

LAPORAN TUGAS BESAR

IF2111 Algoritma dan Struktur Data STI

WayangWave


Dipersiapkan oleh:

Kelompok 9

Thalita Zahra Sutejo	18222023
Alfandito Rais Akbar	18222037
Givari Al Fachri	18222045
Dama Dhananjaya Daliman	18222047
Satria Wisnu Wibowo	18222087

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		IF2111-TB-01-09		3
		Revisi	1	25 Oktober 2023

Daftar Isi

1	Ringkasan.....	3
2	Struktur Data (ADT).....	5
	2.1 Struktur Data Lagu.....	5
	2.2 Struktur Data Linked List.....	6
	2.3 Struktur Data List Dinamis.....	6
	2.4 Struktur Data List Statis.....	6
	2.5 Struktur Data Map.....	7
	2.6 Struktur Data Mesin Karakter.....	7
	2.7 Struktur Data Mesin Kata.....	7
	2.8 Struktur Data Queue.....	8
	2.9 Struktur Data Set.....	8
	2.10 Struktur Data Stack.....	9
3	Program Utama.....	9
4	Algoritma-Algoritma Menarik.....	10
	4.1 Enqueue First.....	10
5	Data Test.....	11
	5.1 Data Test START.....	11
	5.2 Data Test LOAD.....	13
	5.3 Data Test LIST DEFAULT.....	14
	5.4 Data Test LIST PLAYLIST.....	14
	5.5 Data Test PLAY SONG.....	14
	5.6 Data Test PLAY PLAYLIST.....	15
	5.7 Data Test QUEUE SONG.....	16
	5.8 Data Test QUEUE PLAYLIST.....	17
	5.9 Data Test QUEUE SWAP.....	18
	5.10 Data Test QUEUE REMOVE.....	20
	5.11 Data Test QUEUE CLEAR.....	21
	5.12 Data Test SONG NEXT.....	21
	5.13 Data Test SONG PREVIOUS.....	22
	5.14 Data Test PLAYLIST CREATE.....	23
	5.15 Data Test PLAYLIST ADD SONG.....	23
	5.16 Data Test PLAYLIST ADD ALBUM.....	24
	5.17 Data Test PLAYLIST SWAP.....	24
	5.18 Data Test PLAYLIST REMOVE.....	25

5.19 Data Test PLAYLIST DELETE.....	25
5.20 Data Test STATUS.....	25
5.21 Data Test SAVE.....	25
5.22 Data Test QUIT.....	27
5.23 Data Test HELP.....	28
6 Test Script.....	29
7 Pembagian Kerja dalam Kelompok.....	33
8 Lampiran.....	34
8.1 Deskripsi Tugas Besar.....	34
1. About the System.....	34
2. Main Menu.....	34
3. Command.....	34
a. START.....	35
b. LOAD <filename>.....	35
c. LIST.....	35
d. PLAY.....	35
e. QUEUE.....	36
f. SONG.....	36
g. PLAYLIST.....	36
h. STATUS.....	37
i. SAVE <filename>.....	37
j. QUIT.....	37
k. HELP.....	38
8.2 Notulen Asistensi.....	38
8.2.1 Notulensi Asistensi 1.....	38
8.2.1 Notulensi Asistensi 2.....	39
8.3 Log Activity Anggota Kelompok.....	41

1 Ringkasan

WayangWave merupakan suatu aplikasi yang mensimulasikan pemutaran musik berbasis *Command-Line Interface* (CLI) yang memungkinkan pengguna dapat menikmati berbagai fitur terkait pemutaran lagu. WayangWave memiliki beberapa fitur utama, yaitu memutar lagu, menampilkan daftar lagu, membuat dan menghapus *playlist*, mengatur urutan dimainkannya lagu, dan menampilkan status dari aplikasi tersebut. Semua perubahan yang telah dilakukan terhadap aplikasi pemutaran musik, seperti penggunaan fitur yang telah disebutkan, akan disimpan pada sebuah *file* dengan *type* txt. Aplikasi ini dirancang menggunakan bahasa pemrograman C dan mengimplementasikan struktur data yang relevan untuk memenuhi kebutuhan layanan musik yang ditawarkan.

