

LAPORAN TUGAS BESAR

IF2111 Algoritma dan Struktur Data STI

WayangWave


Dipersiapkan oleh:

Kelompok 5

Bobby Dhammawan TANG	- 18220037
Jihan Aurelia	- 18222001
Benedicta Eryka Santosa	- 18222031
Firsa Athaya Raissa Alifah	- 18222051
Muhammad Faiz A	- 18222063

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		IF2111-TB-01-05		47
		Revisi	0	Jumat, 24 Oktober 2023

Daftar Isi

1	Ringkasan	3
2	Penjelasan Tambahan Spesifikasi Tugas	5
2.1	Fitur Banner	5
2.2	Fitur QUIT	6
2.3	Fitur List Playlist	6
2.4	Fitur Enhance	7
2.5	Fitur Multi User	7
3	Struktur Data (ADT)	8
3.1	Struktur ADT Song	8
3.2	Struktur Data Object	9
3.3	Struktur Data List	9
3.4	Struktur Data ArrayDin	10
3.5	Struktur Data ArrayDinWord	12
3.6	Struktur Data Mesin Karakter	13
3.7	Struktur Data Mesin Kata	14
3.8	Struktur Data Queue	15
3.9	Struktur Data Stack	16
3.10	Struktur Data Set	17
3.11	Struktur Data Map	18
3.12	Struktur List dengan Struktur Berkait	19
3.13	Struktur ADT User	21
4	Program Utama	22
5	Algoritma-Algoritma Menarik	22
5.1	Enhance	22
6	Data Test	24
6.1	Data Test START	24
6.2	Data Test LOAD	24
6.3	Data Test LIST DEFAULT	25
6.4	Data Test LIST PLAYLIST	25
6.5	Data Test PLAY SONG	26
6.6	Data Test PLAY PLAYLIST	26
6.7	Data Test QUEUE SONG	27
6.8	Data Test QUEUE PLAYLIST	27
6.9	Data Test QUEUE SWAP	27

6.10 Data Test QUEUE REMOVE	28
6.11 Data Test QUEUE CLEAR	28
6.12 Data Test SONG NEXT	28
6.13 Data Test SONG PREVIOUS	28
6.14 Data Test PLAYLIST CREATE	29
6.15 Data Test PLAYLIST ADD	29
6.16 Data Test PLAYLIST SWAP	30
6.17 Data Test PLAYLIST REMOVE	30
6.18 Data Test PLAYLIST DELETE	31
6.19 Data Test STATUS	31
6.20 Data Test SAVE	32
6.21 Data Test QUIT	32
6.22 Data Test HELP	32
6.23 Data Test <INVALID COMMAND>	33
7 Test Script	33
8 Pembagian Kerja dalam Kelompok	35
9 Lampiran	36
9.1 Deskripsi Tugas Besar	36
9.2 Notulen Rapat	40
9.3 Log Activity Anggota Kelompok	44

1 Ringkasan

WayangWave adalah sebuah aplikasi yang bisa mensimulasikan *service* pemutaran musik. Aplikasi ini memiliki fitur utama, yaitu memutar lagu, menampilkan daftar lagu, membuat dan menghapus *playlist*, mengatur urutan dimainkannya lagu, dan menampilkan status dari aplikasi. Berikut adalah penjelasan mengenai cara kerja dari program WayangWave ini.

Pada saat awal memulai program WayangWave, program akan menampilkan *main menu* yang berisi *welcome page*. Pada fase tersebut, pengguna diberi tiga pilihan opsi mengenai cara untuk menjalankan program ini yaitu START, LOAD, dan HELP. Para pengguna dapat memilih satu dari ketiga *command* yang diberikan. Ketika pengguna memberikan *command* lain sebagai perintah pertama, program akan menampilkan bahwa masukan dari pengguna tidak valid, dan akan diminta untuk memberikan perintah kembali sampai perintah yang diberikan oleh pengguna adalah salah satu dari START, LOAD, dan HELP.

Perbedaan diantara ketiga opsi perintah pertama terletak pada kondisi *state* yang digunakan. Ketika pengguna memilih START untuk memulai lalu menekan ENTER, maka program akan membaca *file* konfigurasi *default* yang berisi penyanyi serta album yang dimiliki. Ketika pengguna memilih LOAD lalu menekan ENTER, program akan membaca *save file* *<filename>*. *Command* LOAD ini, memiliki satu argumen yaitu *filename* yang didapatkan dari folder *save*. *Save file* *<filename>* adalah *file* yang berisi list penyanyi, album, dan lagu yang bisa diputar. Sedangkan, apabila pengguna memilih HELP lalu menekan ENTER, program akan menampilkan daftar *command* yang mungkin untuk dieksekusi beserta dengan deskripsinya.

Saat pengguna telah memilih cara dan mulai menjalankan program, pengguna dapat menjalankan fitur-fitur yang terdapat pada program WayangWave melalui pemanggilan *command*. Daftar *command* yang dapat diberikan adalah antara lain LIST, PLAY, QUEUE, SONG, PLAYLIST, STATUS, SAVE *<filename>*, QUIT, dan HELP. Apabila pengguna memberikan *command* selain *command-command* yang telah disebutkan, maka masukan dari pengguna tersebut akan dianggap tidak valid (*command tidak ditemukan!*) dan tidak akan diproses oleh program. Penjelasan singkat mengenai *command* fitur-fitur pada program WayangWave adalah sebagai berikut.

Pertama, *command* LIST adalah perintah yang digunakan untuk menampilkan *list playlist* yang ada, *list* penyanyi, *list* album dari penyanyi, dan *list* lagu yang ada di album. Pada *command* LIST terdapat dua jenis *list*, yaitu LIST DEFAULT yang digunakan untuk melihat *list* penyanyi yang ada, lalu album dari penyanyi yang dipilih, kemudian lagu yang ada dari album yang dipilih. Lalu, ada LIST PLAYLIST yang digunakan untuk menampilkan *playlist* yang ada pada pengguna. Kedua, terdapat *command* PLAY yang merupakan *command* untuk memutar lagu atau *playlist* yang dipilih. Saat dieksekusi, *queue* yang ada dihapus ketika memainkan lagu atau digantikan oleh lagu yang berada di *playlist* ketika memainkan *playlist*. Pada *command* PLAY, terdapat dua jenis *play*, yaitu SONG dan PLAYLIST. PLAY SONG adalah *command* yang digunakan untuk memainkan lagu berdasarkan masukan nama penyanyi, nama album, dan id lagu. Saat dieksekusi, *queue* dan riwayat lagu akan menjadi kosong. *Command* yang kedua adalah PLAY PLAYLIST yang berguna untuk memainkan lagu berdasarkan *id playlist*. Saat

dieksekusi, *current song* akan menjadi lagu pada urutan pertama *playlist* dan *queue* akan berisi semua lagu yang ada dalam *playlist* dengan urutan yang di-reverse. Ketiga, terdapat *command* QUEUE untuk memanipulasi *queue* lagu yang memiliki 5 tipe, yaitu QUEUE SONG (menambahkan lagu ke dalam *queue* dengan input berdasarkan nama penyanyi, nama album, dan id lagu), QUEUE PLAYLIST (menambahkan lagu yang ada di dalam *playlist* ke dalam *queue* dengan *input id playlist*), QUEUE SWAP (menukar urutan lagu), QUEUE REMOVE (menghapus lagu dari *queue*), dan QUEUE CLEAR (mengosongkan *queue*). Keempat, terdapat *command* SONG untuk navigasi lagu yang ada pada *queue*. Command ini memiliki 2 tipe navigasi, yaitu SONG NEXT (memutar lagu yang berada di dalam *queue*) dan SONG PREVIOUS (memutar lagu yang terakhir kali diputar). Kelima, terdapat *command* PLAYLIST yang terdiri dari 5 tipe, yaitu PLAYLIST CREATE (membuat *playlist* baru dan ditambahkan pada daftar *playlist* pengguna), PLAYLIST ADD (menambahkan lagu pada suatu *playlist* yang telah ada sebelumnya pada daftar *playlist* pengguna, PLAYLIST SWAP (menukar lagu di *playlist*), PLAYLIST REMOVE (menghapus lagu pada *playlist*), dan PLAYLIST DELETE (menghapus *playlist* dalam daftar *playlist* pengguna). Keenam, terdapat *command* STATUS untuk menampilkan lagu yang sedang dimainkan beserta *queue song* yang ada dan informasi *playlist*-nya. Terakhir, ada *command* QUIT yang digunakan untuk keluar dari aplikasi WayangWave.

Pada aplikasi ini, saat memulai setiap *session*, sebuah *file* konfigurasi akan dibaca sesuai dengan *session* yang berlaku. Terdapat dua jenis *file* konfigurasi yang mewakili sebuah *state* yang berbeda, yaitu *state default* ketika suatu *session* baru benar-benar dimulai, dan *state* ketika suatu *session* sudah pernah berjalan dan *state*-nya di-save ke dalam *file* lalu ingin di-load kembali.

Program WayangWave ini menggunakan bahasa pemrograman C dan akan menerapkan beberapa struktur data yang sesuai. ADT yang digunakan pada program ini adalah ADT List, ADT Mesin Karakter dan Mesin Kata, ADT Queue, ADT Stack, ADT Set & Map, dan ADT List dengan Struktur Berkait. Hal-hal lain mengenai ADT yang digunakan pada program ini akan dijelaskan lebih *detail* pada poin nomor 3. Selain itu, terdapat program utama atau *main* yang digunakan sebagai wadah sinkronisasi antar program *command* sehingga program WayangWave dapat dijalankan secara utuh dan selaras. Hal-hal mengenai program utama akan dijelaskan lebih lanjut pada poin nomor empat, diikuti dengan Data Test dan juga Test Script mengenai program ini secara keseluruhan yang akan dijabarkan pada poin nomor enam dan tujuh.

2 Penjelasan Tambahan Spesifikasi Tugas

2.1 Fitur Banner

Fitur ini sebenarnya hanya fitur kosmetik untuk memperindah tampilan sehingga benar-benar merepresentasikan aplikasi yang telah dirancang. Dapat dilihat pada gambar 2.1.1, terdapat judul dari aplikasi ini, yaitu WayangWave.

