

**LAPORAN TUGAS BESAR**  
**IF2111 Algoritma dan Struktur Data STI**

**“WayangWave”**

Dipersiapkan oleh:

Kelompok 6

Gracya Tio Damena Sidabutar / 18222110


Moh Afnan Fawaz / 18222111

Gabriel Marcellino Joestiawan / 18222115

Jason Jahja / 18222116

Hanan Fitra Salam / 18222133

Sekolah Teknik Elektro dan Informatika – Institut Teknologi Bandung  
Jl. Ganesha 10, Bandung 40132

	<b>Sekolah Teknik Elektro dan Informatika ITB</b>	<b>Nomor Dokumen</b>		<b>Halaman</b>
		<i>IF2111-TB-K3-6</i>		<i>&lt;jml hlm&gt;</i> <b>&lt;40</b>
		<i>Revisi</i>	<i>&lt;no revisi&gt;</i> <b>1</b>	<i>&lt;Tgl release&gt;</i> <b>22 November 2023</b>

## Daftar Isi

1.	Ringkasan .....	4
2.	Penjelasan Tambahan Spesifikasi Tugas .....	4
2.1	Spesifikasi Fitur Tambahan Enhance.....	4
2.2	Spesifikasi Fitur Tambahan 1 .....	Error! Bookmark not defined.
3.	Struktur Data (ADT) .....	4
3.1	ADT List .....	4
3.2	ADT Map .....	5
3.3	ADT Set.....	5
3.4	ADT Queue .....	5
3.5	ADT Stack.....	6
3.6	ADT Mesin Kata .....	6
3.7	ADT Mesin Baris.....	7
3.8	ADT Mesin Karakter.....	7
3.9	ADT List Dinamis .....	7
3.10	ADT List Berkait.....	8
3.11	AD .....	Error! Bookmark not defined.
3.12	AAD.....	Error! Bookmark not defined.
3.13	ADA.....	Error! Bookmark not defined.
3.14	D.....	Error! Bookmark not defined.
3.15	HJB.....	Error! Bookmark not defined.
4.	Program Utama.....	8
5.	Algoritma-Algoritma Menarik .....	9
5.1	Load dan Save .....	9
6.	Data Test .....	10
6.1	INISIALISASI (WELCOME PAGE).....	10
6.2	START.....	10
6.3	LOAD.....	10
6.4	HELP.....	10
6.5	LIST DEFAULT .....	11
6.6	LIST PLAYLIST .....	12
6.7	PLAY SONG.....	12
6.8	PLAY PLAYLIST.....	13

6.9	QUEUE SONG .....	13
6.10	QUEUE PLAYLIST .....	14
6.11	QUEUE SWAP <> <> .....	14
6.12	QUEUE REMOVE <> .....	14
6.13	QUEUE CLEAR .....	15
6.14	SONG NEXT .....	15
6.15	SONG PREVIOUS.....	15
6.16	PLAYLIST CREATE.....	16
6.17	PLAYLIST ADD SONG .....	16
6.18	PLAYLIST ADD ALBUM .....	16
6.19	PLAYLIST SWAP <> <> <>.....	17
6.20	PLAYLIST REMOVE <> <> .....	17
6.21	PLAYLIST DELETE.....	17
6.22	STATUS.....	17
6.23	SAVE .....	18
6.24	QUIT .....	18
6.25	ENHANCE.....	18
6.26	INVALID COMMAND.....	18
7.	Text Script.....	19
8.	Pembagian Kerja dalam Kelompok .....	23
9.	Lampiran .....	24
9.1	Deskripsi Tugas Besar.....	24
9.2	Notulensi Asistensi.....	24
9.3	Notulensi Rapat .....	27
9.4	Log Activity Kelompok.....	27
9.5	Lain.....	Error! Bookmark not defined.
9.6	Lain.....	Error! Bookmark not defined.
9.7	Lain.....	Error! Bookmark not defined.

## 1. Ringkasan

WayangWave merupakan aplikasi simulasi yang dibuat untuk menggantikan perangkat lunak di *walkman* Roro. Aplikasi ini dibuat untuk Bondowoso gunakan dalam melunakkan hati Roro. Roro yang selama ini hanya senang mendengarkan satu genre lagu dari walkman-nya membuat Bondowo menyayangkan hal tersebut. Aplikasi ini diharapkan bisa memutar lebih banyak macam lagu dan membuat Roro luluh pada Bondowoso dan selera musiknya.

Dari deskripsi permasalahan tersebut kami membuat aplikasi simulasi berbasis CLI (*command-line interface*) yang memiliki 5 fitur utama yaitu memutar lagu, menampilkan daftar lagu, membuat dan menghapus *playlist*, membuat antrian lagu, dan menampilkan status dari aplikasi.

Laporan mencakup deskripsi umum persoalan aplikasi, penjelasan tambahan spesifikasi tugas, penjelasan struktur data-struktur data yang digunakan, penjelasan program utama, data-data dan script yang digunakan dalam pengujian aplikasi, dan pembagian tugas serta notulensi asistensi/rapat dan lampiran lainnya.

Program ini dibuat dengan menggunakan bahasa C dan memanfaatkan ADT yang telah dipelajari pada mata kuliah IF2111 Algoritma Struktur Data STI. Aplikasi ini akan dimulai dengan menampilkan menu utama yang dapat dipilih *user* untuk melakukan *start* untuk memulai aplikasi, *load* untuk membaca dan menjalankan aplikasi dengan *file* berisikan data yang sudah disimpan sebelumnya, atau *help* untuk menampilkan informasi kegunaan tiap *command* yang dapat digunakan *user* dalam menjalankan aplikasi ini. Setelah aplikasi dimulai, *user* dapat mengakses berbagai fitur (dengan *command*) yang ada pada aplikasi. *Command* yang valid pada aplikasi ini adalah *list default*, *list playlist*, *play song*, *play playlist*, *queue song*, *queue playlist*, *queue swap* *<x>* *<y>*, *queue remove* *<id>*, *queue clear*, *song next*, *song previous*, *playlist create*, *playlist add*, *playlist swap*, *playlist remove* *<id>* *<n>*, *playlist delete*, *status*, *save*, dan *quit*.

## 2. Penjelasan Tambahan Spesifikasi Tugas

### 2.1 Spesifikasi Fitur Tambahan Enhance

Fitur tambahan "ENHANCE" pada program ini memungkinkan pengguna untuk meningkatkan pengalaman mendengarkan musik dengan menambahkan lagu-lagu rekomendasi ke dalam playlist yang dipilih. Pengguna diminta untuk memasukkan id playlist yang ingin di-enhance, setelah itu program akan menampilkan daftar lagu-lagu rekomendasi random dari berbagai penyanyi dan album. Proses pemilihan lagu dilakukan dengan memilih secara acak, memberikan sentuhan kejutan dan variasi dalam playlist. Fitur ini memberikan pengguna kesempatan untuk mengeksplorasi lebih banyak musik dan membuat pengalaman mendengarkan musik menjadi lebih dinamis dan menyenangkan.

