album yang dipilih : **BORN PINK;**

Daftar Lagu Album BORN PINK oleh BLACKPINK :

1. Pink Venom
2. Shut Down
3. Typa Girl
4. Yeah Yeah Yeah
5. Hard to Love
6. The Happiest Girl
7. Tally
8. Ready for Love

Masukkan ID Lagu yang dipilih: **3;**

Berhasil menambahkan lagu "Type Girl" oleh "BLACKPINK" ke queue.

## 2. QUEUE PLAYLIST

Command QUEUE PLAYLIST digunakan untuk menambahkan lagu yang ada dalam playlist ke dalam queue. Command ini menerima input dari id playlist yang ingin dimasukkan ke dalam queue.

**>> QUEUE PLAYLIST;**

Masukkan ID Playlist: **2**;

Berhasil menambahkan playlist “K-pop Enjoyer” ke queue.

### 3. **QUEUE SWAP <x> <y>**

Command **QUEUE SWAP** digunakan untuk menukar lagu pada urutan ke **x** dan juga urutan ke **y**.

**>> QUEUE SWAP 1 2;**

Lagu “Kokoro wo Komete” berhasil ditukar dengan “Mirror”

**>> QUEUE SWAP 80 2;**

Lagu dengan urutan ke 80 tidak terdapat dalam queue!

### 4. **QUEUE REMOVE <id>**

Command **QUEUE REMOVE** digunakan untuk menghapus lagu dari queue. Command ini menerima input berupa urutan lagu (**id**) yang ingin dihapus dari queue.

**>> QUEUE REMOVE 2;**

Lagu “Kokoro wo Komete” oleh “Aoi Teshima” telah dihapus dari queue!

**>> QUEUE REMOVE 40;**

Lagu dengan urutan ke 40 tidak ada.

### 5. **QUEUE CLEAR**

Command **QUEUE CLEAR** digunakan untuk mengosongkan queue.

**>> QUEUE CLEAR;**

Queue berhasil dikosongkan.

f. **SONG**

SONG merupakan command yang digunakan untuk navigasi lagu yang ada pada queue lagu saat ini. Terdapat 2 tipe navigasi yaitu **NEXT** dan **PREVIOUS**.

1. **SONG NEXT**

Command SONG NEXT digunakan untuk memutar lagu yang berada di dalam queue. Lagu yang sedang diputar kemudian ditambah ke dalam daftar riwayat putar lagu. Jika queue kosong, yang diputar adalah lagu yang sedang diputar.

>> **SONG NEXT;**

Memutar lagu selanjutnya  
"Hype Boy" oleh "New Jeans"

>> **SONG NEXT;**

Queue kosong, memutar kembali lagu  
"Mirror" oleh "Yasuda Rei"

2. **SONG PREVIOUS**

Command SONG PREVIOUS digunakan untuk memutar lagu yang terakhir kali diputar. Lagu yang sedang diputar kemudian ditambah ke dalam queue dengan urutan pertama. Jika daftar riwayat lagu kosong, yang diputar adalah lagu yang sedang diputar.

>> **SONG PREVIOUS;**

Memutar lagu sebelumnya  
"Hype Boy" oleh "New Jeans"

>> **SONG PREVIOUS;**

Riwayat lagu kosong, memutar kembali lagu  
"Mirror" oleh "Yasuda Rei"

g. **PLAYLIST**

Command ini digunakan untuk melakukan basic command untuk playlist yaitu **CREATE**, **ADD**, **SWAP**, **REMOVE** dan **DELETE**.

## 1. PLAYLIST CREATE

Command **PLAYLIST CREATE** digunakan untuk membuat playlist baru dan ditambahkan pada daftar playlist pengguna. Keadaan awal playlist adalah kosong. Nama playlist dapat sama dengan playlist yang sudah ada.

**>> PLAYLIST CREATE;**

Masukkan nama playlist yang ingin dibuat : **K-pop Enjoyer;**

Playlist **K-pop Enjoyer** berhasil dibuat!

Silakan masukkan lagu - lagu artis terkini kesayangan Anda!