2.1.1 Tampilan banner WayangWave

```
888      888      888      888
888  o  888      888  o  888
888  d8b 888      888  d8b 888
888 d888b 888 8888b. 888 888 8888b. 88888b. .d88b. 888 d888b 888 8888b. 888 888 .d88b.
888d88888b888 "88b888 888 "88b888 "88bd88P"88b888d88888b888 "88b888 888d8P Y8b
88888P Y88888.d888888888 888.d888888888 888888 88888888P Y88888.d888888Y88 88P88888888
8888P Y8888888 888Y88b 888888 888888 888Y88b 8888888P Y8888888 888 Y8bd8P Y8b.
888P Y888"Y888888 "Y88888"Y888888888 888 "Y88888888P Y888"Y88888 Y88P "Y8888
      888      888
      Y8b d88P      Y8b d88P
      "Y88P"      "Y88P"

Welcome to WayangWave
```

Pada fitur ini juga menyampaikan bahwa *command* yang bisa digunakan hanya START, LOAD, atau HELP.

2.2 Fitur QUIT

Pada fitur ini kami kami menambahkan bahwa pengguna dapat melakukan SAVE sebelum keluar dari program dan dapat memilih untuk tidak melakukan save.

2.2.1 Tampilan command QUIT saat pengguna ingin menyimpan datanya terlebih dahulu

```
>> ENTER COMMAND: QUIT
Apakah kamu ingin menyimpan data sesi sekarang (Y/N)? Y
Lokasi penyimpanan data sesi sekarang (contoh: config.txt): save.txt
```

2.2.2 Tampilan command QUIT saat pengguna ingin langsung keluar tanpa menyimpan datanya

```
>> ENTER COMMAND: QUIT
Apakah kamu ingin menyimpan data sesi sekarang (Y/N)? N
Kamu keluar dari WayangWave.
Dadah ^ ^/
```

2.3 Fitur List Playlist

Pada fitur ini kami kami menambahkan bahwa pengguna dapat melihat isi playlist setelah melakukan LIST PLAYLIST.

2.3.1 Tampilan command LIST PLAYLIST

```
>> ENTER COMMAND: LIST PLAYLIST
Daftar Playlist Pengguna :
    1. BLACKPINK My Top Three
    2. Arctic Monkeys My Top 5
    3. Mixed Playlist

Ingin melihat isi playlist yang ada?(Y/N): Y
Masukkan ID Playlist yang dipilih : 1
Lagu yang berada pada playlist "BLACKPINK My Top Three":
    1. BLACKPINK - How You Like That - THE ALBUM
    2. BLACKPINK - Tyla Girl - BORN PINK
    3. BLACKPINK - Pink Venom - BORN PINK
```

2.4 Fitur Enhance

Command ENHANCE digunakan untuk menambahkan lagu pada playlist pengguna. Algoritma ini dianggap menarik karena penambahan lagu tidak hanya menggunakan fungsi random, melainkan menggunakan data seperti lagu yang mirip terhadap lagu lainnya pada playlist atau lagu yang mirip terhadap lagu yang pernah dimainkan. Ketika pengguna melakukan ENHANCE pada suatu playlist, WayangWave akan merekomendasikan dan menambahkan lagu rekomendasi berdasarkan lagu yang ada pada playlist tersebut (lagu dengan album yang sama pada playlist tersebut) bila playlist tersebut tidak kosong. Jika playlist tersebut kosong dan terdapat lagu pada Stack Riwayat, WayangWave akan merekomendasikan dan menambahkan lagu rekomendasi berdasarkan lagu yang pernah dimainkan (lagu dengan album yang sama pada Stack Riwayat), dan apabila playlist tersebut dan Stack Riwayat pengguna kosong, WayangWave akan merekomendasikan dan menambahkan lagu random. Contoh command lengkapnya dapat dilihat pada bagian [Algoritma-Algoritma Menarik: Enhance](#).

2.5 Fitur Multi User

Fitur Multi User memungkinkan data lagu yang sedang dimainkan, antrian putar, riwayat putar, dan playlist terkait pada sebuah pengguna. Setelah melakukan START atau LOAD, pengguna perlu melakukan login bila sudah terdapat pengguna terdaftar atau register sebelum menggunakan WayangWave.

2.5.1 Tampilan command LOGIN untuk masuk ke WayangWave

```
>> ENTER COMMAND: LOAD default.txt
Save file berhasil dibaca. Lakukan Login untuk menjalankan WayangWave.

>> ENTER COMMAND: LOGIN
Masukkan username user WayangWave : bandung bondowoso
Berhasil masuk. Selamat datang, bandung bondowoso!

>> ENTER COMMAND: LOGOUT
Berhasil keluar. Login kembali untuk menjalankan WayangWave.
```

2.5.2 Tampilan command REGISTER untuk mendaftarkan pengguna baru

```
>> ENTER COMMAND: LOGIN
Masukkan username user WayangWave : Roro
Tidak ada user dengan username "Roro". Silahkan coba lagi!

>> ENTER COMMAND: REGISTER
Masukkan username baru untuk WayangWave : Roro
Berhasil mendaftarkan pengguna. Selamat datang, Roro!
```

3 Struktur Data (ADT)

Pada program yang kami buat, kami menggunakan beberapa struktur data (ADT) untuk menyelesaikan permasalahan yang ada dan membuat program WayangWave menjadi program yang dapat berjalan dengan baik dan tepat. Beberapa struktur data (ADT) yang kami gunakan adalah ADT List, ADT Mesin Karakter dan Mesin Kata, ADT Queue, ADT Stack, ADT Set & Map, dan ADT List dengan Struktur Berkait.

3.1 Struktur ADT Song

3.1.1 Sketsa Struktur Data

```
typedef struct {  
    int PenyanyiID;  
    int AlbumID;  
    int LaguID;  
    int PlaylistID;  
} Song;
```

Berikut primitif yang digunakan dalam ADT ini:

1. Konstruktor
 - a. void CreateEmptyLagu()
 - b. Song CreateLagu(int PenyanyiID, int AlbumID, int LaguID, int PlaylistID)
2. Tes Empty
 - a. boolean SongIsEmpty(Song s)
3. Lain Lain
 - a. boolean SongIsEqual(Song s1, Song s2)

3.1.2 Persoalan yang dapat diselesaikan

ADT ini digunakan untuk menyimpan data lagu di Queue (antrian putar), Stack (riwayat putar), dan List Berkait (playlist). Dibutuhkannya ADT yang dapat menyimpan informasi lagu berupa Penyanyi, Album, dan Playlist.

3.1.3 Alasan Pemilihan ADT

Dibutuhkan ADT yang dapat menyimpan informasi lagu berupa Penyanyi, Album, dan Playlist. Dengan menggunakan ADT Song, informasi yang perlu disimpan hanya ID Penyanyi, ID Album, ID Lagu, dan ID Playlist.

3.1.4 Implementasi sebagai ADT

./src/ADT/Lagu/
song.h, song.c, driversong.c

3.2 Struktur Data Object

3.2.1 Sketsa Struktur Data

```
typedef struct {  
    int ID;  
    Word Title;  
} Object;
```

Berikut primitif yang digunakan dalam ADT ini:

1. Konstruktor
 - a. Object CreateObject(int ID, Word Title)
2. Lain Lain
 - a. boolean ObjectIsEqual(Object o1, Object o2)

3.2.2 Persoalan yang dapat diselesaikan

Album disimpan dalam Map dengan Key nama penyanyi dan Lagu disimpan dalam Map dengan Key nama album. Persoalan akan datang apabila isi ADT Set dari Value Map merupakan Word dan terdapat dua penyanyi yang memiliki nama album yang sama persis.

3.2.3 Alasan Pemilihan ADT

ADT Object digunakan untuk mencegah permasalahan dua penyanyi yang memiliki album dengan nama yang sama persis.

3.2.4 Implementasi sebagai ADT

Struktur data ini digunakan dalam sebagai salah satu struktur data di ADT Song. pengaksesan dilakukan dengan cara berikut,

./src/ADT/Lagu/
song.h, song.c, driversong.c

3.3 Struktur Data List

3.3.1 Sketsa Struktur Data

```
typedef struct {
```

```

        Word TK[IdxMax-IdxMin+1];
        int Neff;
    } TabKata;

```

Berikut primitif yang digunakan dalam ADT ini:

1. Konstruktor
 - a. void MakeEmpty (TabKata *T)
2. Tes Index
 - a. boolean IsIdxValid (TabKata T, int i)
 - b. boolean IsIdxEff (TabKata T, int i)
3. Prototype
 - a. int searchList(TabKata T, Word v)
 - b. boolean IsMember (TabKata T, Word v)
4. Selektor
 - a. int NbElmt (TabKata T)
 - b. int MaxNbEl (TabKata T)
- c. int GetFirstIdx (TabKata T)
- d. int GetLastIdx (TabKata T)
- e. Word GetElmt (TabKata T, int i)
5. Tes Empty/Full
 - a. boolean IsEmpty (TabKata T)
 - b. boolean IsFull (TabKata T)
6. Insert/Delete
 - a. void SetTab (TabKata Tin, TabKata *Tout)
 - b. void SetEl (TabKata *T, int i, Word v)
 - c. void SetNeff (TabKata *T, int N)
7. Lain Lain
 - a. void TulisIsi (TabKata T)

3.3.2 Persoalan yang dapat diselesaikan

Persoalan yang dapat diselesaikan adalah akses nama penyanyi.

3.3.3 Alasan Pemilihan ADT

ADT ini digunakan karena bisa menyimpan daftar penyanyi bertipe *word*.

3.3.4 Implementasi sebagai ADT

Implementasi dari tipe data List, yaitu TabKata Penyanyi dengan akses ADT pengaksesan sebagai berikut,

```

./src/ADT/List/
array.h, array.c, driverarray.c

```

3.4 Struktur Data ArrayDin

3.4.1 Sketsa Struktur Data

```

#define InitialSize 10

```

```
typedef struct {
    ListLinier *A;
    int Capacity;
    int Neff;
} ArrayDin;
```

Berikut primitif yang digunakan dalam ADT ini:

1. Konstruktor
 - a. ArrayDin MakeArrayDin()
2. Destruktor
 - a. void DeallocateArrayDin(ArrayDin *array)
3. Prototype
 - a. boolean IsEmptyArrayDin(ArrayDin array)
 - b. int LengthArrayDin(ArrayDin array)
 - c. int GetCapacityArrayDin(ArrayDin array)
4. Selektor
 - a. ListLinier GetArrayDin(ArrayDin array, int i)
 - b. ArrayDin CopyArrayDin(ArrayDin array)
5. Insert/Delete
 - a. void InsertAtArrayDin(ArrayDin *array, ListLinier el, int i)
 - b. void InsertLastArrayDin(ArrayDin *array, ListLinier el)
 - c. void InsertFirstArrayDin(ArrayDin *array, ListLinier el)
 - d. void DeleteAtArrayDin(ArrayDin *array, int i)
 - e. void DeleteLastArrayDin(ArrayDin *array)
 - f. void DeleteFirstArrayDin(ArrayDin *array)
6. Lain Lain
 - a. void ReverseArrayDin(ArrayDin *array)

3.4.2 Persoalan yang dapat diselesaikan

ADT list dinamis ini dapat menyelesaikan persoalan terkait menyimpan dan mengelola data dari setiap playlist yang dimiliki pengguna dalam bentuk list dinamis yang terdiri dari struktur berkait (list linier). Data dari playlist mencakup daftar lagu yang ada pada playlist serta informasi tentang ID penyanyi, ID album, ID lagu, dan ID playlist tempat lagu itu berada untuk semua lagu pada setiap playlist. ArrayDin terdiri dari daftar playlist yang ditampung dalam list dinamis. Setiap playlist bertipe linked list dan menyimpan address pertama dari playlist. Address (node dari linked list) menyimpan detail dari lagu-lagu yang ada di playlist tersebut.