## 3. Struktur Data (ADT)

### 3.1 ADT List

Sketsa struktur data:

STEI- ITB	IF2111-TB-K3-6	Halaman 4 dari 28 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.		

```
typedef struct {
    Penyanyi listpenyanyi[MaxElList];
} ListofPenyanyi;

typedef struct {
    Word namaPenyanyi;
    MapofAlbum album;
} Penyanyi;
```

Gambar 3.1 Sketsa struktur data ADT List

ListofPenyanyi ini berupa list statis yang digunakan untuk menyimpan informasi penyanyi-penyanyi beserta album yang dimilikinya dan juga lagu-lagu dalam album tersebut. Kami memilih ADT ini karena dapat menyimpan informasi lebih dari satu penyanyi. ADT ini diimplementasikan sebagai ADT list dengan nama *file header* “list.h”.

### 3.2 ADT Map

Sketsa struktur data:

```
typedef struct
{
    Album listalbum[MaxElMap];
} MapofAlbum;

typedef struct
{
    Word namaAlbum;
    SetofSong listLagu;
} Album;
```

Gambar 3.2 Sketsa struktur data ADT Map

MapofAlbum ini menggunakan ADT Map yang digunakan untuk menyimpan album-album yang dimiliki seorang penyanyi beserta informasi lagu-lagu yang ada didalam tiap album tersebut. ADT ini dipilih karena dapat menyimpan key yang berupa nama dari album yang dimiliki penyanyi serta value dari key tersebut yang berisi SetofSong dimana SetofSong berisikan informasi lagu-lagu yang ada didalam album tersebut.

### 3.3 ADT Set

Sketsa struktur data:

```
typedef struct
{
    Song Lagu[MaxElSet];
} SetofSong;

typedef struct
{
    Word judul;
    Word album;
    Word penyanyi;
    int status;
} Song;
```

Gambar 3.3 Sketsa struktur data ADT Set

SetofSong menggunakan ADT Set yang digunakan untuk merepresentasikan lagu-lagu yang terdapat didalam suatu album. ADT ini dipilih agar memastikan bahwa dalam satu album tidak terdapat judul lagu yang sama dan juga bisa menyimpan informasi-informasi semua lagu yang ada di album. ADT ini diimplementasikan sebagai ADT Set dengan nama *file header* “set.h”.

### 3.4 ADT Queue

Sketsa struktur data:

```
typedef struct {
    Word song[IDX_MAXQueue+1];
    Word album[IDX_MAXQueue+1];
    Word singer[IDX_MAXQueue+1];
    int idxHead;
    int idxTail;
} Queue;
```

Gambar 3.4 Sketsa struktur data ADT Queue

Queue digunakan untuk menyelesaikan persoalan mengenai antrian lagu. Queue menyimpan informasi lagu-lagu yang user masukkan kedalam antrian seperti informasi judul lagu, nama album dari lagu tersebut, serta nama penyanyi dari lagu tersebut. ADT Queue ini dipilih karena bisa menyimpan elemen (informasi lagu) yang berurutan sesuai penyisipannya dengan konsep FIFO (*First In First Out*). ADT ini diimplementasikan sebagai ADT Queue dengan nama *file header* “queue.h”.

### 3.5 ADT Stack

Sketsa struktur data:

```
typedef struct {
    Word song[MaxEl];
    Word album[MaxEl];
    Word singer[MaxEl];
    int TOP;
} songHistory;
```

Gambar 3.5 Sketsa struktur data ADT Stack

SongHistory yang menggunakan konsep ADT Stack ini digunakan untuk menyelesaikan persoalan mengenai riwayat lagu. songHistory ini akan menyimpan lagu-lagu yang telah diputarkan oleh user. ADT Stack ini dipilih karena konsep FILO(*First In Last Out*) ini sangat berkesinambungan dengan konsep riwayat lagu dan fitur Song Previous yang nanti ada dalam aplikasi. ADT ini diimplementasikan sebagai ADT Stack dengan nama *file header* “stack.h”.

### 3.6 ADT Mesin Kata

Sketsa struktur data:

```
typedef struct
{
    char TabWord[NMax];
    int Length;
} Word;
```

Gambar 3.6 Sketsa struktur data ADT Mesin Kata

Word yang menggunakan ADT dari Mesin Kata ini berguna untuk menyelesaikan permasalahan mengenai keberadaan *string* pada bahasa pemrograman C. Dengan membuat Word ini memiliki sebuah *array of char*, Word dapat menggantikan peran *string* dan dapat dimanfaatkan untuk penulisan kata pada program. Word juga memiliki *length* yang menyimpan informasi mengenai panjang huruf dari suatu “kata”. ADT ini

diimplementasikan sebagai ADT Mesin Kata dengan nama *file header* “mesinkata.h”

### 3.7 ADT Mesin Baris

Sketsa struktur data:

```
typedef struct
{
    char kalimat[Nmax];
    int lengthsentence;
} Sentence;
```

Gambar 3.7 Sketsa struktur data ADT Mesin Baris

*Sentence* ini berguna sebagai tingkatan lanjutan dari *Word* sebagai representasi dari sebuah kalimat. Walaupun memiliki struktur yang sama dengan *Word*, namun *Sentence* memiliki fungsi yang berbeda dengan *Word*. *Sentence* berguna untuk menyimpan hasil bacaan dari suatu file sehingga dapat dibaca secara satu *line* penuh, yang mana dapat terdiri dari beberapa kata. ADT ini menggunakan konsep dasar dari ADT Mesin Kata namun dengan pengembangan terhadap *Mark* yang terdapat pada ADT Mesin Kata dengan nama *file header* “mesinbaris.h”.

### 3.8 ADT Mesin Karakter

```
#define MARK '\r'
/* State Mesin */
extern char currentChar;
extern boolean EOP;
static int retval;
```

Gambar 3.10 State mesin ADT Mesin Karakter

ADT Mesin Karakter tidak memiliki struktur khusus karena pada dasarnya struktur ini hanya merepresentasikan sebuah karakter. Variabel *currentChar* merupakan struktur dari ADT Mesin Karakter ini yang bertipe data *char*. ADT Mesin Kata berperan sebagai dasar landasan ADT Mesin Kata maupun ADT Mesin Baris. ADT ini memiliki fungsi untuk menginisiasi dalam membaca suatu file, memeriksa setiap huruf, dan membantu dalam meng-copy suatu *Word* maupun *Sentence*. ADT ini memiliki nama *file header* “mesinkata.h”

### 3.9 ADT List Dinamis

Sketsa struktur data:

```
#define Sizelist 10
typedef struct
{
    Word *namaPlaylist;
    ListBerkait *list;
    int nEff;
    int Capacity;
} ListofPlaylist;
```

Gambar 3.9 Sketsa struktur data ADT List Dinamis

STEI- ITB	IF2111-TB-K3-6	Halaman 7 dari 28 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.		