**>> PLAYLIST CREATE;**

Masukkan nama playlist yang ingin dibuat : ;

Minimal terdapat 3 karakter selain *whitespace* dalam nama playlist.  
Silakan coba lagi.

## 2. PLAYLIST ADD

Command **PLAYLIST ADD** digunakan untuk menambahkan lagu pada suatu playlist yang telah ada sebelumnya pada daftar playlist pengguna. Pada defaultnya, command ini hanya dapat menambahkan satu spesifik lagu atau semua lagu yang ada pada album kepada suatu *existing* playlist.

Apabila lagu atau lagu - lagu di dalam album yang ingin ditambahkan sudah ada di dalam suatu playlist pengguna maka lagu - lagu yang ditambahkan adalah yang belum ada di playlist pengguna.

Tampilkan pesan *error* apabila masukkan pengguna tidak valid pada setiap permintaan masukkan.

**>> PLAYLIST ADD SONG;**

Daftar Penyanyi :

1. New Jeans

2. BLACKPINK
3. Central Cee
4. Rich Brian
5. Denny Caknan
6. Via Vallen

Masukkan Nama Penyanyi yang dipilih : **BLACKPINK;**

Daftar Album oleh BLACKPINK :

1. BORN PINK
2. THE ALBUM
3. SQUARE ONE
4. SQUARE TWO

Masukkan Judul Album yang dipilih : **BORN PINK;**

Daftar Lagu Album BORN PINK oleh BLACKPINK :

1. Pink Venom
2. Shut Down
3. Typa Girl
4. Yeah Yeah Yeah
5. Hard to Love
6. The Happiest Girl
7. Tally
8. Ready for Love

Masukkan ID Lagu yang dipilih : **1;**

Daftar Playlist Pengguna :

1. RnB
2. Indo Songs
3. Dangdut Enjoyer
4. WIBU Songs
5. RANDOM SONGS
6. On Repeat Daily
7. For Night Ride
8. Romantic
9. Crying Session
10. K-Pop Enjoyer

Masukkan ID Playlist yang dipilih : **10;**

Lagu dengan judul “Pink Venom” pada album BORN PINK oleh penyanyi BLACKPINK berhasil ditambahkan ke dalam playlist K-Pop Enjoyer.

**>> PLAYLIST ADD ALBUM;**

Daftar Penyanyi :

1. New Jeans
2. BLACKPINK
3. Central Cee
4. Rich Brian
5. Denny Caknan
6. Via Vallen

Masukkan Nama Penyanyi yang dipilih : **BLACKPINK;**

Daftar Album oleh BLACKPINK :

1. BORN PINK
2. THE ALBUM
3. SQUARE ONE
4. SQUARE TWO

Masukkan Judul Album yang dipilih : **BORN PINK;**

Daftar Playlist Pengguna :

1. RnB
2. Indo Songs
3. Dangdut Enjoyer
4. WIBU Songs
5. RANDOM SONGS
6. On Repeat Daily
7. For Night Ride
8. Romantic
9. Crying Session
10. K-Pop Enjoyer

Masukkan ID Playlist yang dipilih : **10;**

Album dengan judul “BORN PINK” berhasil ditambahkan ke dalam pada playlist pengguna “K-Pop Enjoyer”.

**>> PLAYLIST ADD SONG;**

Daftar Penyanyi :

1. New Jeans
2. BLACKPINK
3. Central Cee
4. Rich Brian
5. Denny Caknan
6. Via Vallen

Masukkan Nama Penyanyi yang dipilih : **BLACKPIN;**

Penyanyi BLACKPIN tidak ada dalam daftar. Silakan coba lagi.

**>> PLAYLIST ADD ALBUM;**

Daftar Penyanyi :

1. New Jeans
2. BLACKPINK
3. Central Cee
4. Rich Brian
5. Denny Caknan
6. Via Vallen

Masukkan Nama Penyanyi yang dipilih : **BLACKPINK;**

Daftar Album oleh BLACKPINK :

1. BORN PINK
2. THE ALBUM
3. SQUARE ONE
4. SQUARE TWO

Masukkan Judul Album yang dipilih : **BORN PI;**

Album BORN PI tidak ada dalam daftar. Silakan coba lagi.