3.4.3 Alasan Pemilihan ADT

ADT ini dipilih karena dapat menyimpan lagu-lagu di dalam playlist pengguna dengan metode yang efisien. Perubahan-perubahan seperti jumlah lagu pada playlist (misalnya ketika ingin menghapus lagu dari playlist) menjadi lebih mudah ditangani berkat kapasitas list dinamis yang fleksibel. Struktur berkait untuk data setiap lagu juga digunakan karena pemakaian memori yang efisien dan cocok untuk dipakai pada struktur yang bersarang seperti ADT ini.

3.4.4 Implementasi sebagai ADT

Implementasi dari ADT ArrayDin adalah untuk menampung lagu-lagu dan data terperinci dari setiap lagu pada playlist yang dimiliki pengguna. ADT ini dapat diakses dengan cara berikut:

```
./src/ADT/List/  
arraydin.h, arraydin.c, driverarraydin.c
```

3.5 Struktur Data ArrayDinWord

3.5.1 Sketsa Struktur Data

```
#define InitialSize 10  
  
typedef struct {  
    Word *A;  
    int Capacity;  
    int Neff;  
} ArrayDinWord;
```

Berikut primitif yang digunakan dalam ADT ini:

1. Konstruktor
 - a. ArrayDinWord MakeArrayDinWord()
2. Destruktor
 - a. void DeallocateArrayDinWord(ArrayDinWord *array)
3. Prototype
 - a. boolean IsEmptyArrayDinWord(ArrayDinWord array)
 - b. int LengthArrayDinWord(ArrayDinWord array)
 - c. int GetCapacityArrayDinWord(ArrayDinWord array)
4. Selektor
 - a. Word GetArrayDinWord(ArrayDinWord array, int i)
 - b. ArrayDinWord CopyArrayDinWord(ArrayDinWord array)
 - c. int SearchArrayDinWord(ArrayDinWord array, Word el)
5. Insert/Delete
 - a. void InsertAtArrayDinWord(ArrayDinWord *array, Word el, int i)
 - b. void InsertLastArrayDinWord(ArrayDinWord *array, Word el)

- c. void InsertFirstArrayDinWord(ArrayDinWord *array, Word el)
 - d. void DeleteAtArrayDinWord(ArrayDinWord *array, int i)
 - e. void DeleteLastArrayDinWord(ArrayDinWord *array)
 - f. void DeleteFirstArrayDinWord(ArrayDinWord *array)
6. Lain Lain
- a. void PrintArrayDinWord(ArrayDinWord array)
 - b. void ReverseArrayDinWord(ArrayDinWord *array)

3.5.2 Persoalan yang dapat diselesaikan

ADT list dinamis yang terdiri atas judul playlist bertipe Word, kapasitas list, dan jumlah elemen list yang sudah terisi ini dapat menyelesaikan persoalan terkait mengakses dan memodifikasi daftar playlist yang dimiliki pengguna.

3.5.3 Alasan Pemilihan ADT

ADT list dinamis dipilih karena pengaplikasiannya lebih efisien pada operasi-operasi seperti penambahan atau pengurangan elemen list, yang merupakan playlist pengguna. Hal ini berkaitan dengan alokasi memori pada list dinamis dimana kita dapat mengubah kapasitas dari list saat *runtime*, tidak seperti list statis yang ukurannya sudah ditentukan terlebih dahulu sehingga kapasitas list bersifat tetap selama proses *compile*.

3.5.4 Implementasi sebagai ADT

Implementasi dari ADT ArrayDinWord adalah untuk menampung dan mengelola judul playlist yang dimiliki pengguna. ADT tersebut dapat diakses dengan cara berikut:

```
./src/ADT/List/  
arraydinword.h, arraydinword.c, driverarraydin.c
```

3.6 Struktur Data Mesin Karakter

3.6.1 Sketsa Struktur Data

```
#define MARK '~'  
extern char currentChar;  
extern boolean EOP;
```

Berikut primitif yang digunakan dalam ADT ini:

- a. void START(FILE *input)
- b. void ADV()

3.6.2 Persoalan yang dapat diselesaikan

Struktur ini berperan dalam pembacaan sebuah input baik dari pengguna secara langsung, maupun pembacaan dari sebuah file.

3.6.3 Alasan Pemilihan ADT

Kami memilih untuk menggunakan struktur data Mesin Karakter dengan alasan karena di program WayangWave ini, terdapat pembatasan dalam menerima input dari pengguna, yaitu tidak dapat menggunakan *scanf* ataupun *fgets* ketika memproses *command* dan *file* konfigurasi.

3.6.4 Implementasi sebagai ADT

Penerapan struktur data Mesin Karakter ini terletak pada proses pembacaan isi dari *file* yang berisi *list* penyanyi, album, dan lagu pada *command* START maupun LOAD, namun pada proses ini Mesin Karakter perlu modifikasi tambahan yang akan dilengkapi pada ADT Mesin Kata.

./src/ADT/Mesin/
mesinkarakter.h, mesinkarakter.c

3.7 Struktur Data Mesin Kata

3.7.1 Sketsa Struktur Data

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

Berikut primitif yang digunakan dalam ADT ini:

1. Operasi - operasi
 - a. void IgnoreBlanks()
 - b. void STARTWORD(FILE *input)
 - c. void STARTINPUT()
 - d. void ADVWORD()
 - e. void CopyWord()
2. Selektor
 - a. Word GetWord()
 - b. void GetCommand()
 - c. Word AccessCommand(Word comm, int Idx)
 - d. Word AccessConfig(Word comm, int Idx)
 - e. Word AccessConfig2(Word comm, int Idx)
3. Konverter
 - a. Word toKata(char* str)
 - b. int WordToInt(Word currentWord)
 - c. char *WordToStr(Word kata)

- d. Word intToWord(int n)
- e. Word toLower(Word kata)
- f. Word toUpper(Word kata)
- 4. Identitas
 - a. int stringLength (char* str)
- 5. Tes Kebenaran
 - a. boolean WordCompare(Word currentWord, Word inputWord)
 - b. boolean isEndWord()
- 6. Lain Lain
 - a. void PrintWord(Word kata)
 - b. void concatWord(Word *kata1, Word kata2)

3.7.2 Persoalan yang dapat diselesaikan

Struktur ini berperan dalam menyempurnakan peran Mesin Karakter dalam pembacaan sebuah masukan baik dari pengguna secara langsung, maupun dari *file txt*.

3.7.3 Alasan Pemilihan ADT

Penggunaan ADT ini, masukan dari pengguna dapat diproses lebih lanjut dan lebih mudah, seperti mengidentifikasi masukan *command* dari pengguna dengan menggunakan fungsi WordCompare sehingga proses kondisional pada program utama dapat berjalan dengan lebih mudah.

3.7.4 Implementasi sebagai ADT

Contoh penerapan yaitu pada pembacaan isi dari sebuah *file* dan pembacaan masukan dari pengguna, seperti pembacaan *command* yang diberikan pengguna, dan pembacaan nama penyanyi, nama album, nama lagu dan nama playlist.

./src/ADT/Mesin/
mesinkata.h, mesinkata.c

3.8 Struktur Data Queue

3.8.1 Sketsa Struktur Data

```
typedef struct {
    Song buffer[CAPACITY];
    int idxHead;
    int idxTail;
} Queue;
```

Berikut primitif yang digunakan dalam ADT ini:

1. Konstruktor
 - a. void CreateQueue(Queue *q)
2. Prototype
 - a. boolean queue_isEmpty(Queue q)
 - b. boolean queue_isFull(Queue q)
 - c. int queue_length(Queue q)
 - d. boolean queue_IsMember(Queue q, int a)
3. Insert/Delete
 - a. void enqueue(Queue *q, Song val)
 - b. void queue_insert(Queue *q, Song X)
 - c. void dequeue(Queue *q, Song *val)
 - d. void queue_delln(Queue *q, int a, Song *del)

3.8.2 Persoalan yang dapat diselesaikan

Program Wayang Wave memanfaatkan ADT Queue untuk menyimpan judul-judul lagu yang akan diputar.

3.8.3 Alasan Pemilihan ADT

Penggunaan struktur queue bertujuan untuk merapihkan urutan akses lagu yang diakses pengguna.

3.8.4 Implementasi sebagai ADT

Struktur data queue digunakan untuk menyimpan lagu-lagu yang sedang diputar dan akan diputar. Penyimpanan lagu di dalam queue menggunakan aturan *First In First Out* (FIFO). Penerapannya terdapat di dalam hampir setiap procedure yang ada.

./src/ADT/Queue/
queue.h, queue.c, driverQueue.c

3.9 Struktur Data Stack

3.9.1 Sketsa Struktur Data

```
typedef struct {
    Song T[StackMaxEl];
    int TOP;
} Stack;
```

Berikut primitif yang digunakan dalam ADT ini:

1. Konstruktor
 - a. void CreateEmptyStack(Stack *S)

2. Prototype
 - a. boolean IsEmptyStack(Stack S)
 - b. boolean IsFullStack(Stack S)
 - c. int NbElmtStack(Stack S)
3. Insert/Delete
 - a. void PushStack(Stack * S, Song X)
 - b. void PopStack(Stack * S, Song * X)
 - c. void InverseStack(Stack *Asal, Stack *Tujuan)

3.9.2 Persoalan yang dapat diselesaikan

Struktur data stack digunakan dalam menyimpan riwayat lagu yang sudah diputar.