*ListofPlaylist* berfungsi dalam menyelesaikan persoalan mengenai penyimpanan *playlist* yang dimiliki oleh user. *List* ini menggunakan konsep ADT List Dinamis agar dapat menjelaskan bahwa setiap user bisa tidak memiliki *playlist* dan bisa memiliki *playlist*. Struktur *ListofPlaylist* terdiri dari 4 tipe data, yaitu *namaPlaylist* yang menyediakan informasi mengenai nama *playlist*, *list* sebagai penanda atau indeks tiap-tiap *playlist*, *nEff* untuk memberitahukan informasi banyaknya *playlist* yang ada, dan *Capacity* sebagai total maximum *playlist* dapat ditampung. *ListofPlaylist* memiliki nama *file header* “listdin.h”

### 3.10 ADT List Berkait

Sketsa struktur data:

```
typedef struct tElmtlist *address;

typedef struct {
    Song Lagu;
}infoLagu;
// typedef infoLagu infotype;

typedef struct tElmtlist {
    infoLagu info;
    address next;
} ElmtList;

// typedef infoLagu infotype;
typedef struct {
    address First;
} ListBerkait;
```

Gambar 3.10 Sketsa struktur data ADT List Berkait

*ListBerkait*, *ElmtList*, dan *infoLagu* merupakan tiga struktur yang saling berkesinambungan. *ListBerkait* merupakan inisiasi awal agar dapat mengakses “node” pertama atau lagu pertama dari sebuah *playlist*. Pada struct *ElmtList*, terdapat *infoLagu* yang terdiri dari satu tipe data *Song* yang menyimpan data dari sebuah lagu. ADT ini berguna untuk menuntaskan permasalahan mengenai *playlist* yang dimiliki oleh user, mulai dari membuat *playlist* maupun memodifikasi isi dari *playlist* tersebut. Dengan menggunakan konsep ini, lagu yang disimpan di dalam *playlist* user dapat bersifat bebas, yang berarti dapat menyimpan berbagai lagu walaupun dari album maupun penyanyi yang berbeda-beda. ADT List Berkait ini memiliki nama *file header* “listberkait.h”

## 4. Program Utama

Program utama WayangWave akan menampilkan sebuah *welcome page* dan beberapa *command* awal yang dapat diakses *user* yaitu START, LOAD, dan HELP. *Command* START adalah *command* yang bisa digunakan *user* pertama kali. Jika *user* pada *welcome page* memilih *command* START maka program akan membaca *file default* konfigurasi. Lalu, *Command* LOAD adalah *command* yang bisa digunakan *user* ketika pernah menyimpan sebuah progress aktivitas aplikasi WayangWave. Sedikit berbeda isi dari *file default* konfigurasi, tapi fungsi ini berguna untuk membaca *file* teks juga. Sedangkan *command* HELP digunakan oleh *user* untuk melihat ada apa saja fungsi yang tersedia pada aplikasi WayangWave. *Command* yang dapat dipanggil pada WayangWave setelah *user* telah memanggil *command* START atau LOAD adalah sebagai berikut:

STEI- ITB	IF2111-TB-K3-6	Halaman 8 dari 28 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.		



- a. LIST DEFAULT
- b. LIST PLAYLIST
- c. PLAY SONG
- d. PLAY PLAYLIST
- e. QUEUE ADD
- f. QUEUE PLAYLIST
- g. QUEUE SWAP
- h. QUEUE REMOVE
- i. QUEUE CLEAR
- j. SONG NEXT
- k. SONG PREVIOUS
- l. PLAYLIST CREATE
- m. PLAYLIST ADD
- n. PLAYLIST SWAP
- o. PLAYLIST REMOVE
- p. PLAYLIST DELETE
- q. STATUS
- r. SAVE
- s. QUIT

Beberapa *command* yang dapat dipanggil pada *welcome page*, *command* START dan LOAD, tidak bisa digunakan kembali jika sudah memanggil salah satu diantara kedua *command* tersebut. Selain dari kedua *command* sebelumnya dapat digunakan lebih dari satu kali. Setelah *user* menggunakan berbagai *command* yang tertera pada WayangWave, *user* dapat memilih untuk mengakhiri aktivitas pada aplikasi WayangWave. *Command* yang bisa dipilih untuk menutup program WayangWave adalah SAVE dan QUIT. *Command* SAVE digunakan untuk menyimpan perubahan yang sudah dilakukan pada aplikasi WayangWave. *Command* ini akan menyimpan *state* aplikasi terbaru ke dalam suatu *file*. Sedangkan *command* QUIT digunakan untuk keluar dari aplikasi WayangWave tanpa menyimpan perubahan yang telah dilakukan. Dalam menjalankan beberapa *command* diatas, dapat terjadi kasus dimana pengguna memasukan inputan yang salah, memilih *command* yang tidak tertera pada program, dll. Hal tersebut dapat membuat program mengeluarkan pesan berupa INVALID COMMAND yang mengartikan bahwa terdapat sesuatu ketidaktepatan *input* yang user lakukan.

## 5. Algoritma-Algoritma Menarik

### 5.1 Load dan Save

Algoritma menarik pertama yang kami temukan yaitu, keterkaitan antara fungsi *save* dan *load*. Karena keterkaitan kedua fungsi tersebut membuat kami merancang suatu program yang bisa membaca suatu *input-an* yaitu nama *file* dari fungsi *save* sehingga nantinya dapat digunakan juga pada fungsi *load*. Tantangan yang dihadapi adalah setiap kali kami membuat *file save* yang baru, nama *file* akan berubah-ubah sesuai keinginan *user* yang akan menyimpan perubahan terakhirnya. Dari situ dapat dikatakan bahwa sistem ini sistem yang dinamis, tidak bisa ditentukan di awal. Menurut kami algoritma ini menarik karena cukup menantang.

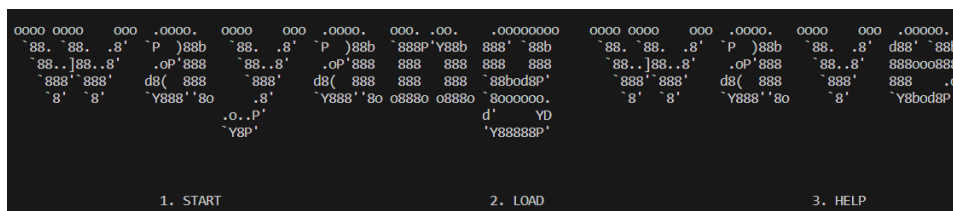
STEI- ITB	IF2111-TB-K3-6	Halaman 9 dari 28 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

## 6. Data Test

### 6.1 INISIALISASI (WELCOME PAGE)

Test ini dilakukan untuk memastikan bahwa program sudah dapat berjalan dan menampilkan tampilan judul dari program yaitu WayangWave serta main menu. Cara melakukan kompilasi program adalah pada path C:...\\src menggunakan perintah gcc -o main Spesifikasi/Start/Start.c Spesifikasi/Help/Help.c Spesifikasi/List/List.c Spesifikasi/Play/Play.c Spesifikasi/Queue/Queue.c Spesifikasi/Save/Save.c Spesifikasi/Song/Song.c Spesifikasi/Load/Load.c ADT/list/list.c ADT/listdin/listdin.c ADT/map/map.c ADT/mesinbaris/mesinbaris.c ADT/mesinkata/mesinkata.c ADT/mesinkarakter/mesinkarakter.c ADT/queue/queue.c ADT/set/set.c ADT/stack/stack.c ADT/strukturberkai/listberkai.c main.c Spesifikasi/Status/status.c Spesifikasi/Inisialisasi/Inisialisasi.c Spesifikasi/Playlist/Playlist.c

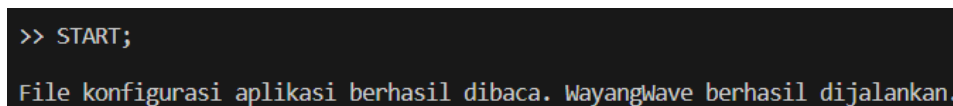
Lalu mengetikkan ./main pada command prompt atau terminal lainnya.