Aplikasi WayangWave memiliki alur kerja dan interaksi pengguna. Pertama, **keberjalanan aplikasi** dimulai dari opsi START, LOAD, dan HELP. Dengan memilih START, aplikasi membaca file konfigurasi *default* yang berisi daftar penyanyi, album, dan lagu yang tersedia. Kemudian, LOAD memungkinkan pengguna untuk memuat *save file* sebelumnya dari *folder* tertentu untuk melanjutkan sesi sebelumnya.

Kedua, terkait **navigasi daftar musik**. Pengguna dapat menggunakan *command* LIST untuk melihat daftar penyanyi, album, dan lagu yang ada, baik dalam *default* maupun dalam *playlist*. Pengguna juga dapat memilih penyanyi dan album memungkinkan pengguna untuk melihat lagu-lagu yang tersedia dan memutuskan apakah ingin melihat lebih lanjut atau tidak.

Ketiga, **pemutaran lagu atau pengaksesan *playlist***. Pengguna dapat menggunakan *command* PLAY yang digunakan untuk memainkan lagu atau *playlist*. Pengguna dapat memilih lagu berdasarkan penyanyi, album, dan id lagu atau id *playlist* yang ingin diputar. Pilihan ini memungkinkan pengguna untuk mengelola urutan pemutaran, serta menampilkan lagu yang sedang diputar.

Keempat, **manajemen *queue***. *Command* QUEUE memungkinkan pengguna untuk memanipulasi *queue* lagu, termasuk menambahkan lagu atau *playlist* ke dalam *queue*, menukar posisi lagu dalam *queue*, menghapus lagu dari *queue*, atau membersihkan *queue*.

Kelima, **melakukan navigasi lagu ke dalam *queue***. *Command* SONG memungkinkan pengguna untuk menavigasi lagu dalam *queue* dengan menampilkan lagu selanjutnya atau sebelumnya dalam urutan pemutaran.

Keenam, **terkait manajemen suatu *playlist***. Pengguna dapat akses *command* PLAYLIST yang dapat digunakan untuk membuat *playlist* baru, menambahkan lagu atau album ke dalam *playlist* yang ada, mengelola urutan lagu dalam *playlist*, menghapus lagu dari *playlist*, atau menghapus *playlist*.

Ketujuh, **terkait status pemutaran lagu**. Pengguna dapat mengakses *command* STATUS dan memungkinkan pengguna untuk melihat status pemutaran saat ini, termasuk lagu yang sedang diputar, daftar *queue* lagu, dan informasi mengenai *playlist* mana lagu tersebut sedang diputar.

Kedelapan, **terkait simpan status dan keluar dari aplikasi**. *Command* SAVE digunakan untuk menyimpan status atau *state* aplikasi ke dalam *file save*, memungkinkan pengguna untuk mengaksesnya di sesi selanjutnya. *Command* QUIT memungkinkan pengguna untuk keluar dari sesi aplikasi, dengan atau tanpa menyimpan data sesi terkini, sesuai dengan pilihan pengguna.

Terakhir, terkait *menu bantuan* atau *help*. *Command HELP* memberikan bantuan dengan menampilkan daftar *command* yang dapat digunakan beserta deskripsi fungsinya, memandu pengguna dalam menggunakan aplikasi.

Dengan *beragam command dan fungsi yang disediakan*, WayangWave memberikan pengalaman yang fleksibel dan mudah digunakan bagi pengguna untuk menikmati pemutaran musik sesuai keinginan mereka, melakukan manajemen *playlist*, serta mengatur urutan pemutaran lagu dengan nyaman dan efisien melalui *command-line interface* yang sederhana.

2 Struktur Data (ADT)

Dalam program ini, kami menggunakan beberapa ADT, yaitu ADT Lagu, ADT Linked List, ADT List Dinamis, ADT List Statis, ADT Map, ADT Mesin Karakter, ADT Mesin Kata, ADT Queue, ADT Set, dan ADT Stack.