3.9.3 Alasan Pemilihan ADT

Struktur data ini menerapkan prinsip *Last In First Out* yang sesuai dengan konsep riwayat lagu untuk program WayangWave.

3.9.4 Implementasi sebagai ADT

Contoh penerapan ADT Stack ada pada procedure SongPrevious yang akan memutar lagu yang terakhir dimasukkan ke riwayat lagu. Cara pengaksesan ADT ini adalah sebagai berikut:

```
./src/ADT/Stack/  
stack.h, stack.c, driverstack.c
```

3.10 Struktur Data Set

3.10.1 Sketsa Struktur Data

```
typedef struct {  
    Object buffer[CAPACITY];  
    int length;  
} Set;
```

Berikut primitif yang digunakan dalam ADT ini:

1. Konstruktor
 - a. void SetCreateEmpty(Set *s);
2. Prototype
 - a. boolean SetIsEmpty(Set s)
 - b. boolean SetIsFull(Set s)
 - c. boolean SetIsIn(Set s, Object O)
 - d. boolean SetsIsEqual(Set s1, Set s2)
3. Selektor

- a. `int SearchSet(Set s, Object O)`
- 4. Insert/Delete
 - a. `void SetAdd(Set *s, Object O)`
 - b. `void SetRemoveElmt(Set *s, Object O)`
- 5. Lain Lain
 - a. `Set SetUnion(Set s1, Set s2)`
 - b. `Set SetIntersection(Set s1, Set s2)`
 - c. `Set SetDifference(Set s1, Set s2)`
 - d. `Set SetCopy(Set s)`
 - e. `boolean SetIsSubset(Set s1, Set s2)`
 - f. `void DisplaySet(Set s)`

3.10.2 Persoalan yang dapat diselesaikan

Struktur data ini digunakan pada penginputan nama album, judul-judul lagu, dan nama playlist agar value yang dimasukkan tidak *duplicate* dan dapat dipastikan bahwa tiap elemen di dalamnya tidak identik.

3.10.3 Alasan Pemilihan ADT

Pemilihan struktur data ini disebabkan oleh kebutuhan untuk menyediakan nama album, judul lagu, dan nama playlist yang unik. ADT Object digunakan sebagai isi dari set untuk mencegah permasalahan dua penyanyi yang memiliki album dengan nama yang sama persis.

3.10.4 Implementasi sebagai ADT

Implementasinya dapat dilihat pada struktur data map yang diimplementasikan pada Map Album dan Map Lagu.

`./src/ADT/SetMap/
set.h, set.c`

3.11 Struktur Data Map

3.11.1.1 Sketsa Struktur Data

```
typedef struct {
    Word Key;
    Set Value;
} dictionary;

typedef struct {
    dictionary Elements[MaxEl];
    int Count;
```

```
} Map;
```

Berikut primitif yang digunakan dalam ADT ini:

1. Konstruktor
 - a. void MapCreateEmpty(Map *M)
2. Prototype
 - a. boolean MapIsEmpty(Map M)
 - b. boolean MapIsFull(Map M)
 - c. boolean MapIsMember(Map M, Word k)
3. Selektor
 - a. void MapValue(Map M, Word k, Set *val)
 - b. int searchMap(Map M, Word k)
4. Tes Empty/Full
 - a. void MapInsert(Map *M, Word k, Set v)
 - b. void MapDelete(Map *M, Word k)

3.11.1.2 Persoalan yang dapat diselesaikan

Struktur data ini dapat dapat menyelesaikan permasalahan akses playlist, nama album, dan nama playlist dari sebuah lagu. ADT ini akan menyimpan informasi-informasi yang dibutuhkan dan memudahkan akses untuk menampilkan data lagu atau album ke program WayangWave.

3.11.1.3 Alasan Pemilihan ADT

Dengan menggunakan struktur data ini, akses elemen untuk menampilkan informasi dari lagu dan album menjadi lebih mudah. Penggunaan ADT Set pada *value* juga memastikan tidak ada *value* yang identik.

3.11.1.4 Implementasi sebagai ADT

Implementasinya bisa dilihat pada penggunaan Map Album dan Map Lagu pada direktori berikut:

```
./src/ADT/SetMap/  
map.h, map.c
```

3.12 Struktur List dengan Struktur Berkait

3.12.1.1 Sketsa Struktur Data

```
typedef struct node *address;  
typedef struct node {
```

```

    Song info;
    address next;
} ElmtListLinier;
typedef struct {
    address First;
} ListLinier;

```

Berikut primitif yang digunakan dalam ADT ini:

1. Konstruktor
 - a. void CreateEmptyListLinier(ListLinier *L)
 - b. address AlokasiListLinier(Song X)
2. Destruktor
 - a. void DealokasiListLinier(address *P)
3. Prototype
 - a. boolean IsEmptyListLinier(ListLinier L)
 - b. int NumberElmtListLinier(ListLinier L)
4. Selektor
 - a. address SearchListLinier(ListLinier L, Song X)
 - b. address addrAtListLinier(ListLinier L, int Idx)
 - c. Song getValAtListLinier(ListLinier L, int Idx)
5. Insert/Delete
 - a. void InsVFirstListLinier(ListLinier *L, Song X)
 - b. void InsVLastListLinier(ListLinier *L, Song X)
 - c. void DelVFirstListLinier(ListLinier *L, Song *X)
 - d. void DelVLastListLinier(ListLinier *L, Song *X)
 - e. void InsertFirstListLinier(ListLinier *L, address P)
 - f. void InsertAfterListLinier(ListLinier *L, address P, address Prec)
 - g. void InsertLastListLinier(ListLinier *L, address P)
 - h. void DelFirstListLinier(ListLinier *L, address *P)
 - i. void DelPListLinier(ListLinier *L, Song X)
 - j. void DelLastListLinier(ListLinier *L, address *P)
 - k. void DelAfterListLinier(ListLinier *L, address *Pdel, address Prec)
 - l. void setValAtListLinier(ListLinier *L, int Idx, Song set)
 - m. void DelAtListLinier(ListLinier *L, int Idx, Song *del)
6. Lain Lain
 - a. void PrintInfoListLinier(ListLinier L)
 - b. void InversListLinier(ListLinier *L)
 - c. void KonkatListLinier(ListLinier *L1, ListLinier *L2, ListLinier *L3)

3.12.2 Persoalan yang dapat diselesaikan

ADT list linier digunakan untuk merepresentasikan kumpulan playlist secara kontigu.

3.12.3 Alasan Pemilihan ADT

Struktur dari ADT ini sangat cocok digunakan untuk merepresentasikan spesifikasi dari sebuah playlist pengguna.

3.12.4 Implementasi sebagai ADT

Implementasinya dapat dilihat pada ADT ArrayDin dengan contoh aplikasi di bagian PlaylistData dari pengguna. Pemanggilan fungsi ini menggunakan perintah berikut,

```
./src/ADT/List Berkait/  
listlinier.h, listlinier.c, driverlistberkait.c
```

3.13 Struktur ADT User

3.13.1 Sketsa Struktur Data

```
#define maxuser 20  
typedef struct {  
    Word Username;  
    Song Playing;  
    Queue Antrian;  
    Stack Riwayat;  
    ArrayDinWord PlaylistTitle;  
    ArrayDin PlaylistData;  
} UserData;  
typedef struct {  
    UserData storage[maxuser];  
    int count;  
} UserStorage;
```

Berikut primitif yang digunakan dalam ADT ini:

1. void initiateUserStorage(UserStorage *User)
2. int createNewUser(UserStorage *User, Word Username)
3. int searchUser(UserStorage *User, Word Username)

3.13.2 Persoalan yang dapat diselesaikan

ADT ini digunakan untuk merepresentasikan penyimpanan data pengguna untuk fitur bonus, yaitu *MultiUser*.

3.13.3 Alasan Pemilihan ADT

ADT ini dibuat untuk menyederhanakan proses menambah dan mengakses data user. ADT ini dirancang sehingga ketika penambahan prosedur fitur bonus, perubahan struktur data hanya terjadi di `main.c`

3.13.4 Implementasi sebagai ADT

Implementasinya bisa dilihat pada penggunaan User pada direktori berikut:
`./src/ADT/User/ user.h, user.c`

4 Program Utama

Program utama ini dapat diakses melalui file `src/main.c`. Di dalam program `main.c` terdapat sebuah procedure `userManagement` yang berguna untuk menjalankan perintah register dan login untuk fitur bonus *multiUser*. Selanjutnya, di bagian *main* program akan menampilkan *banner* WayangWave yang menandakan sesi mulai berjalan. Di awal memasuki sesi, pengguna akan diarahkan untuk memasukkan *command* START atau LOAD. Jika memasukkan *command* START, maka secara otomatis program akan membaca file `config.txt`. Jika memasukkan *command* LOAD, maka pengguna perlu memasukkan file yang akan dibaca, contohnya adalah file `default.txt` yang berisi *state* program dari sesi sebelumnya yang disimpan.

Setelah melakukan START atau LOAD, *command* selanjutnya yang bisa dilakukan ialah,

- LIST
 - LIST DEFAULT
 - LIST PLAYLIST
- PLAY
 - PLAY SONG
 - PLAY PLAYLIST
- QUEUE
 - QUEUE SONG
 - QUEUE PLAYLIST
 - QUEUE SWAP <X> <Y>
 - QUEUE REMOVE <id>
 - QUEUE CLEAR
- SONG
 - SONG NEXT
 - SONG PREVIOUS
- PLAYLIST
 - PLAYLIST CREATE
 - PLAYLIST ADD
 - PLAYLIST SWAP <id> <X> <Y>
 - PLAYLIST REMOVE <id> <n>
 - PLAYLIST DELETE
- STATUS
- SAVE <filename>
- QUIT
- HELP

Perlu diingat bahwa procedure Queue Playlist, Play Playlist, dan fitur Playlist hanya bisa dijalankan ketika pengguna memiliki playlist yang valid. Di akhir program, yaitu ketika

memasukkan *command* Quit, pengguna bisa menyimpan *state* dari program dengan memanggil *procedure* save.