Gambar 6. 1. Welcome Page WayangWave

### 6.2 START

Test ini dilakukan untuk memastikan input “START” pada main menu sudah berjalan dengan sesuai.



Gambar 6. 2. START berhasil dijalankan

### 6.3 LOAD

Test ini dilakukan untuk memastikan input “LOAD” pada main menu sudah berjalan dengan sesuai. Tampilan dibawah adalah tampilan jika terdapat file yang sudah pernah disimpan sebelumnya. Jika nama file tidak sesuai dengan nama file yang tersedia, maka akan load akan gagal.

```
>> LOAD test.txt;  
Save file berhasil dibaca. WayangWave berhasil dijalankan.
```

Gambar 6.3 LOAD berhasil dijalankan

### 6.4 HELP

Test ini dilakukan untuk memastikan input “HELP” pada main menu sudah berjalan dengan sesuai.

STEI- ITB	IF2111-TB-K3-6	Halaman 10 dari 28 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.		

```
>> HELP;

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

1. START -> Untuk masuk sesi baru
2. LOAD <filename> -> Untuk memulai sesi berdasarkan file konfigurasi

=====
```

Gambar 6.4.1. Tampilan HELP pada Welcome Page

Sedangkan yang dibawah ini adalah tampilan HELP ketika sudah memilih command START atau LOAD.

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

==== [ Menu Help WayangWave ] =====
1a. LIST DEFAULT -> Untuk melihat list penyanyi, album, dan lagu yang ada
1b. LIST PLAYLIST -> Untuk menampilkan playlist yang ada pada pengguna
2a. PLAY SONG -> Untuk memainkan lagu berdasarkan masukan nama penyanyi, nama album, dan id lagu
2b. PLAY PLAYLIST -> Untuk memainkan lagu berdasarkan id playlist
3a. QUEUE SONG -> Untuk menambahkan lagu ke dalam queue
3b. QUEUE PLAYLIST -> Untuk menambahkan lagu yang ada dalam playlist ke dalam queue
3c. QUEUE SWAP <x> <y> -> Untuk menukar lagu pada urutan ke x dan juga urutan ke y
3d. QUEUE REMOVE <id> -> Untuk menghapus lagu dari queue
3e. QUEUE CLEAR -> Untuk mengosongkan queue
4a. SONG NEXT -> Untuk memutar lagu yang berada di dalam queue
4b. SONG PREVIOUS -> Untuk memutar lagu yang terakhir kali diputar
5a. PLAYLIST CREATE -> Untuk membuat playlist baru dan ditambahkan pada daftar playlist pengguna
5b. PLAYLIST ADD ALBUM -> Untuk menambahkan semua lagu pada Album ke suatu playlist yang ada pada daftar playlist pengguna
5c. PLAYLIST ADD SONG -> Untuk menambahkan lagu ke suatu playlist yang telah ada sebelumnya pada daftar playlist pengguna
5d. PLAYLIST SWAP <x> <y> -> Untuk menukar lagu pada urutan ke x dan juga urutan ke y di playlist dengan urutan ke id
5e. PLAYLIST REMOVE <id> <n> -> Untuk menghapus lagu dengan urutan n pada playlist dengan index id
5f. PLAYLIST DELETE -> Untuk melakukan penghapusan suatu existing playlist dalam daftar playlist pengguna
6. STATUS -> Untuk menampilkan lagu yang sedang dimainkan beserta Queue song yang ada dan dari playlist mana lagu itu diputar
7. SAVE <filename> -> Untuk menyimpan state ke dalam file
8. QUIT -> Untuk keluar dari sesi
```

Gambar 6.4.2 Tampilan HELP setelah Welcome Page

## 6.5 LIST DEFAULT

Test ini dilakukan untuk memastikan atau menguji command “LIST DEFAULT” sudah berjalan dengan sesuai. Command ini menampilkan Daftar Penyanyi yang ada (sesuai data pada file konfigurasi). Lalu akan meminta inputan apakah *user* ingin melihat album dan meminta untuk menginput artis yang albumnya ingin dilihat. Setelah itu *user* juga ditanyakan apakah ingin melihat lagu yang ada pada album-album tersebut dan meminta inputan nama album yang lagu-lagunya ingin ditampilkan. Hasil akhir tampilannya adalah seperti dibawah ini.

```

>> 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 di BORN PINK:
  1. Pink Venom
  2. Shut Down
  3. Typa Girl
  4. Ready For Love

```

Gambar 6.5 Tampilan LIST DEFAULT berhasil

## 6.6 LIST PLAYLIST

Test ini dilakukan untuk memastikan atau menguji command “LIST PLAYLIST” sudah berjalan dengan sesuai. Command ini menampilkan nama-nama playlist yang dimiliki *user*. Apabila *user* belum memiliki playlist akan keluar seperti gambar 6.6.1 sedangkan jika *user* sudah memiliki playlist akan keluar seperti gambar 6.1.2.

```

>> LIST PLAYLIST;
Daftar playlist yang kamu miliki:
Kamu tidak memiliki playlist.

```

Gambar 6.6.1 Tampilan LIST PLAYLIST ketika lisy playlist kosong

```

>> LIST PLAYLIST;
Daftar playlist yang kamu miliki:
  1. K-Pop
  2. Chillin'

```

Gambar 6.6.2 Tampilan LIST PLAYLIST ketika *user* memiliki playlist

## 6.7 PLAY SONG

Test ini dilakukan untuk memastikan atau menguji command “PLAY SONG” sudah berjalan dengan sesuai. Command ini akan memainkan lagu yang dipilih *user* melalui tahap pemilihan penyanyi, lalu album, dan ID Lagunya. Tampilan akhirnya adalah seperti pada gambar dibawah.

```

>> PLAY SONG;

Daftar Penyanyi:
1. BLACKPINK
2. Arctic Monkeys

Masukkan nama penyanyi yang dipilih: BLACKPINK;

Daftar Album:
1. BORN PINK
2. THE ALBUM

Masukkan nama album yang dipilih: BORN PINK;

Daftar Lagu:
1. Pink Venom
2. Shut Down
3. Typa Girl
4. Ready For Love

Masukkan ID lagu yang dipilih: Pink Venom;
ID Lagu tidak terdaftar!
Masukkan ID lagu yang dipilih: 1
;
Memutar lagu "Pink Venom" oleh "BLACKPINK"

```

Gambar 6.7 Tampilan PLAY SONG ketika berhasil

## 6.8 PLAY PLAYLIST

Test ini dilakukan untuk memastikan atau menguji command “PLAY PLAYLIST” sudah berjalan dengan sesuai. Command ini akan memainkan playlist sesuai pilihan playlist yang ada. Tampilan akhirnya akan seperti pada gambar 6.8.1. Apabila *user* belum memiliki playlist, maka tampilan akan seperti gambar 6.8.2.