**3. PLAYLIST SWAP <id> <x> <y>**

Command PLAYLIST SWAP digunakan untuk menukar lagu pada urutan ke **x** dan juga urutan ke **y** di playlist dengan urutan ke **id**.



**>> PLAYLIST SWAP 1 1 2;**

Berhasil menukar lagu dengan nama “Blue Bird” dengan “Silhouette” di playlist “Naruto Vibes 🤖”

**>> PLAYLIST SWAP 500 1 2;**

Tidak ada playlist dengan playlist ID 500

**>> PLAYLIST SWAP 1 900 2;**

Tidak ada lagu dengan urutan 900 di playlist “Galau”

#### 4. **PLAYLIST REMOVE <id> <n>**

Command **PLAYLIST REMOVE** digunakan untuk menghapus lagu dengan urutan **n** pada playlist dengan index **id**.

**>> PLAYLIST REMOVE 2 4;**

Lagu “Ruang Rindu” oleh “Letto” telah dihapus dari playlist “Indo Songs”!

**>> PLAYLIST REMOVE 100 2;**

Tidak ada playlist dengan ID 100.

**>> PLAYLIST REMOVE 4 40;**

Tidak ada lagu dengan urutan 40 di playlist “WIBU Songs”!

#### 5. **PLAYLIST DELETE**

Command **PLAYLIST DELETE** digunakan untuk melakukan penghapusan suatu *existing* playlist dalam daftar playlist pengguna. Tampilkan pesan *error* apabila masukkan pengguna tidak valid pada setiap permintaan masukkan.

**>> PLAYLIST DELETE;**

Daftar Playlist Pengguna :

1. RnB
2. Indo Songs
3. Dangdut Enjoyer
4. WIBU Songs
5. RANDOM SONGS
6. On Repeat Daily
7. For Night Ride
8. Romantic
9. Crying Session
10. K-Pop Enjoyer

Masukkan ID Playlist yang dipilih : 10

Playlist ID 10 dengan judul “K-pop Enjoyer” berhasil dihapus.

#### >> **PLAYLIST DELETE;**

Daftar Playlist Pengguna :

1. RnB
2. Indo Songs
3. Dangdut Enjoyer
4. WIBU Songs
5. RANDOM SONGS
6. On Repeat Daily
7. For Night Ride
8. Romantic
9. Crying Session
10. K-Pop Enjoyer

Masukkan ID Playlist yang dipilih : 11

Tidak ada playlist dengan ID 11 dalam daftar playlist pengguna. Silakan coba lagi.

#### h. **STATUS**

STATUS merupakan command yang digunakan untuk menampilkan lagu yang sedang dimainkan beserta Queue song yang ada dan dari playlist mana lagu itu diputar.

>> **STATUS;**

// Tidak ada lagu yang diputar

Now Playing:

No songs have been played yet. Please search for a song to begin playback.

Queue:

Your queue is empty.

>> **STATUS;**

// Ada lagu, tapi queue kosong

Now Playing:

Coldplay - Up&Up - A Head Full of Dreams

Queue:

Your queue is empty.

>> **STATUS;**

// Ada lagu dan ada queue

Now Playing:

Coldplay - Up&Up - A Head Full of Dreams

Queue:

1. Arctic Monkeys - R U Mine? - AM
2. BLACKPINK - WHISTLE - SQUARE ONE

>> **STATUS;**

// Ada lagu, queue, dan playlist

// Hanya ditampilkan ketika melakukan playlist. Tidak ditampilkan jika terdapat lagu selain dari playlist

Current Playlist: Copium

Now Playing:

Coldplay - Up&Up - A Head Full of Dreams

Queue:

1. Arctic Monkeys - R U Mine? - AM
2. BLACKPINK - WHISTLE - SQUARE ONE

i. **SAVE <filename>**

SAVE merupakan command yang digunakan untuk menyimpan state aplikasi terbaru ke dalam suatu file. Command **SAVE** memiliki satu argumen yang merepresentasikan nama file yang akan disimpan. Penyimpanan dilakukan pada folder tertentu, misal folder save.

```
>> SAVE savefile.txt;
```

Save file berhasil disimpan.  
// File disimpan pada /save/savefile.txt

j. **QUIT**

QUIT merupakan command yang digunakan untuk keluar dari aplikasi WayangWave.

```
>> QUIT;
```

Apakah kamu ingin menyimpan data sesi sekarang? **N**;

Kamu keluar dari WayangWave.  
Dadah ^\_^/

```
>> QUIT;
```

Apakah kamu ingin menyimpan data sesi sekarang? **Y**;  
// memanggil save

k. **HELP**

HELP merupakan command yang digunakan menampilkan daftar command yang mungkin untuk dieksekusi dengan deskripsinya. Penjelasan dari deskripsi dibebaskan selama masih mendeskripsikan command sesuai dengan spek.

```
>> HELP; // ketika perintah dipanggil sebelum memasuki sesi
```

====[ Menu Help WayangWave ]=====

1. START -> Untuk masuk sesi baru

2. LOAD -> Untuk memulai sesi berdasarkan file konfigurasi

>> **HELP;** // ketika perintah dipanggil setelah memasuki sesi

====[ Menu Help WayangWave ]====

1. LIST -> Untuk menampilkan daftar lagu, playlist...

2. PLAY -> Untuk memulai suatu lagu

...

9. SAVE -> Untuk menyimpan state ke dalam file

10. QUIT -> Untuk keluar dari sesi

#### I. <INVALID COMMAND>

Command-command selain yang disebutkan di atas dinyatakan akan tidak valid dan hanya akan mengeluarkan teks error.

>> **TUBES ALSTRUKDAT;**

Command tidak diketahui!

// Mengeksekusi LOAD sesudah memasuki sesi

>> **LOAD filekonfigurasi.txt;**

Command tidak bisa dieksekusi!

// Mengeksekusi LIST sebelum memasuki sesi

>> **LIST DEFAULT;**

Command tidak bisa dieksekusi!

## 8.2 Notulen Rapat

### Form Asistensi Tugas Besar



IF2111/Algoritma dan Struktur Data STI





Sem. 1 2023/2024

No. Kelompok /Kelas : Kelompok 7/ Kelas K-01  
Nama Kelompok : ASD GACOR  
Anggota Kelompok (Nama/NIM) :  
1. Muhammad Rafly/18222067  
2. Dzulfaqor Ali D/ 18222017  
3. Willhelmina Rachel S/ 18222049  
4. Yoga Putra Pratama/ 18222073  
5. Habib Akhmad Al Farisi/ 18222029

Asisten Pembimbing : M Syahrul Surya Putra






#### Asistensi I

<b>Tanggal : 30 Oktober 2023</b>	<b>Catatan Asistensi:</b>  Catatan tambahan untuk spesifikasi: <ul style="list-style-type: none"><li>• Bikin simulasi aplikasi (SPOTIFY)</li><li>• ADT list sesuai spesifikasi dan kebutuhan masing2</li><li>• LOAD sama SAVE satu folder yang sudah predetermined</li><li>• Config file disave sesuai urutan</li><li>• Playlist baru bisa pake nama (Ada validnya)</li><li>• Check masing-masing case (queue kosong, gaada yang main, dll)</li></ul> Pertanyaan + Jawaban: <ul style="list-style-type: none"><li>• Untuk previous song, currently played song itu dimasukkan queue dari belakang atau depan? Dari depan (jadi index pertama lagi)</li><li>• Max Capacity? Soft limit = 100</li><li>• Cara biar primitif ADT ga define ulang? Ganti nama</li><li>• Cara rapih?</li></ul>
<b>Tempat : Zoom Meeting</b>	
<b>Kehadiran Anggota Kelompok:</b>	
No NIM Tanda tangan	
1 18222067 	
2 18222017 	
3	