5 Algoritma-Algoritma Menarik

5.1 Enhance

Algoritma ini digunakan sebagai penunjang dari fitur bonus, yaitu ENHANCE. Command ENHANCE digunakan untuk menambahkan lagu pada playlist pengguna. Algoritma ini dianggap menarik karena penambahan lagu tidak hanya menggunakan fungsi random, melainkan menggunakan data seperti lagu yang mirip terhadap lagu lainnya pada playlist atau lagu yang mirip terhadap lagu yang pernah dimainkan. Ketika pengguna melakukan ENHANCE pada suatu playlist, WayangWave akan merekomendasikan dan menambahkan lagu rekomendasi berdasarkan lagu yang ada pada playlist tersebut (lagu dengan album yang sama pada playlist tersebut) bila playlist tersebut tidak kosong. Jika playlist tersebut kosong dan terdapat lagu pada Stack Riwayat, WayangWave akan merekomendasikan dan menambahkan lagu rekomendasi berdasarkan lagu yang pernah dimainkan (lagu dengan album yang sama pada Stack Riwayat), dan apabila playlist tersebut dan Stack Riwayat pengguna kosong, WayangWave akan merekomendasikan dan menambahkan lagu random.

Gambar 5.1.1 Menambahkan lagu berdasarkan lagu yang ada pada playlist tersebut (lagu dengan album yang sama pada playlist tersebut)

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

Masukkan Nama Playlist yang akan di-enhance : BLACKPINK My Top Three
Lagu yang berada pada playlist "BLACKPINK My Top Three":
  1. BLACKPINK - How You Like That - THE ALBUM
  2. BLACKPINK - Typa Girl - BORN PINK
  3. BLACKPINK - Pink Venom - BORN PINK
Menambahkan Lagu pada Playlist berdasarkan lagu yang berada pada playlist tersebut
Lagu yang ditambahkan:
  4. BLACKPINK - Ice Cream (with Selena Gomez) - THE ALBUM
  5. BLACKPINK - Bet You Wanna (Feat. Cardi B) - THE ALBUM
  6. BLACKPINK - Ready For Love - BORN PINK
```

Gambar 5.1.2 Menambahkan lagu berdasarkan lagu yang pernah dimainkan (lagu dengan album yang sama pada Stack Riwayat)

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

Masukkan Nama Playlist yang akan di-enhance : Mixed Playlist
Lagu yang berada pada playlist "Mixed Playlist":
Playlist Kosong!
Menambahkan Lagu pada Playlist berdasarkan lagu yang pernah dimainkan
Lagu yang ditambahkan:
    1. BLACKPINK - Pink Venom - BORN PINK
    2. BLACKPINK - Shut Down - BORN PINK
    3. BLACKPINK - Typa Girl - BORN PINK
```

Gambar 5.1.3 Menambahkan lagu random pada playlist kosong

```
>> ENTER COMMAND: ENHANCE
Daftar Playlist Pengguna :
    1. BLACKPINK My Top Three
    2. Arctic Monkeys My Top 5
    3. Mixed Playlist
    4. Random Songs

Masukkan Nama Playlist yang akan di-enhance : Random Songs
Lagu yang berada pada playlist "Random Songs":
Playlist Kosong!
Menambahkan Lagu pada Playlist dengan lagu random
Lagu yang ditambahkan:
    1. Arctic Monkeys - Bet You Wanna (Feat. Cardi B) - THE ALBUM
    2. BLACKPINK - Bet You Wanna (Feat. Cardi B) - THE ALBUM
    3. Arctic Monkeys - Typa Girl - BORN PINK
```

6 Data Test

Program WayangWave ini terdiri dari beberapa fitur atau *command* yang dapat diberikan oleh pengguna. Oleh karena itu, diperlukan *testing* pada beberapa kondisi untuk memastikan apakah tiap fitur tersebut dapat berjalan dengan tepat dan benar. Berikut adalah *testing* yang kami lakukan terhadap *command-command* yang terdapat pada program WayangWave ini, beserta dengan penjelasan mengenai hasil atau *output* yang akan diberikan oleh program pada tiap kondisinya.

6.1 Data Test START

Pada awal menjalankan program, pengguna WayangWave diharuskan memilih opsi antara START, LOAD, atau HELP untuk memulai program. Ketika pengguna memilih START, maka program akan berhasil dijalankan.

Gambar 6.1.1 Tampilan command START saat dijalankan

```
>> ENTER COMMAND: START  
File konfigurasi sistem berhasil dibaca. WayangWave berhasil dijalankan.
```

6.2 Data Test LOAD

Pada awal menjalankan program, selain memulai WayangWave dengan menggunakan *command* START atau HELP, pengguna juga memiliki opsi untuk memulai WayangWave dengan LOAD. *Command* ini harus diikuti dengan nama file yang ingin dibuka. Ketika pengguna memberikan *command* LOAD, *file* yang dapat berasal dari suatu *state* yang sudah pernah disimpan oleh pengguna. Saat diikuti nama *file* yang valid, program akan menjalankan aplikasi setelah membaca *file* yang telah di-save. Apabila pengguna memberikan nama *file* yang tidak valid atau dengan format selain '.txt' atau '.TXT', program akan menampilkan pesan kesalahan dan program tidak dapat dijalankan.

Gambar 6.3.1 Tampilan command LOAD saat berhasil dijalankan

```
>> ENTER COMMAND: LOAD default.txt  
Save file berhasil dibaca. WayangWave berhasil dijalankan.
```

Gambar 6.3.2 Tampilan command LOAD saat save file tidak ditemukan atau salah ketik

```
>> ENTER COMMAND: LOAD false.txt  
Save file tidak ditemukan. WayangWave gagal dijalankan.
```

6.3 Data Test LIST DEFAULT

Pengguna program dapat memanggil *command* LIST DEFAULT untuk melihat *list* penyanyi yang ada. Lalu, pengguna dapat memilih untuk melihat album dari penyanyi yang dipilih. Pengguna juga diberikan pilihan untuk melihat daftar lagu yang ada di album yang dipilih oleh pengguna.

Gambar 6.4.1 Tampilan command LIST DEFAULT saat pengguna ingin melihat album dan lagu

```
>> ENTER COMMAND: 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 BLACKPINK oleh BLACKPINK :

  1. Pink Venom
  2. Shut Down
  3. Typa Girl
  4. Ready For Love
```

Gambar 6.4.2 Tampilan command LIST DEFAULT saat pengguna tidak ingin melihat album

```
>> ENTER COMMAND: LIST DEFAULT
Daftar Penyanyi :
  1. BLACKPINK
  2. Arctic Monkeys

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

6.4 Data Test LIST PLAYLIST

Pengguna program dapat memanggil *command* LIST PLAYLIST untuk menampilkan *playlist* yang dimiliki pengguna.

Gambar 6.5.1 Tampilan command LIST PLAYLIST saat pengguna memiliki playlist

```
>> ENTER COMMAND: LIST PLAYLIST

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

Gambar 6.5.2 Tampilan command LIST PLAYLIST saat pengguna tidak memiliki playlist

```
>> ENTER COMMAND: LIST PLAYLIST

Daftar playlist yang kamu miliki:
Kamu tidak memiliki playlist.
```

6.5 Data Test PLAY SONG

Pengguna dapat memanggil *command* PLAY SONG untuk memainkan lagu berdasarkan masukan nama penyanyi, nama album, dan id lagu. Saat *command* ini berhasil dieksekusi, *queue* dan riwayat lagu akan menjadi kosong.

6.6.1 Tampilan command PLAY SONG saat dijalankan

```
>> ENTER COMMAND: PLAY SONG
Daftar Penyanyi :
  1. BLACKPINK
  2. Arctic Monkeys

Masukkan Nama Penyanyi yang dipilih : BLACKPINK

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

Masukkan Judul Album yang dipilih : BORN PINK
Daftar Lagu Album BLACKPINK oleh BLACKPINK :

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

Masukkan ID Lagu yang dipilih : 1
Memutar lagu "Pink Venom" oleh "BLACKPINK"
```

6.6 Data Test PLAY PLAYLIST

Pengguna dapat memanggil *command* PLAY PLAYLIST untuk memainkan lagu berdasarkan id playlist. Saat dieksekusi, *current song* akan menjadi lagu pada urutan pertama *playlist* dan *queue* berisi semua lagu yang ada di dalam *playlist*.

6.7.1 Tampilan command PLAY PLAYLIST saat dijalankan

```
>> ENTER COMMAND: PLAY PLAYLIST

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

Masukkan ID Playlist yang dipilih : 1
Berhasil menambahkan playlist "BLACKPINK My Top Three" ke queue.
```

6.7 Data Test QUEUE SONG

Pengguna dapat memanggil *command* QUEUE SONG untuk menambahkan lagu ke dalam *queue*, berdasarkan nama penyanyi, nama album, dan id lagu yang dimasukkan ke dalam *queue*.

6.8.1 Tampilan command *QUEUE SONG* saat dijalankan

```
>> ENTER COMMAND: QUEUE SONG
Daftar Penyanyi :
  1. BLACKPINK
  2. Arctic Monkeys

Masukkan Nama Penyanyi yang dipilih : BLACKPINK

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

Masukkan Judul Album yang dipilih : THE ALBUM
Daftar Lagu Album Arctic Monkeys oleh BLACKPINK :
  1. How You Like That
  2. Ice Cream (with Selena Gomez)
  3. Bet You Wanna (Feat. Cardi B)

Masukkan ID Lagu yang dipilih : 3
Berhasil menambahkan lagu "Bet You Wanna (Feat. Cardi B)" oleh "BLACKPINK" ke queue.
```

6.8 Data Test *QUEUE PLAYLIST*

Pengguna dapat menggunakan *command QUEUE PLAYLIST* untuk menambahkan lagu yang ada di dalam *playlist* ke dalam *queue* dengan memasukkan id *playlist*-nya.

6.9.1 Tampilan command *QUEUE PLAYLIST* saat dijalankan

```
>> ENTER COMMAND: QUEUE PLAYLIST

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

Masukkan ID Playlist yang dipilih : 3
Berhasil menambahkan playlist "Mixed Playlist" ke queue.
```

6.9 Data Test *QUEUE SWAP*

Pengguna dapat menggunakan *command QUEUE SWAP* untuk menukar lagu pada urutan tertentu. Apabila urutannya tidak ada dalam *queue* program akan menampilkan pesan tidak valid.

6.10.1 Tampilan command *QUEUE SWAP* saat berhasil dijalankan

```
>> ENTER COMMAND: QUEUE SWAP 1 2
Lagu "Typa Girl" berhasil ditukar dengan "How You Like That"
```

6.10.2 Tampilan command *QUEUE SWAP* saat lagu dengan urutan tertentu tidak ditemukan

```
>> ENTER COMMAND: QUEUE SWAP 70 80
Lagu dengan urutan ke 70 dan 80 tidak terdapat dalam queue!
```

6.10 Data Test QUEUE REMOVE

Pengguna dapat menggunakan *command* QUEUE REMOVE untuk menghapus lagu dari *queue* dengan masukan id lagu yang ingin dihapus. Apabila id tidak ada dalam *queue*, maka program akan menampilkan pesan tidak valid.