```

>> PLAY PLAYLIST;
Daftar Playlist:
1. K-Pop
2. Chillin'

Masukkan ID Playlist: 1;

Memutar playlist "K-Pop".

```

Gambar 6.8.1 Tampilan PLAY PLAYLIST ketika berhasil

Gambar 6.8.2 Tampilan PLAY PLAYLIST ketika list playlist kosong

## 6.9 QUEUE SONG

Test ini dilakukan untuk memastikan atau menguji command “QUEUE SONG” sudah berjalan dengan sesuai. Command ini memasukkan lagu kedalam antrian. User akan diminta untuk memasukkan pilihan penyanyi, album, lalu ID Lagu yang ingin dimasukkan kedalam daftar antrian/*queue*. Tampilan akhirnya akan seperti gambar dibawah.

```
>> QUEUE SONG;

Daftar Penyanyi:
1. BLACKPINK
2. Arctic Monkeys

Masukkan nama penyanyi yang dipilih: BLACKPINK;

Daftar Album:
1. BORN PINK
2. THE ALBUM

Masukkan nama album yang dipilih: BORN PINK;

Daftar Lagu:
1. Pink Venom
2. Shut Down
3. Typa Girl
4. Ready For Love

Masukkan ID lagu yang dipilih: 1;

Menambahkan lagu. . .

Berhasil menambahkan lagu "Pink Venom" oleh "BLACKPINK" ke queue.
```

Gambar 6.9 Tampilan QUEUE SONG ketika berhasil

### 6.10 QUEUE PLAYLIST

Test ini dilakukan untuk memastikan atau menguji command “QUEUE PLAYLIST” sudah berjalan dengan sesuai. Command ini memasukkan lagu-lagu pada playlist kedalam daftar antrian. Tampilan akhir adalah seperti pada gambar dibawah.

```
>> QUEUE PLAYLIST;
Daftar Playlist:
1. K-Pop
2. Chillin'

Masukkan ID Playlist: 2;

Berhasil menambahkan playlist "Chillin'" ke queue.
```

Gambar 6. 10. Tampilan QUEUE PLAYLIST ketika berhasil

### 6.11 QUEUE SWAP <>

Test ini dilakukan untuk memastikan atau menguji command “QUEUE SWAP” sudah berjalan dengan sesuai. Command ini akan menukar urutan lagu dengan

```
>> QUEUE SWAP 1 2;

Mencari lagu dengan urutan ke 1 dan 2 dalam queue. . .

Lagu "R U Mine?" berhasil ditukar dengan "How You Like That".
```

Gambar 6. 11. Tampilan QUEUE SWAP ketika berhasil

### 6.12 QUEUE REMOVE <>

Test ini dilakukan untuk memastikan atau menguji command “QUEUE REMOVE” sudah berjalan dengan sesuai.

```
>> QUEUE REMOVE;

Mencari lagu dengan urutan ke 0 dalam queue. . .

Lagu dengan urutan ke 0 tidak ada.
```

STEI- ITB	IF2111-TB-K3-6	Halaman 14 dari 28 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.		

Gambar 6. 12. Tampilan QUEUE REMOVE ketika gagal

### 6.13 QUEUE CLEAR

Test ini dilakukan untuk memastikan atau menguji command “QUEUE CLEAR” sudah berjalan dengan sesuai.

```
>> QUEUE CLEAR;  
  
Mengosongkan queue. . .  
  
Queue berhasil dikosongkan.
```

Gambar 6. 13. Tampilan QUEUE CLEAR ketika berhasil

### 6.14 SONG NEXT

Test ini dilakukan untuk memastikan atau menguji command “SONG NEXT” sudah berjalan dengan sesuai.

```
>> SONG NEXT;  
  
Tidak ada lagu dalam antrian.  
  
Tidak ada lagu yang dapat diputar.
```

```
>> SONG NEXT;  
  
Tidak ada lagu dalam antrian.  
  
Memutar kembali lagu. . .  
Memainkan lagu "505" oleh "Arctic Monkeys"
```

```
>> SONG NEXT;  
  
Memutar lagu selanjutnya. . .  
  
Memainkan lagu "Ready For Love" oleh "BLACKPINK"
```

### 6.15 SONG PREVIOUS

Test ini dilakukan untuk memastikan atau menguji command “SONG PREVIOUS” sudah berjalan dengan sesuai.

```
>> SONG PREVIOUS;  
  
Belum ada lagu yang diputar.
```

```
>> SONG PREVIOUS;

Riwayat lagu kosong.

Memutar kembali lagu. . .

Memainkan lagu "505" oleh "Arctic Monkeys"
```

```
>> SONG PREVIOUS;

Memutar lagu sebelumnya. . .

Memainkan lagu "Pink Venom" oleh "BLACKPINK"
```

## 6.16 PLAYLIST CREATE

Test ini dilakukan untuk memastikan atau menguji command “PLAYLIST CREATE” sudah berjalan dengan sesuai.

```
>> PLAYLIST CREATE;

Masukkan nama playlist yang ingin dibuat : K-Pop;

Playlist K-Pop berhasil dibuat!

Silahkan masukkan lagu - lagu artis terkini kesayangan Anda!
```

Gambar 6.16 Tampilan PLAYLIST CREATE apabila berhasil

## 6.17 PLAYLIST ADD SONG

Test ini dilakukan untuk memastikan atau menguji command “PLAYLIST ADD SONG” sudah berjalan dengan sesuai.

```
>> 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 : AM;

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

Masukkan ID lagu yang dipilih: 3;
Daftar Playlist Pengguna :
1. K-Pop
2. Chillin'

Masukkan ID Playlist yang dipilih : 2;

Lagu dengan judul 'Arabella' pada album AM oleh penyanyi Arctic Monkeys berhasil ditambahkan ke dalam playlist Chillin'.
```

Gambar 6.17 Tampilan PLAYLIST ADD SONG apabila berhasil

## 6.18 PLAYLIST ADD ALBUM

Test ini dilakukan untuk memastikan atau menguji command “PLAYLIST ADD ALBUM” sudah berjalan dengan sesuai.

STEI- ITB	IF2111-TB-K3-6	Halaman 16 dari 28 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.		



```

>> 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 : Humbug;

Daftar Playlist Pengguna :
1. K-Pop
2. Chillin'

Masukkan ID Playlist yang dipilih : 2;

Album dengan judul 'Humbug' berhasil ditambahkan ke dalam playlist Chillin'.

```

Gambar 6.18 Tampilan PLAYLIST ADD ALBUM apabila berhasil

### 6.19 PLAYLIST SWAP ◇ ◇ ◇

Test ini dilakukan untuk memastikan atau menguji command “PLAYLIST SWAP” sudah berjalan dengan sesuai.

### 6.20 PLAYLIST REMOVE ◇ ◇

Test ini dilakukan untuk memastikan atau menguji command “PLAYLIST REMOVE” sudah berjalan dengan sesuai.

### 6.21 PLAYLIST DELETE

Test ini dilakukan untuk memastikan atau menguji command “PLAYLIST DELETE” sudah berjalan dengan sesuai.