2.1 Struktur Data Lagu

Pada pengerjaan program WayangWave, kami menggunakan struktur data Lagu (*struct Song*). Struktur data ini merupakan struktur yang merepresentasikan tipe bentukan sebuah Lagu. Struktur data Lagu (*Song*) yang kami gunakan ini menyimpan beberapa data seperti tabel penyimpanan data **Penyanyi**, **Album**, dan **Lagu** dengan tipe data *Word*. Data penyanyi disimpan dalam tipe data *Word* atau kata yang merupakan hasil bentukan dari ADT Mesin Kata. Data penyanyi ini merepresentasikan nama seseorang yang menciptakan/menyanyikan lagu berdasarkan data lagu yang tersimpan. Selanjutnya, data album juga disimpan dalam tipe data *Word* atau kata. Data album ini merepresentasikan nama album tempat dimana data lagu tersimpan. Terakhir, data lagu yang disimpan dengan tipe data *Word* atau kata. Data lagu ini merepresentasikan judul dari lagu yang tersimpan.

Struktur data ***Song*** digunakan untuk menyimpan deskripsi dari sebuah lagu, yang terdiri dari informasi tentang penyanyi, album, dan judul lagu. Struktur data ini yang nantinya akan diputar, ditampilkan, diantrikan, dan dinyatakan status pemutarannya di aplikasi WayangWave.

Kemudian, kami akan menjelaskan terkait alasan pemilihan implementasi. Kami memilih tipe data *Word* karena tipe data ini digunakan untuk menyimpan informasi teks. Hal ini memungkinkan adanya fleksibilitas dalam penggunaan teks yang bisa disimpan dalam struktur data ***Song***. Selanjutnya, terkait implementasi fungsi-fungsi, seperti *CreateSong*, *IsEmptySong*, *PasteSong*, dan *DisplaySong* digunakan untuk memanipulasi dan menampilkan informasi terkait deskripsi lagu, memberikan kemampuan untuk membuat, memeriksa kekosongan, menyalin, dan menampilkan informasi tersebut.

File 'lagu.h' dan 'lagu.c' membentuk sebuah ADT "Deskripsi Lagu", yang berisi struktur data ***Song*** dan fungsi-fungsi yang beroperasi pada struktur data tersebut. Pendekatan ini memungkinkan penggunaan yang jelas dalam manipulasi data ***Song***, menyembunyikan detail implementasi, dan menyediakan fungsionalitas terstruktur untuk mengelola informasi terkait lagu. Ini memberikan modularitas, abstraksi, dan memungkinkan penggunaan ulang fungsi-fungsi tersebut di berbagai bagian program.

2.2 Struktur Data Linked List

Pada pengerjaan program WayangWave, kami menggunakan struktur data Linked List (*struct **LinkedList***) dan struktur data Node (*struct **Node***). Struktur linked list yang kami gunakan terdiri atas *first* yang merujuk ke node pertama linked list dan *last* yang merujuk ke node terakhir linked list dan masing masing bertipe *address*. Address sebagai pointer ke suatu node yang digunakan untuk menyimpan alamat dari suatu node ke dalam linked list. Sedangkan struktur node terdiri atas *info* yang menyimpan nilai dari elemen linked list, dalam hal ini bertipe *song* dan *next* yang menyimpan alamat dari node berikutnya dalam linked list.

Struktur linked list ini merupakan struktur yang merepresentasikan sebuah playlist dimana playlist merupakan suatu kumpulan lagu yang kontigu. Struktur data LinkedList diterapkan pada program main yang kami buat dalam persoalan playlist, karena membantu dalam penambahan dan penghapusan playlist dengan efisien. Ketika playlist baru dibuat atau dihapus, maka dapat dilakukan tanpa perlu memindahkan atau menggeser elemen lainnya.