6.11.1 Tampilan command QUEUE REMOVE saat berhasil dijalankan

```
>> ENTER COMMAND: QUEUE REMOVE 2  
Lagu "How You Like That" oleh "BLACKPINK" telah dihapus dari queue!
```

6.11.2 Tampilan command QUEUE REMOVE saat lagu dengan urutan tertentu tidak ditemukan

```
>> ENTER COMMAND: QUEUE REMOVE 30  
Lagu dengan urutan ke 30 tidak ada.
```

6.11 Data Test QUEUE CLEAR

Pengguna dapat menggunakan *command* QUEUE CLEAR untuk mengosongkan *queue*.

6.12.1 Tampilan command QUEUE CLEAR saat dijalankan

```
>> ENTER COMMAND: QUEUE CLEAR  
Queue berhasil dikosongkan.
```

6.12 Data Test SONG NEXT

Pengguna dapat menggunakan *command* SONG NEXT untuk memutar lagu yang beads di dalam *queue*. Lagu yang sedang diputar, akan ditambahkan ke dalam daftar riwayat putar lagu. Apabila *queue* kosong, yang diputar adalah lagu yang sedang diputar.

6.13.1 Tampilan command SONG NEXT saat ada lagu selanjutnya pada queue

```
>> ENTER COMMAND: SONG NEXT  
Memutar lagu selanjutnya  
"Typa Girl" oleh "BLACKPINK"
```

6.13.2 Tampilan command SONG NEXT saat queue kosong

```
>> ENTER COMMAND: SONG NEXT  
Queue kosong, memutar kembali lagu  
"Typa Girl" oleh "BLACKPINK"
```

6.13 Data Test SONG PREVIOUS

Pengguna dapat menggunakan *command SONG PREVIOUS* untuk memutar lagu yang terakhir kali diputar lalu ditambah ke dalam *queue* dengan urutan pertama. Apabila *queue* riwayat lagu kosong, yang diputar adalah lagu yang sedang diputar.

6.14.1 Tampilan *command SONG PREVIOUS* saat pengguna pernah memutar lagu sebelumnya

```
>> ENTER COMMAND: SONG PREVIOUS
Memutar lagu sebelumnya
"Typa Girl" oleh "BLACKPINK"
```

6.14.2 Tampilan *command SONG PREVIOUS* saat pengguna belum pernah memutar lagu

```
>> ENTER COMMAND: SONG PREVIOUS
Riwayat lagu kosong, memutar kembali lagu
"Pink Venom" oleh "BLACKPINK"
```

6.14 Data Test PLAYLIST CREATE

Pengguna dapat menggunakan *command PLAYLIST CREATE* untuk membuat *playlist* baru dan ditambahkan pada daftar *playlist* pengguna. Awalnya *playlist* kosong dan nama *playlist* dapat sama dengan *playlist* yang sudah ada.

6.15.1 Tampilan *command PLAYLIST CREATE* saat berhasil dijalankan

```
>> ENTER COMMAND: PLAYLIST CREATE
Masukkan nama playlist yang ingin dibuat : TEST
Playlist TEST berhasil dibuat! Silakan masukkan lagu - lagu artis terkini kesayangan Anda!
```

6.15.2 Tampilan *command PLAYLIST CREATE* saat sudah ada *playlist* dengan nama yang sama

```
>> ENTER COMMAND: PLAYLIST CREATE
Masukkan nama playlist yang ingin dibuat : TEST
Sudah ada Playlist dengan nama tersebut! Silakan coba lagi.
Masukkan nama playlist yang ingin dibuat : test
Playlist test berhasil dibuat! Silakan masukkan lagu - lagu artis terkini kesayangan Anda!
```

6.15.3 Tampilan *command PLAYLIST CREATE* saat nama *playlist* tidak memenuhi syarat

```
>> ENTER COMMAND: PLAYLIST CREATE
Masukkan nama playlist yang ingin dibuat :
Minimal terdapat 3 karakter selain whitespace dalam nama playlist. Silakan coba lagi.
Masukkan nama playlist yang ingin dibuat : testing
Playlist testing berhasil dibuat! Silakan masukkan lagu - lagu artis terkini kesayangan Anda!
```

6.15 Data Test PLAYLIST ADD

Pengguna dapat menggunakan *command PLAYLIST ADD* untuk menambahkan lagu pada suatu *playlist* yang telah ada sebelumnya pada daftar *playlist* pengguna. Lagu-lagu yang ditambahkan adalah lagu-lagu yang belum ada pada *playlist* pengguna. Pesan *error* akan ditampilkan apabila masukkan pengguna tidak valid.

6.16.1 Tampilan command *PLAYLIST ADD* saat pengguna berhasil menambahkan lagu ke playlist

```
>> ENTER COMMAND: PLAYLIST ADD SONG
Daftar Penyanyi :
  1. BLACKPINK
  2. Arctic Monkeys

Masukkan Nama Penyanyi yang dipilih : BLACKPINK

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

Masukkan Judul Album yang dipilih : BORN PINK
Daftar Lagu Album BORN PINK oleh BLACKPINK :
  1. Pink Venom
  2. Shut Down
  3. Typa Girl
  4. Ready For Love

Masukkan ID Lagu yang dipilih : 1

Daftar Playlist Pengguna :
  1. TEST

Masukkan ID Playlist yang dipilih : 1
Lagu dengan judul "Pink Venom" pada album BORN PINK oleh penyanyi BLACKPINK berhasil ditambahkan ke dalam playlist TEST.
```

6.16.2 Tampilan command *PLAYLIST ADD* saat pengguna berhasil menambahkan album ke playlist

```
>> ENTER COMMAND: PLAYLIST ADD ALBUM
Daftar Penyanyi :
  1. BLACKPINK
  2. Arctic Monkeys

Masukkan Nama Penyanyi yang dipilih : BLACKPINK

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

Masukkan Judul Album yang dipilih : BORN PINK

Daftar Playlist Pengguna :
  1. TEST

Masukkan ID Playlist yang dipilih : 1
Album dengan judul "BORN PINK" berhasil ditambahkan ke dalam pada playlist pengguna "TEST".
```

6.16.3 Tampilan command *PLAYLIST ADD* saat pengguna salah mengetik nama penyanyi

```
>> ENTER COMMAND: PLAYLIST ADD SONG
Daftar Penyanyi :
  1. BLACKPINK
  2. Arctic Monkeys

Masukkan Nama Penyanyi yang dipilih : BLACKPIN
Penyanyi BLACKPIN tidak ada dalam daftar. Silakan coba lagi.
```

6.16.4 Tampilan command *PLAYLIST ADD* saat pengguna salah mengetik nama album

```
>> ENTER COMMAND: PLAYLIST ADD ALBUM
Daftar Penyanyi :
  1. BLACKPINK
  2. Arctic Monkeys

Masukkan Nama Penyanyi yang dipilih : BLACKPINK

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

Masukkan Judul Album yang dipilih : BORN PI
Album BORN PI tidak ada dalam daftar. Silakan coba lagi.
```

6.16 Data Test **PLAYLIST SWAP**

Pengguna dapat menggunakan *command* PLAYLIST SWAP untuk menukar lagu sesuai urutan yang diinginkan di *playlist*. Apabila lagu tidak ada di dalam *playlist*, maka program akan menampilkan pesan *error*.

6.17.1 Tampilan command PLAYLIST SWAP saat berhasil dijalankan

```
>> ENTER COMMAND: PLAYLIST SWAP 1 1 2  
Berhasil menukar lagu dengan nama "Pink Venom" dengan "Shut Down" di playlist "TEST"
```

6.17.2 Tampilan command PLAYLIST SWAP saat playlist tidak ditemukan

```
>> ENTER COMMAND: PLAYLIST SWAP 500 1 2  
Tidak ada playlist dengan ID 500.
```

6.17.3 Tampilan command PLAYLIST SWAP saat lagu tidak ditemukan

```
>> ENTER COMMAND: PLAYLIST SWAP 1 900 2  
Tidak ada lagu dengan urutan 900 di playlist "TEST"
```

6.17 Data Test **PLAYLIST REMOVE**

Pengguna dapat menggunakan *command* PLAYLIST REMOVE untuk menghapus lagu dengan info urutannya pada *playlist* yang dipilih sesuai id.

6.18.1 Tampilan command PLAYLIST REMOVE saat berhasil dijalankan

```
>> ENTER COMMAND: PLAYLIST REMOVE 1 2  
Lagu "Shut Down" oleh "BLACKPINK" telah dihapus dari playlist "TEST"!
```

6.18.2 Tampilan command PLAYLIST REMOVE saat playlist tidak ditemukan

```
>> ENTER COMMAND: PLAYLIST REMOVE 100 2  
Tidak ada playlist dengan ID 100.
```

6.18.3 Tampilan command PLAYLIST REMOVE saat lagu tidak ditemukan

```
>> ENTER COMMAND: PLAYLIST REMOVE 1 40  
Tidak ada lagu dengan urutan 40 di playlist "TEST"
```

6.18 Data Test **PLAYLIST DELETE**

Pengguna dapat menggunakan *command* PLAYLIST DELETE untuk menghapus *playlist* yang ada dalam daftar *playlist* pengguna. Program akan menampilkan pesan *error* apabila masukkan pengguna tidak valid setiap permintaan masukkan.

6.19.1 Tampilan command *PLAYLIST DELETE* saat berhasil dijalankan

```
>> ENTER COMMAND: PLAYLIST DELETE

Daftar Playlist Pengguna :
    1. TEST

Masukkan ID Playlist yang dipilih : 1
Playlist ID 1 dengan judul TEST berhasil dihapus.
```

6.19.2 Tampilan command *PLAYLIST DELETE* saat playlist tidak ditemukan

```
>> ENTER COMMAND: PLAYLIST DELETE

Daftar Playlist Pengguna :
    1. TEST

Masukkan ID Playlist yang dipilih : 2
Tidak ada playlist dengan ID 2 dalam daftar playlist pengguna. Silakan coba lagi.
```

6.19 Data Test *STATUS*

Pengguna dapat menggunakan *command STATUS* untuk menampilkan lagu yang sedang dimainkan beserta *queue song* yang ada dan informasi *playlist* lagu yang diputar.