```

>> PLAYLIST DELETE;

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

Masukkan ID Playlist yang dipilih : 1;

```

### 6.22 STATUS

Test ini dilakukan untuk memastikan atau menguji command “STATUS” sudah berjalan dengan sesuai.

```

>> STATUS;

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

Queue:
Your queue is empty.

```

```
>> STATUS;

Now Playing:
BLACKPINK - How You Like That - THE ALBUM

Queue:
Your queue is empty.
```

```
>> STATUS;

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

Queue:
1. BLACKPINK - Ready For Love - BORN PINK
2. BLACKPINK - Shut Down - BORN PINK
```

```
>> STATUS;

Current Playlist: K-Pop

Now Playing:
BLACKPINK - Pink Venom - BORN PINK

Queue:
Your queue is empty.
```

```
>> STATUS;

Current Playlist: K-Pop

Now Playing:
BLACKPINK - Pink Venom - BORN PINK

Queue:
1. Arctic Monkeys - Crying Lightning - Humbug
2. Arctic Monkeys - Arabella - AM
3. BLACKPINK - Bet You Wanna (Feat. Cardi B) - THE ALBUM
>>
```

### 6.23 SAVE

Test ini dilakukan untuk memastikan atau menguji command “SAVE” sudah berjalan dengan sesuai.

### 6.24 QUIT

Test ini dilakukan untuk memastikan atau menguji command “QUIT” sudah berjalan dengan sesuai.

### 6.25 ENHANCE

### 6.26 INVALID COMMAND

## 7. Text Script

Tabel 7.1 Text Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Fitur Start	Memeriksa apakah fitur start dapat membaca file txt	Mengetikkan command START pada terminal	START	Fungsi berhasil mengeluarkan output yang diharapkan dan berhasil membaca file config.txt	Sesuai yang diharapkan
2	Fitur Load	Memeriksa apakah fitur load dapat me-load file eksternal	Mengetikkan command LOAD dilanjutkan dengan file txt konfigurasinya	LOAD	Fungsi berhasil dijalankan, mengeluarkan output yang diharapkan, dan berhasil melakukan load file eksternal	Sesuai yang diharapkan
3	Fitur List Default	Memeriksa apakah fitur list default dapat menampilkan list penyanyi yang ada	Mengetikkan command LIST DEFAULT pada terminal	LIST DEFAULT	Fungsi berhasil menampilkan daftar penyanyi, diikuti dengan menerima inputan apakah ingin melihat album dan menampilkannya, dilanjut dengan menerima input apakah ingin melihat lagu dan menampilkannya	Sesuai yang diharapkan
4	Fitur List Playlist	Memeriksa apakah fitur list playlist dapat menampilkan playlist yang ada pada pengguna	Mengetikkan command LIST PLAYLIST pada terminal	LIST PLAYLIST	Fungsi dapat menampilkan daftar playlist yang dimiliki	Sesuai yang diharapkan
5	Fitur Play Song	Memeriksa apakah fitur play song dapat digunakan untuk memainkan lagu berdasarkan inputan	Mengetikkan command PLAY SONG pada terminal, lalu mengetikkan nama artis dan album yang ada pada list, setelah itu memilih ID lagu yang ingin diputar	PLAY SONG	Fungsi dapat menampilkan dan menerima masukan untuk memutar lagu dan dapat mengosongkan queue dan riwayat lagu	Sesuai yang diharapkan

6	Fitur Play Playlist	Memeriksa apakah fitur play playlist dapat digunakan untuk memainkan lagu berdasarkan id playlist	Mengetikkan command PLAY PPLAYLIST pada terminal, lalu mengetikkan ID playlist yang ingin diputar	PLAY PLAYLIST	Fungsi dapat menampilkan playlist yang sedang dimainkan dan queue berisi lagu yang ada dalam playlist dan isi riwayat lagu di-reverse dari urutan queue	Sesuai yang diharapkan
7	Fitur Queue Song	Memeriksa apakah fitur queue song dapat menambahkan lagu ke dalam queue	Mengetikkan command QUEUE SONG pada terminal, lalu mengetikkan nama artis dan album yang ada pada list, setelah itu memilih ID lagu yang ingin dimasukkan ke queue	QUEUE SONG	Fungsi dapat menampilkan permintaan masukan, dan inputan berhasil dimasukkan ke dalam queue	Sesuai yang diharapkan
8	Fitur Queue Playlist	Memeriksa apakah fitur queue playlist dapat menambahkan lagu yang ada dalam playlist ke dalam queue	Mengetikkan command QUEUE PLAYLIST pada terminal, lalu mengetikkan ID playlist yang ingin dimasukkan ke queue	QUEUE PLAYLIST	Fungsi dapat menampilkan permintaan masukan id playlist dan berhasil dimasukkan ke dalam queue	Sesuai yang diharapkan
9	Fitur Queue Swap	Memeriksa apakah fitur queue swap dapat menukar urutan lagu	Mengetikkan command QUEUE SWAP dilanjutkan dengan 2 buah integer yang merupakan urutan lagu ke berapa pada queue yang ingin di tukar	QUEUE SWAP	Fungsi berhasil menukar urutan queue yang diminta oleh user	Sesuai yang diharapkan
10	Fitur Queue Remove	Memeriksa apakah fitur queue remove dapat menghapus lagu dari queue	Mengetikkan command QUEUE REMOVE dilanjutkan dengan integer yang menunjukkan lagu di urutan berapa	QUEUE REMOVE	Fungsi dapat menampilkan perintah input urutan lagu dan lagu berhasil dihapus dari queue	Sesuai yang diharapkan

			yang ingin dihapus dari queue			
11	Fitur Queue Clear	Memeriksa apakah fitur queue clear dapat mengosongkan queue	Mengetikkan command QUEUE CLEAR	QUEUE CLEAR	Fungsi dapat menampilkan output queue berhasil dikosongkan	Sesuai yang diharapkan
12	Fitur Song Next	Memeriksa apakah fitur song next dapat digunakan untuk memutar lagu dalam queue	Mengetikkan command SONG NEXT pada terminal	SONG NEXT	Fungsi dapat menampilkan output lagu yang sedang diputar kemudian ditambah ke daftar riwayat lagu	Sesuai yang diharapkan
13	Fitur Song Previous	Memeriksa apakah fitur song previous dapat memutar lagu yang terakhir kali diputar	Mengetikkan command SONG PREVIOUS pada terminal	SONG PREVIOUS	Fungsi dapat menampilkan output lagu yang terakhir diputar dan menambahkan ke dalam queue urutan pertama	Sesuai yang diharapkan
14	Fitur Playlist Create	Memeriksa apakah fitur playlist create dapat membuat playlist baru dan ditambahkan pada daftar playlist pengguna	Mengetikkan command PLAYLIST CREATE lalu mengetikkan nama playlist yang ingin dibuat.	PLAYLIST CREATE	Fungsi dapat mengeluarkan perintah input nama playlist dan ditambah ke daftar playlist pengguna	Sesuai yang diharapkan