File 'linkedlist_v2.h' dan 'linkedlist_v2.c' membentuk sebuah ADT Linked List yang berisi struktur data '**LinkedList**' dan struktur data '**Node**'. Di dalam file tersebut, terdapat implementasi fungsi fungsi yang digunakan untuk memanipulasi dan menampilkan informasi terkait playlist.

2.3 Struktur Data List Dinamis

Pada pengerjaan program WayangWave, kami menggunakan struktur data List Dinamis (*struct **ArrayDin***). Struktur list yang kami gunakan menyimpan data yaitu elemen *ElType*, *Capacity*, dan *Neff*. Elemen *ElType* di sini adalah memori tempat penyimpanan elemen pada *array*. Data *Capacity* pada struktur data ini bertipe *integer* dan merupakan sebuah bilangan yang menandakan kapasitas dari ArrayDin yang tersedia. Data *Neff* pada struktur data ini bertipe *integer* dan berperan sebagai penanda jumlah elemen efektif pada *array*. Elemen pada **List** disimpan dalam tipe *Word* atau kata yang merupakan hasil bentukan dari ADT Mesin Kata.

Struktur list ini merupakan struktur yang merepresentasikan kumpulan playlist yang nantinya berada di dalam list **daftarPlaylist**. Struktur data ArrayDin diterapkan pada program main yang kami buat dalam persoalan penyimpanan playlist. Selain itu, penggunaan ArrayDin dapat menjadi tabel penyimpanan yang tepat bagi daftar playlist yang tersedia pada program. Daftar playlist tersebut disimpan untuk batasan jumlah penyimpanan yang cenderung lebih fleksibel karena dapat melakukan alokasi dan dealokasi sehingga lebih efektif.

File 'arraydin.h' dan 'arraydin.c' membentuk sebuah ADT array dinamis yang berisi struktur data '**ArrayDin**'. Di dalam file tersebut, terdapat implementasi fungsi fungsi yang digunakan untuk memanipulasi dan menampilkan informasi terkait daftar playlist.

2.4 Struktur Data List Statis

Pada pengerjaan program WayangWave, kami menggunakan struktur data List Statis (*struct **List***). Struktur list yang kami gunakan menyimpan data yaitu elemen *ElType*. Elemen *ElType* di sini adalah memori tempat penyimpanan elemen. Elemen pada **List** disimpan dalam tipe *Word* atau kata yang merupakan hasil bentukan dari ADT Mesin Kata. Struktur list ini merupakan struktur yang merepresentasikan kumpulan nama penyanyi yang nantinya berada di dalam list **daftarPenyanyi**.

List sudah mulai berperan sejak awal program **main** dijalankan dengan membuat suatu

list yaitu list *daftarPenyanyi*. Pada saat *command* STARTWAYANGWAVE, program akan membaca daftar penyanyi yang terdapat pada file, kemudian akan memasukkannya ke dalam list *daftarPenyanyi* yang berstruktur *List*. Setelah itu, list *daftarPenyanyi* juga akan digunakan dalam *command* LISTDEFAULT dan PLAYSONG untuk menampilkan daftar penyanyi yang berada dalam list tersebut.

File 'list.h' dan 'list.c' membentuk sebuah ADT List Statis yang berisi struktur data *List*. Di dalam file tersebut, terdapat implementasi fungsi fungsi yang digunakan untuk memanipulasi dan menampilkan informasi terkait daftar penyanyi.

2.5 Struktur Data Map

Pada pengerjaan program WayangWave, kami menggunakan struktur data Map(*struct Map*). Struktur data ini merupakan struktur yang merepresentasikan album-album yang dimiliki penyanyi dan lagu-lagu suatu album.

ADT Map digunakan untuk membuat dua buah map yaitu *penyanyiAlbum* yang menjadikan penyanyi sebagai *Key* dan album-album sebagai set of list sebagai *Value* dalam map dan *albumLagu* yang menjadikan nama album sebagai *Key* dan lagu-lagu sebagai set of list sebagai *Value* dalam map.