6.20.1 Tampilan command *STATUS* saat ada lagu yang sedang diputar dan *queue* terisi

```
>> ENTER COMMAND: STATUS
Now Playing:
Arctic Monkeys - R U Mine? - AM

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

6.20.2 Tampilan command *STATUS* saat ada lagu yang sedang diputar dari *playlist* dan *queue* terisi

```
>> ENTER COMMAND: STATUS
Current Playlist: testing

Now Playing:
BLACKPINK - Pink Venom - BORN PINK

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

6.20.3 Tampilan command STATUS saat tidak ada lagu yang sedang diputar dan queue kosong

```
>> ENTER COMMAND: STATUS
Now Playing:
No songs have been played yet. Please search for a song to begin playback.

Queue:
Your queue is empty.
```

6.20 Data Test SAVE

Pengguna dapat menggunakan *command* SAVE untuk menyimpan update yang dilakukan pada aplikasi ke dalam suatu file, baik file baru atau file yang sudah tercipta sebelumnya.

6.21.1 Tampilan command SAVE saat berhasil dijalankan

```
>> ENTER COMMAND: save newConfig.txt
Save file berhasil disimpan.
```

6.21 Data Test QUIT

Pengguna dapat menggunakan *command* QUIT untuk keluar dari aplikasi WayangWave dengan pilihan apakah pengguna ingin menyimpan data sesi sekarang atau tidak.

6.22.1 Tampilan command QUIT saat pengguna ingin menyimpan datanya

```
>> ENTER COMMAND: QUIT
Apakah kamu ingin menyimpan data sesi sekarang (Y/N)? Y
Lokasi penyimpanan data sesi sekarang (contoh: config.txt): save.txt
```

6.22.2 Tampilan command QUIT saat pengguna tidak ingin menyimpan datanya

```
>> ENTER COMMAND: QUIT
Apakah kamu ingin menyimpan data sesi sekarang (Y/N)? N
Kamu keluar dari WayangWave.
Dadah ^ _ ^/
```

6.22 Data Test HELP

Pengguna dapat menggunakan *command* HELP untuk menampilkan daftar *command* yang mungkin untuk dieksekusi dengan deskripsinya. Setelah berhasil memasuki program, daftar *command* yang tersedia adalah sebagai berikut.

6.23.1 Tampilan saat pengguna belum masuk ke state START

```
>> ENTER COMMAND: help
====[ Menu Help WayangWave ]====
1. START -> Untuk masuk sesi baru
2. LOAD -> Untuk memulai sesi berdasarkan file konfigurasi
```

6.23.2 Tampilan saat pengguna sudah masuk ke state START (sesudah multi-user)

```
>> ENTER COMMAND: help
====[ Menu Help WayangWave ]====
1. REGISTER -> Untuk mendaftar sebagai pengguna baru
2. LOGIN -> Untuk masuk sebagai pengguna terdaftar
3. QUIT -> Untuk keluar dari sesi
```

6.23.3 Tampilan saat pengguna sudah masuk ke state START

```
>> ENTER COMMAND: help
====[ Menu Help WayangWave ]====
1. LIST -> Untuk menampilkan daftar lagu, playlist...
2. PLAY -> Untuk memulai suatu lagu
3. QUEUE -> Untuk menampilkan antrian lagu
4. SONG -> Untuk menampilkan lagu
5. PLAYLIST -> Untuk menampilkan playlist
6. STATUS -> Untuk menampilkan status
7. SAVE -> Untuk menyimpan state ke dalam file
8. LOGOUT -> Untuk keluar dari sesi pengguna
9. QUIT -> Untuk keluar dari sesi
```

6.23 Data Test <INVALID COMMAND>

Jika pengguna memasukkan command selain yang sudah dirincikan di atas (termasuk kesalahan dalam pengetikan), akan muncul output *error* sebagai berikut:

6.24.1 Tampilan saat pengguna memasukkan command yang tidak tersedia

```
>> ENTER COMMAND: TUBES ALSTRUKDAT
Command tidak bisa dieksekusi!
```

6.24.2 Tampilan saat pengguna memasukkan nama file yang tidak tersedia untuk di-load

```
>> ENTER COMMAND: load nonexistent.txt
Save file tidak ditemukan. WayangWave gagal dijalankan.

Save file gagal dibaca. Silahkan coba lagi.
```

6.24.3 Tampilan saat pengguna memasukkan command lain sebelum mengakses START atau LOAD

```
>> ENTER COMMAND: LIST DEFAULT
Command tidak bisa dieksekusi!
```

7 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah - Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Start	Membaca file konfigurasi	<i>command</i> START	START	Gambar 6.1.1	Sesuai dengan hasil yang diharapkan
2	Load	Menjalankan file yang ada	<i>command</i> LOAD diikuti dengan nama file yang valid dengan format txt	LOAD <nama file>.txt	Gambar 6.3.1, Gambar 6.3.2	Sesuai dengan hasil yang diharapkan
3	List Default	Melihat list penyanyi, album dan lagu yang diinginkan	<i>command</i> LIST DEFAULT diikuti konfirmasi Y/N	LIST DEFAULT	Gambar 6.4.1, Gambar 6.4.2	Sesuai dengan hasil yang diharapkan

4	List Playlist	Menampilkan daftar playlist yang ada	Memberikan <i>command</i> LIST PLAYLIST	LIST PLAYLIST	Gambar 6.5.1, Gambar 6.5.2	Sesuai dengan hasil yang diharapkan
5	Play Song	Memainkan lagu	Memberikan <i>command</i> PLAY SONG lalu memasukkan nama penyanyi, album, dan id lagu yang dipilih	PLAY SONG	Gambar 6.6.1	Sesuai dengan hasil yang diharapkan
6	Play Playlist	Memainkan lagu pertama di playlist yang dipilih dan mengubah queue lagu sesuai dengan playlist	Memberikan <i>command</i> PLAY PLAYLIST lalu memasukkan id playlist yang dipilih	PLAY PLAYLIST	Gambar 6.7.1	Sesuai dengan hasil yang diharapkan
7	Queue Song	Menambahkan lagu ke dalam queue	Memberikan <i>command</i> QUEUE SONG lalu memasukkan nama penyanyi, album, dan id lagu yang dipilih	QUEUE SONG	Gambar 6.8.1	Sesuai dengan hasil yang diharapkan
8	Queue Playlist	Menambahkan lagu dari playlist yang dipilih ke dalam queue	Memberikan <i>command</i> PLAY PLAYLIST lalu memasukkan id playlist yang dipilih	QUEUE PLAYLIST	Gambar 6.9.1	Sesuai dengan hasil yang diharapkan
9	Queue Swap	Menukar urutan lagu dalam queue	Memberikan <i>command</i> QUEUE SWAP dan urutan lagu	QUEUE SWAP <X> <Y>	Gambar 6.10.1, Gambar 6.10.2	Sesuai dengan hasil yang diharapkan
9	Queue Remove	Menghapus lagu dari queue	Memberikan <i>command</i> QUEUE REMOVE dan id playlist yang dipilih	QUEUE REMOVE <id>	Gambar 6.11.1, Gambar 6.11.2	Sesuai dengan hasil yang diharapkan
10	Queue Clear	Mengosongkan queue	Memberikan <i>command</i> QUEUE CLEAR	QUEUE CLEAR	Gambar 6.12.1	Sesuai dengan hasil yang diharapkan
11	Song Next	memutar lagu selanjutnya dalam queue	Memberikan <i>command</i> SONG NEXT	SONG NEXT	Gambar 6.13.1, Gambar 6.13.2	Sesuai dengan hasil yang diharapkan
12	Song Previous	memutar lagu sebelumnya dari queue	Memberikan <i>command</i> SONG PREVIOUS	SONG PREVIOUS	Gambar 6.14.1, Gambar 6.14.2	Sesuai dengan hasil yang diharapkan
13	Playlist Create	Membuat playlist baru dengan ketentuan nama khusus	Memberikan <i>command</i> PLAYLIST CREATE dan nama playlist	PLAYLIST CREATE	Gambar 6.15.1, Gambar 6.15.2, Gambar 6.15.3	Sesuai dengan hasil yang diharapkan
14	Playlist Add	Menambahkan lagu/ album ke dalam ke dalam playlist pengguna	Memberikan <i>command</i> PLAYLIST ADD SONG untuk menambahkan lagu dan <i>command</i>	PLAYLIST ADD SONG PLAYLIST ADD ALBUM	Gambar 6.16.1, Gambar 6.16.2, Gambar 6.16.3, Gambar 6.16.4	Sesuai dengan hasil yang diharapkan

			PLAYLIST ADD ALBUM untuk menambahkan album			
15	Playlist Swap	Menukar urutan lagu di playlist pengguna	Memberikan <i>command</i> PLAYLIST SWAP, id playlist, dan urutan lagu	PLAYLIST SWAP <id> <X> <Y>	Gambar 6.17.1, Gambar 6.17.2, Gambar 6.17.3	Sesuai dengan hasil yang diharapkan
16	Playlist Remove	Menghapus lagu dengan id tertentu dari playlist	Memberikan <i>command</i> QUEUE REMOVE dan id playlist	PLAYLIST REMOVE <id> <n>	Gambar 6.18.1, Gambar 6.18.2, Gambar 6.18.3	Sesuai dengan hasil yang diharapkan
17	Playlist Delete	Menghapus playlist yang tersedia	Memberikan <i>command</i> PLAYLIST DELETE dan playlist yang ingin dihapus	PLAYLIST DELETE	Gambar 6.19.1, Gambar 6.19.2	Sesuai dengan hasil yang diharapkan
18	Status	Menampilkan lagu yang sedang dimainkan dan queue yang ada	Memberikan <i>command</i> STATUS	STATUS	Gambar 6.20.1, Gambar 6.20.2, Gambar 6.20.3	Sesuai dengan hasil yang diharapkan
19	Save	Menyimpan state aplikasi ke dalam suatu file	Memberikan <i>command</i> SAVE diikuti nama file lokasi penyimpanan	SAVE <nama file>.txt	Gambar 6.21.1	Sesuai dengan hasil yang diharapkan
20	Quit	Keluar dari aplikasi WayangWave	Memberikan <i>command</i> QUIT dan konfirmasi untuk menyimpan data	QUIT	Gambar 6.22.1, Gambar 6.22.2	Sesuai dengan hasil yang diharapkan
21	Help	Menampilkan daftar command selanjutnya yang mungkin	Memberikan <i>command</i> HELP	HELP	Gambar 6.23.1, Gambar 6.23.2, Gambar 6.23.3	Sesuai dengan hasil yang diharapkan