15	Fitur Playlist Add Song	Memeriksa apakah fitur playlist add song dapat menambahkan lagu pada playlist yang sudah ada	Mengetikkan command PLAYLIST ADD SONG lalu mengetikkan nama penyanyi dan album sesuai list dan mengetikkan ID lagu yang ingin dimasukkan kedalam playlist, lalu tahap terakhir adalah memilih ID Playlist tempat lagu tersebut akan dimasukkan.	PLAYLIST ADD SONG	Fungsi dapat mengeluarkan perintah input dan menambahkan lagu pada suatu playlist yang sudah ada	Sesuai yang diharapkan

16	Fitur Playlist Add Album	Memeriksa apakah fitur playlist add album dapat menambahkan lagu pada playlist yang sudah ada	Mengetikkan command <b>PLAYLIST ADD ALBUM</b> lalu memilih nama album yang ingin dimasukkan serta memilih ID Playlist termpat album tersebut dimasukkan	<b>PLAYLIST ADD ALBUM</b>	Fungsi dapat mengeluarkan perintah input dan menambahkan lagu-lagu dari suatu album pada playlist yang sudah ada	Sesuai yang diharapkan
17	Fitur Playlist Swap	Memeriksa apakah fitur playlist swap dapat menukar urutan lagu pada playlist	Mengetikkan command <b>PLAYLIST SWAP</b> diikuti dengan 2 integer yang merepresentasikan urutan lagu yang ada di playlist yang ingin ditukar urutannya	<b>PLAYLIST SWAP</b>	Fungsi dapat mengeluarkan perintah input dan menukar urutan lagu pada playlist yang diminta	Sesuai yang diharapkan
18	Fitur Playlist Remove	Memeriksa apakah fitur playlist remove dapat menghapus lagu pada playlist	Mengetikkan command <b>PLAYLIST REMOVE</b> diikuti dengan integer sebagai urutan lagu yang ingin dihapus dari playlist	<b>PLAYLIST REMOVE</b>	Fungsi dapat mengeluarkan perintah input dan menghapus lagu pada urutan playlist	Sesuai yang diharapkan
19	Fitur Playlist Delete	Memeriksa apakah fitur playlist delete dapat menghapus playlist yang sudah ada	Mengetikkan commad <b>PLAYLIST DELETE</b> lalu memasukkan ID Playlist yang ingin dihapus dari daftar playlist	<b>PLAYLIST DELETE</b>	Fungsi dapat mengeluarkan perintah input dan menghapus playlist yang ada dalam daftar playlist pengguna	Sesuai yang diharapkan
20	Fitur Status	Memeriksa apakah fitur status dapat menampilkan lagu, queue, dan playlist yang sedang dimainkan	Mengetikkan command <b>STATUS</b> pada terminal	<b>STATUS</b>	Fungsi dapat menampilkan lagu yang sedang dimainkan beserta queue yang ada, dan dari playlist mana lagu diputar	Sesuai yang diharapkan

21	Fitur Save	Memeriksa apakah fitur save dapat menyimpan state aplikasi ke dalam suatu file	Mengetikkan command SAVE diikuti dengan nama file	SAVE	Fungsi dapat menyimpan state aplikasi terbaru ke dalam suatu file	Sesuai yang diharapkan
22	Fitur Quit	Memeriksa apakah fitur quit dapat mengeluarkan dari aplikasi WayangWave	Mengetikkan command QUIT lalu mengetikan Y untuk melanjutkan dan N untuk membatalkan keluar dari program	Data Test 24	Fungsi dapat mengeluarkan pengguna dari aplikasi WayangWave	Sesuai yang diharapkan
23	Fitur Help	Memeriksa apakah fitur help dapat menampilkan daftar command beserta deskripsinya	Mengetikkan command HELP ke pada terminal	HELP	Fungsi dapat menampilkan pesan atau daftar command beserta deskripsinya	Sesuai yang diharapkan
24	Enhance	Memasukkan lagu secara acak ke dalam playlist	Mengetikkan command ENHANCE di main, lalu input id playlist yang ingin di enhance	ENHANCE		
25	Invalid Command	Memeriksa apakah fitur invalid command dapat menampilkan state eror suatu command	Mengetikkan command yang tidak sesuai dengan yang seharusnya	Data Test 26	Mengeluarkan "command tidak valid"	Sesuai yang diharapkan

## 8. Pembagian Kerja dalam Kelompok

Tabel 8.1 Pembagian Kerja

Fitur	NIM Coder	NIM Tester	Catatan
START	18222115	18222116	
LOAD	18222133	18222116	
LIST	18222111	18222111	
PLAY	18222110	18222116 18222110	
QUEUE	18222110	18222116	

STEI- ITB	IF2111-TB-K3-6	Halaman 23 dari 28 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.		

SONG	18222110	18222116	
PLAYLIST	18222115	18222115	
STATUS	18222133	18222116	
SAVE	18222115	18222116	
QUIT	18222116	18222116	
HELP	18222116	18222116	
INVALID COMMAND	18222116	18222116	
ADT List	18222110 18222111	18222111	
ADT Mesin Karakter & Mesin Kata	18222133 18222111	18222133 18222111	
ADT Queue	18222111	18222116	
ADT Stack	18222116	18222116	
ADT Set & Map	18222111	18222111	
ADT List dengan struktur berkait	18222115	18222115	
Enhance	18222116 18222111	18222116 18222111	

## 9. Lampiran

### 9.1 Deskripsi Tugas Besar

Membuat sebuah aplikasi simulasi berbasis CLI (command-line interface). Sistem ini dibuat dalam bahasa C dengan menggunakan struktur data yang sudah dipelajari di mata kuliah Algoritma Struktur Data STI. Kita boleh menggunakan atau memodifikasi struktur data yang sudah kita 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`

Tentang sistemnya WayangWave merupakan sebuah aplikasi yang bisa mensimulasikan service pemutar 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

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 seperti command PLAY, QUEUE, SONG, PLAYLIST, STATUS, SAVE, QUIT, dan jika suatu command gagal atau eror untuk dijalankan atau bisa disebut INVALID COMMAND.

### 9.2 Notulensi Asistensi

#### 9.2.1. Notulensi Asistensi 1

Tabel 9.2.1 Catatan Asistensi ke-1

STEI- ITB	IF2111-TB-K3-6	Halaman 24 dari 28 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.		









<b>Tanggal : 31 Oktober 2023</b>	<b>Catatan Asistensi:</b>  Start -> memulai aplikasi hanya akan terdapat daftar penyanyi, penyanyi punya album apa, dan di album ada lagu apa. Jd gak ada lagu yg sudah diputar, antrean lagu, playlist.  List playlist: isinya playlist yang nanti playlist itu consist lagu-lagu yang menggunakan ADT list berkait  Play playlist kalau kosong bagaimana? Nanti akan dicoba buat implementasinya sebagai gambaran, seharusnya ada output bahwa playlist kosong. Playlist make list structure berkait, yang mana masing-masing list punya strukturnya masing-masing.  Ada struktur yang menyimpang si list strukturnya sendiri dan ada listnya yang nyimpen setiap lagunya. Ada list yang menyimpan isi si list dari playlistnya dan ada struktur berkait yang nyimpen list playlist  ADT yg dipakai di invalid command : ADT mesin kata sama mesin karakter buat baca command masukkan usernya. Berlaku juga di start dan load dan yang lain-lainnya juga.  Queue ada contoh tentang deret, nanti di output dalam bentuk string atau deretan kata?  File config dalam bentuk txt -> kalau start bisa langsung kebaca dan bisa langsung kepake, list artis  Console: menyimpan fungsi2 yang mewakili semua command, main hanya manggil semua yang ada di console.