File 'map.h' dan 'map.c' membentuk sebuah ADT Map yang berisi struktur data *Map*. Di dalam file tersebut, terdapat implementasi fungsi fungsi yang digunakan untuk memanipulasi dan menampilkan informasi terkait daftar album yang dimiliki suatu penyanyi dan daftar lagu yang dimiliki suatu album.

2.6 Struktur Data Mesin Karakter

Pada pengerjaan program WayangWave, kami menggunakan struktur data ADT Mesin Karakter yang digunakan untuk menjalankan dan membaca *Command Line Input* dari user dengan menggunakan ADT Mesin Kata.

ADT Mesin Karakter digunakan untuk membaca input per karakter yang nantinya akan disimpan dalam struktur data *Word* (Word.Tabword) untuk diolah lebih lanjut pada ADT Mesin Kata. ADT ini juga akan menentukan apakah EOP (End of Pita) sudah menyala atau tidak, yang mana apabila menyala menandakan akhir dari input user dan sebaliknya.

Implementasi fungsi pada ADT Mesin Karakter yaitu START, STARTFILE, ADV, dan isEOP. Fungsi START dan STARTFILE digunakan untuk mengambil input dari user dengan menggunakan *stdin* atau dari file config default atau file save user. ADV digunakan untuk memajukan pita sebanyak satu karakter sampai pada akhirnya menyentuh MARK yaitu ' ; '. Terakhir, isEOP akan mengembalikan boolean yang menandakan apakah pita sudah selesai atau belum.

File 'mesinkarakter.h' dan 'mesinkarakter.c' membentuk sebuah ADT Mesin Karakter yang didalamnya berisi implementasi fungsi-fungsi yang digunakan untuk menerima dan mengolah data yang didapatkan.

2.7 Struktur Data Mesin Kata

Pada pengerjaan program WayangWave, kami menggunakan struktur data ADT Mesin Kata yang digunakan untuk menjalankan *Command Line Input* untuk mengoperasikan program

WayangWave. Selain itu, ADT ini juga digunakan untuk membaca file config default atau save user sebagai syarat program WayangWave ini berjalan.

Dalam struktur ADT Mesin Kata, terdapat struktur data **Word** yang berisi TabWord berupa array sebanyak *NMax* (50) dan length yang merupakan panjang dari array TabWord yang terisi. ADT ini menggunakan ADT Mesin Karakter untuk menjalankan dan menyimpan karakter pada array Tabword sehingga terbentuk kata atau kalimat.

Dalam ADT ini, terdapat fungsi-fungsi yang diimplementasi dengan tujuan untuk mempermudah pengolahan dari input user ataupun file config. Diantara fungsi-fungsi tersebut terdapat *IgnoreBlank*, *STARTWORD*, *STARTSENTENCE*, *STARTWORDFILE*, *ADVWORD*, *ADVSENTENCE*, *CopyWord*, dan *CopySentence* yang digunakan untuk membaca file config dan menerima dan mengolah input dari user. Selain fungsi-fungsi tersebut terdapat fungsi seperti *isWordEq*, *ToKata*, *WordToInt*, *SalinKata*, *DisplayKata*, dan lain-lain, yang digunakan untuk memudahkan dalam pembuatan fungsi-fungsi lain pada program WayangWave ini.

File 'mesinkata.h' dan 'mesinkata.c' membentuk sebuah ADT Mesin Kata yang didalamnya berupa implementasi fungsi-fungsi yang digunakan. ADT Mesin Kata memerlukan bantuan file 'mesinkarakter.h' dan 'mesinkarakter.c' yang merupakan ADT Mesin Karakter.

2.8 Struktur Data Queue

Pada pengerjaan program WayangWave, kami mengimplementasikan struktur data ADT *Queue* yang digunakan untuk menyimpan elemen-elemen bertipe Song dalam suatu antrian dengan aturan FIFO (*First In First Out*). Struktur data Queue ini memiliki beberapa elemen, yaitu '**buffer**' sebagai array yang menyimpan elemen-elemen '*Song*' dalam antrian, '**idxHead**' sebagai indeks dari elemen pertama dalam antrian, dan terakhir '**idxTail**' sebagai indeks dari elemen terakhir dalam antrian.