8 Pembagian Kerja dalam Kelompok

No.	Nama Anggota - NIM	Deskripsi Kontribusi
1.	Bobby Dhammawan Tang - 18222037	ADT Set & Map, Parsing Config, Load, Save & Quit
2.	Jihan Aurelia - 18222001	ADT Queue, procedure Queue dan Song, laporan bagian Daftar Isi, Test Script, Struktur data (ADT), Program Utama, dan testing program
3.	Benedicta Eryka Santosa - 18222031	ADT List Statis, debug command List dan Play, laporan bagian Daftar Isi, Struktur Data, Test Script, memberi keterangan gambar, menghubungi asisten, dan

		membuat form asistensi
4.	Firsa Athaya Raissa Alifah - 18222051	ADT Mesin Karakter dan Mesin Kata, Parsing command, Quit, Help, Invalid Command, Laporan bagian Ringkasan, Fitur Tambahan, ADT, Data Test, dan Lampiran
5.	Muhammad Faiz A - 18222063	ADT List Dinamis, ADT List Berkait, ADT tambahan seperti Song, Object, UserData & UserStorage, Command Playlist, Command Status, Memperbaiki dan menyatukan Command-Command dari setiap anggota, Fitur Bonus Enhance & Multi User, Laporan bagian Fitur Tambahan Spesifik, Algoritma-Algoritma Menarik & Struktur Data (ADT)

9 Lampiran

9.1 Deskripsi Tugas Besar

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 Mechanics

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.

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](#).

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.

2. **LIST PLAYLIST**

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

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.

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*.

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.

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.

3. QUEUE SWAP <x> <y>

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

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.

5. QUEUE CLEAR

Command QUEUE CLEAR digunakan untuk mengosongkan queue.

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.

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.

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.

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.

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**.

4. PLAYLIST REMOVE

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

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.

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.

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.

j. QUIT

QUIT merupakan command yang digunakan untuk keluar dari aplikasi WayangWave.

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.

l. <INVALID COMMAND>

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

9.2 Notulen Rapat

Form Asistensi Tugas Besar

IF2111/Algoritma dan Struktur Data STI

Sem. 1 2023/2024

No. Kelompok/Kelas : 05/01

Nama Kelompok : Kelompok 5 K-01

Anggota Kelompok (Nama/NIM) : 1. Bobby Dhammawan TANG/18220037

2. Jihan Aurelia/18222001

3. Benedicta Eryka Santosa/18222031




4. Firsia Athaya Raissa Alifah/18222051



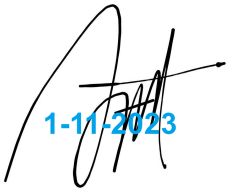
5. Muhammad Faiz A/18222063

Asisten Pembimbing : Aufa Fauqi Ardhiqi


STEI-ITB	IF2111-TB-01-05	Halaman 42 dari 48 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.		





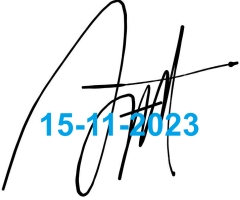
Asistensi I

Tanggal : Rabu, 1 November 2023	Catatan Asistensi: <ul style="list-style-type: none"> • Jangan lupa invite asisten ke Github. • Kalau ada yang bingung bisa cek Spotify dulu sebagai referensi. • Default artinya data statis, kalau data dinamis itu seperti yang digunakan di Load. • Load and Start → command yang <u>utama</u> buat dikerjakan. Kerjakan bareng dengan command Save. Ketiga command ini berurusan dengan file txt. • Song → memanipulasi data yang sudah dibuat untuk ADT Queue dan Stack. • Riwayat lagu pakai Stack. • Song Previous → mengambil lagu dari stack lalu di-play. Perhatikan kalau lagu yg di-play ditaruh di urutan pertama dalam Queue. • Status → lagu di luar playlist ga perlu ditampilkan sebagai status. • Pastikan hanya pakai library yang boleh digunakan. Ga boleh pakai strict.h. • Cara memisahkan nama playlist, penyanyi, dan judul lagu adalah dengan memisahkannya ke dalam list penyanyi, album, dan judul lagu. Pemisahan dilakukan menggunakan for loop yang membaca elemen pertama sebagai jumlah looping, dan elemen selanjutnya ke dalam masing-masing list. • Lagu bukan hanya disimpan dalam bentuk List. Ada di Set juga. Lagu pakai Map, Album pakai Set. • Boleh bikin command penyanyi.add (menerima string) untuk masukan data ke list penyanyi. Command Load adalah yang nentuin kapan penyanyi akan di add. • Command Load lumayan berat jadi boleh dikerjakan bareng. • Jangan lupa aplikasikan invalid command di main.c, bebas caranya gimana. • Apapun yang mau ditanya tolong dikumpulkan ke 1 orang perwakilan (Eryka) dulu baru ditanyain ke kakaknya lewat Line.
Tempat : Zoom Meetings	
Kehadiran Anggota Kelompok: <div style="text-align: center;"> <p>1</p> <p>18220037</p>  <p>2</p> <p>18220001</p>  <p>3</p> <p>18222031</p>  <p>4</p> <p>18222051</p> </div>	

 5 18222063 	
	Tanda Tangan Asisten:  1-11-2023

Asistensi II

Tanggal : Rabu, 15 November 2023	Catatan Asistensi: <ul style="list-style-type: none"> Update progress tubes: ADT secara garis besar sudah jadi, tapi masih bisa ada perubahan. Command-command sedang dikerjakan. Perhatikan program dan laporan, karena deadline sudah dekat. Command seperti start load save termasuk penting. Lagu dengan judul yang sama tapi oleh penyanyi yang berbeda diperbolehkan, karena masih unik (nama penyanyinya beda). Sebenarnya kapan mau nyicil laporan itu dibebaskan ke kalian. Mau nyicil dari sekarang juga oke. Kalau ada command yang gak bisa dikerjain, mending kerjain bareng.
Tempat : Zoom Meetings	
Kehadiran Anggota Kelompok: 1 18220037  2 18220001	

 3 18222031  4 18222051  5 18222063 	<p>Command yang susah bisa dikerjain lebih dari 1 orang.</p> <ul style="list-style-type: none"> • ADT map untuk penyanyi dapat memuat set of integer yang merupakan id dari ADT map untuk album, namun ADT map album harus langsung menyimpan nama lagu. • Kalau ada pertanyaan, langsung chat lewat Line aja.
	<p>Tanda Tangan Asisten:</p>  15-11-2023

9.3 Log Activity Anggota Kelompok

No.	Tanggal	NIM	Nama	Aktivitas
1	30/10/2023	18220037	Bobby D	Membuat repository github
2	30/10/2023	18222031	B. Eryka	Membuat format laporan
3	01/11/2023	18220037	Bobby D	Push set.c, main.c
4	05/11/2023	18222001	Jihan	Push queue.c, queue.h
5	06/11/2023	18222063	Faiz	Push boolean.h, stack.h, stack.c
6	08/11/2023	18222001	Jihan	update gitignore dimarahin faiz
7	08/11/2023	18222063	Faiz	Push start & console, mapPlaylist, fix ADT path, update PLAYLIST ADD
8	13/11/2023	18222063	Faiz	Push PLAYLIST SWAP, REMOVE, DELETE
9	13/11/2023	18222001	Jihan	Push queue procedure
10	14/11/2023	18222051	Firsa	Push ADT mesinkarakter.h, mesinkarakter.c, mesinkata.h, mesinkata.h, drivermesinkata.c
11	14/11/2023	18222063	Faiz	fix path, filenaming, ADT, update PLAYLIST, remove duplicates
12	14/11/2023	18220037	Bobby D	Push map.h, map.c, set.h, set.c, driverSet.c, driverMap.c
13	15/11/2023	18222063	Faiz	update PLAYLIST, add ADT, fix bugs
14	16/11/2023	18222001	Jihan	Push Ismember, queue_delln
15	16/11/2023	18222063	Faiz	Push Lagu, ArraydinWord, ADT Set, driver stack, update PLAYLIST REMOVE & DELETE, ADT Stack, fix ADT

				mesin kata bug
16	16/11/2023	18220037	Bobby D	Push start.h, start.c, parseConfig.h, parseConfig.c, main.c, update map.c
17	18/11/2023	18222063	Faiz	Push ADT lagu, PlaylistID, Status Command, fix Quit & Help
18	18/11/2023	18222051	Firsa	Push help.h, help.c, parsingcommand, invalid_command.h, invalid_command.c
19	18/11/2023	18222001	Jihan	Push queue.h, queue.c
20	19/11/2023	18222001	Jihan	Update queue_Ismember, songPrevious
21	19/11/2023	18222063	Faiz	Update parseConfig, List, Queue, push driver Song & Lagu, List, ListDinamis, ListBerkait
22	20/11/2023	18222051	Firsa	Mengisi laporan bagian ringkasan, spesifikasi, struktur data (ADT), Data Test, Lampiran
23	20/11/2023	18222001	Jihan	Mengisi laporan bagian program utama, spesifikasi, struktur data (ADT), Data Test, Test Script
24	19/11/2023	18222063	Faiz	Mengisi laporan bagian struktur data (ADT), program utama, algoritma-algoritma menarik
25	21/11/2023	18222001	Jihan	update driverqueue.c, queue.h, queue.c, song.c, play playlist, load, format input, fixing
26	21/11/2023	18222063	Faiz	Push play, save, playlist swap command, fix bugs, update

				play.c, isLoad on start.c
27	21/11/2023	18222031	B. Eryka	adjust print format
28	22/11/2023	18222063	Faiz	update save, quit, song previous, push bonus: multi user, fix parseConfig, Playlist swap, delete, remove command
29	22/11/2023	18222051	Firsa	fix status.h, save.h, parseConfig.h, queue.h, user.h
30	23/11/2023	18222063	Faiz	Fix Quit, Load, List Playlist & Riwayat Bug
31	24/11/2023	18222063	Faiz	Fixing format, melengkapi laporan bagian data test, Menambahkan Fitur Bonus: Enhance
32	24/11/2023	18222001	Jihan	Fixing format, melengkapi laporan bagian ADT
33	24/11/2023	18222051	Firsa	mengisi Log Activity, finishing laporan
34	24/11/2023	18222031	B. Eryka	Memberi keterangan tambahan pada laporan, finishing laporan