<b>Tempat : Zoom Meeting</b>	
<b>Kehadiran Anggota Kelompok:</b>  <div style="text-align: center;"> <b>18222110</b>            Gracya Tio Damena Sidabutar   </div> <div style="text-align: center;"> <b>18222111</b>            Moh Afnan Fawaz   </div> <div style="text-align: center;"> <b>18222115</b>            Gabriel Marcellino Josetiawan   </div> <div style="text-align: center;"> <b>18222116</b>            Jason Jahja   </div> <div style="text-align: center;"> <b>18222133</b>            Hanan Fitra Salam   </div>	
	<b>Tanda Tangan Asisten:</b>  

### 9.2.2. Notulensi Asistensi 2

Tabel 9.2.2 Catatan Asistensi ke-2

<b>STEI- ITB</b>	<b>IF2111-TB-K3-6</b>	<b>Halaman 25 dari 28 halaman</b>
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.		

<b>Tanggal : 13 November 2023</b>	<b>Catatan Asistensi:</b>  Apakah sebaiknya fungsi di load tidak menggunakan fungsi yang ada di start? Tidak ada ketentuan, kalo fungsi load tidak boleh menggunakan fungsi dari fungsi start, jangan sampai menutup kemungkinan variasi kode yang bisa digunakan di kedua fungsi tersebut.  Mencoba menggunakan struktur baru, tapi adt saling circular include, sudah dicoba memisahkan ke masing masing file header adakah solusinya kak? Dicoba dicari solusi dari masalahnya, bisa dengan cara primitif yaitu menggambarkan masing masing struktur agar tidak sirkular include atau dependensi satu sama lain.  Apakah sebuah struct a memanggil struct b, apakah struct b bisa memanggil struct a lagi, apakah tipe pointer bisa dimanfaatkan untuk masalah seperti ini? Tidak menutup kemungkinan kalau itu bisa, soalnya pertanyaanya terlalu spesifik, kalau untuk masalah tipe pointer sepertinya beda kasus jadi tidak bisa digunakan, sebaiknya melakukan restrukturisasi tipe yang sudah kita bikin.  Apakah list untuk memakai list statis saja, tidak memakai list dinamis? sudah kategori nya list dinamis dipakai, sayang juga kalo misal memakai list statis nanti isi playlistnya kosong memorinya dipakai untuk mengalokasikan playlist kosong yang seharusnya bisa dipakai untuk hal lain.  Setiap fungsi kan bisa menyimpan variable, baiknya variabel-variabel tersebut biar bisa digunakan global ditaro dimana ya kak? Bisa aja variable tersebut kalo ingin global diinisialisasi di main, yang penting value dari variabel yang ada di main bisa dikendalikan, tapi jangan dipaksain variable ada di main
<b>Tempat : Zoom Meeting</b>	
<b>Kehadiran Anggota Kelompok:</b>  <div style="text-align: center;"> <b>18222110</b>            Gracya Tio Damena Sidabutar   </div> <div style="text-align: center;"> <b>18222111</b>            Moh Afnan Fawaz   </div> <div style="text-align: center;"> <b>18222115</b>            Gabriel Marcellino Josetiawan   </div> <div style="text-align: center;"> <b>18222116</b>            Jason Jahja   </div> <div style="text-align: center;"> <b>18222133</b>            Hanan Fitra Salam   </div>	
	<b>Tanda Tangan Asisten:</b>

	
--	--

### 9.3 Notulensi Rapat

#### 9.3.1. Rapat Perdana

Hari, tanggal: Jumat, 27 Oktober 2023

Waktu: 22.00 – 22.45 WIB

Pembahasan: Pembagian tugas awal, ADT, fungsi *Command*, dan bagian dokumen

#### 9.3.2. Rapat Kedua

Hari, tanggal: Minggu, 12 November 2023

Waktu: 19.00 – 20.00 WIB

Pembahasan: Update progress tugas awal dan pembagian deadline masing-masing tugas

### 9.4 Log Activity Kelompok

#### 9.4.1. Gracya Tio Damena Sidabutar – 18222110

01/11/2023 : Membuat ADT List Statis (masih dengan elemen bentuk int)

19/11/2023: Menyelesaikan code Song

20/11/2023: Menyelesaikan code Play dan Queue

21/11/2023: Mulai mengerjakan Laporan

22/11/2023: Memperbaiki kesalahan minor codingan

23/11/2023: Melakukan data test

#### 9.4.2. Moh Afnan Fawaz – 18222111

01/11/2023: Membuat ADT Queue (masih dengan elemen bentuk int)

08/11/2023: Mengganti dan mengubah ADT Queue dengan elemen bentuk Word

16/11/2023: Membuat struktur-struktur tambahan (Penyanyi, Album, Lagu)

17/11/2023: Membuat struktur tambahan secara terpisah (sesuai dengan ADT yang diperlukan)

19/11/2023: Membuat implementasi dari ADT list, map, dan set

20/11/2023: Membuat driver masing-masing dan gabungan dari ADT list, map, dan set

21/11/2023: Berkontribusi dalam pembuatan Spesifikasi Start dan membuat primitif tambahan pada mesin kata dan mesin baris.

22/11/2023: Membuat spesifikasi List Default dan ADT dari implementasi dari List Dinamis (untuk List Playlist)

23/11/2023: Membuat Spesifikasi List Playlist

#### 9.4.3. Gabriel Marcellino Joestiawan – 18222115

08/11/2023: Mengerjakan ADT

16/11/2023: Membantu mengerjakan fungsi START

17/11/2023: Mengerjakan fungsi SAVE

19/11/2023: Mengerjakan fungsi PLAYLIST

STEI- ITB	IF2111-TB-K3-6	Halaman 27 dari 28 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.		

23/11/2023: Mengisi laporan

**9.4.4. Jason Jahja – 18222116**

01/11/2023: Mengerjakan ADT stack dan spesifikasi help

08/11/2023: Membuat driver untuk beberapa spesifikasi seperti list

16/11/2023: Melengkapi dan merapikan help serta mulai membuat main

17/11/2023: Merevisi stack

19/11/2023: Memfinalisasi stack dan membantu spesifikasi lain

20/11/2023: Merapikan semua file dalam algoritma dan output

21/11/2023: Membuat driver untuk stack dan queue

22/11/2023: Membuat driver Song, main, mesin baris

23/11/2023 : Memfinalisasi main, merapikan output semua spesifikasi

**9.4.5. Hanan Fitra Salam – 18222133**

01/11/2023 : Mengerjakan ADT mesinkata dan mesinkarakter

08/11/2023 : Melanjutkan mengerjakan ADT map dan set

16/11/2023 : Mengerjakan command START

19/11/2023 : Mengerjakan command STATUS

20/11/2023: Mengerjakan command LOAD

21/11/2023: Melanjutkan command LOAD

22/11/2023: Mengerjakan laporan dan komentar pada ADT