<p>18222049</p>  <p>4</p> <p>18222073</p>  <p>5</p> <p>18222029</p> 	<p>Ikutin yang di spesifikasi sebagai panduan (ga harus persis sama) + harus ada driver untuk setiap ADT list.</p> <ul style="list-style-type: none"> <li>Mesin kata dan karakter sebagai parsing argumen kurang paham Input gapake scanf, tapi lewat mesin karakter</li> </ul>
	<p><b>Tanda Tangan Asisten:</b></p> 

## Asistensi II

<b>Tanggal : 15 November 2023</b>	<p>Pertanyaan + Jawaban:</p> <ol style="list-style-type: none"> <li>1. Makefile nya apakah sudah bener? Aman</li> <li>2. Invalid Command untuk play song test itu masuknya command tidak diketahui</li> <li>3. Tidak boleh membuat primitive illegal</li> </ol>
<b>Tempat : Zoom Meeting</b>	

<p><b>Kehadiran Anggota Kelompok:</b></p> <p>No</p> <p>NIM</p> <p>Tanda tangan</p> <p>1</p> <p>18222067</p>  <p>2</p> <p>18222017</p>  <p>3</p> <p>18222049</p>  <p>4</p> <p>18222073</p>  <p>5</p> <p>18222029</p> 	<p>4. Adt disesuaikan dengan kebutuhan tidak terpaku adt pada praktikum</p> <p>5. Penilaian kerapian : tahu letak source file, boleh pake comment untuk kerapian</p> <p>6. Yg penting jalan di linux, handle windows entaran</p> <p>7. Command tanpa enter, misal playlist create; bukan playlist create n;</p> <p>8. Jangan pake unistd</p> <p>9. input nama save bebas, gpp ada 0000</p>
	<p><b>Tanda Tangan Asisten:</b></p>



	
--	--

### 8.3 Log Activity Anggota Kelompok

27/10/2023

- Membuat grup LINE (Semua)

30/10/2023

- Zoom asistensi 1 (Semua)
- Pembagian tugas pembuatan ADT (Semua)
- Pembuatan git repository (Rafly)
- Desain sistem file (Rafly)

31/10/2023

- ADT queue (Dzulfaqor)
- ADT mesin karakter dan mesin kata (Rafly)
- Testing read file dan input (Rafly)

2/11/2023

- ADT stack (Dzulfaqor)
- ADT set (Faris)
- ADT map (Yoga)
- Testing read file dan input (Rafly)

3/11/2023

- Memperbaiki ADT set, queue, dan set (Dzulfaqor)
- Testing read file dan input (Rafly)

6/11/2023

- Pembagian tugas pembuatan command (Semua)

7/11/2023

- Memperbaiki dan melengkapi semua ADT yang ada di spesifikasi (Rafly)
- Memisah ADT mesin kata menjadi mesin file dan mesin input (Rafly)
- Berhasil input (Rafly)

8/11/2023

- Membuat ADT detail (Rafly)
- Membuat primitif display untuk semua ADT (Rafly)
- Memisah ADT mesin kata menjadi mesin file dan mesin input (Rafly)
- Membuat data.c dan console.c (Rafly)

9/11/2023

- Membuat command Playlist (Dzulfaqor)

11/11/2023

- Membenarkan ADT list dinamik dan struktur berkait (Rafly)
- Membenarkan command Playlist (Rafly)
- Kata.c (Rafly)

12/11/2023

- Debugging (Rafly dan Dzulfaqor)
- Membuat command Play dan Song (Faris)

13/11/2023

- Debugging input (Rafly)

14/11/2023

- Readme dan MakeFile (Rafly)
- Debugging (Rafly)
- Membuat command Queue (Yoga)

15/11/2023

- Membenarkan command Queue (Rafly)
- Debugging (Dzulfaqor)

20/11/2023

- Membuat command Help dan List (Will)
- Membuat command Status (Rafly)
- Membenarkan command Help dan List (Rafly)

23/11/2023

- Membuat command Enhance (Rafly)

24/11/2023

- Finalisasi (Semua)