ADT *Queue* digunakan untuk menyimpan dan mengelola kumpulan lagu ('*Song*') dalam urutan masuk dan keluar yang berurutan. Implementasi ADT ini memungkinkan penambahan lagu ke antrian ('*enqueue*') sesuai aturan FIFO serta penghapusan lagu dari antrian ('*dequeue*') dengan mempertahankan urutan yang benar.

Implementasi ini digunakan sebagai ADT Queue dengan representasi array secara eksplisit dan alokasi statik. Keputusan ini diambil karena menggunakan array dapat secara efisien mengatur urutan masuk dan keluar elemen-elemen '*Song*' dalam antrian. Alokasi statik dengan ukuran maksimum *CAPACITY* (100) digunakan untuk kesederhanaan dan karena jumlah maksimum elemen yang dapat ditampung sudah ditentukan. File-file '*queue.h*' dan '*queue.c*' diimplementasikan sebagai ADT Queue untuk menyimpan, menambah, dan menghapus lagu dalam struktur data antrian secara terdefinisi dengan jelas. Penggunaan ADT ini memisahkan logika antrian dari bagian program lain yang membutuhkan struktur data antrian, menyediakan cara yang terstruktur untuk mengelola lagu-lagu dalam antrian, serta memfasilitasi penggunaan kembali fungsi-fungsi tersebut di bagian program yang berbeda tanpa harus memperhatikan detail implementasi internalnya.

2.9 Struktur Data Set

Pada pengerjaan program WayangWave, kami menggunakan struktur data Set (*struct Set*). Struktur data ini merupakan struktur yang merepresentasikan kumpulan album-album dan lagu-lagu. Struktur set menyimpan data yaitu set infotype Elements[SetMaxEl].

ADT Set digunakan untuk menyimpan kumpulan album dan lagu. Implementasi dengan ADT Set karena tidak memungkinkannya ada beberapa album atau lagu dengan nama yang sama. Selanjutnya, set album dan set lagu akan digunakan dalam ADT Map. Set album akan menjadi value sebuah key penyanyi dalam suatu map dan set lagu akan menjadi value sebuah key album dalam suatu map.

File ‘set.h’ dan ‘set.c’ membentuk sebuah ADT Set yang didalamnya berupa implementasi fungsi-fungsi yang digunakan. Di dalam file tersebut, terdapat implementasi fungsi fungsi yang digunakan untuk memanipulasi dan menampilkan informasi terkait set of album dan set of songs.

2.10 Struktur Data Stack

Pada pengerjaan program WayangWave, kami menggunakan struktur data Stack (*struct Stack*). Struktur stack yang kami gunakan menyimpan data yaitu address *TOP* dan tabel infotypeS *T[MaxEl]*. Elemen *InfotypeS* di sini adalah memori tempat penyimpanan elemen. Elemen pada *List* disimpan dalam tipe *Song* atau lagu yang merupakan hasil bentukan dari ADT Lagu.

ADT Stack digunakan untuk menyimpan dan mengelola kumpulan data (*infotypeS*) secara urut *Last In First Out* (LIFO). Implementasi ini memungkinkan penambahan elemen ke stack (*PushStack*) dan penghapusan elemen dari stack (*PopStack*) dengan mengikuti aturan LIFO. Selain itu, juga dapat dilakukan pengecekan apakah stack kosong atau penuh.

Implementasi *stack* diambil dengan representasi tabel kontigu dan alokasi statik karena memungkinkan manipulasi data yang efisien, terutama untuk kasus-kasus yang membutuhkan penyimpanan sementara dengan aturan LIFO. Pemilihan alokasi statik dengan ukuran maksimum *MaxEl* (100) dilakukan untuk kesederhanaan dan kejelasan dalam implementasi. File-file *stack.h* dan *stack.c* diimplementasikan sebagai ADT *Stack* untuk menyimpan, menambah, dan menghapus data dalam struktur data *stack* dengan cara yang terdefinisi dengan jelas.

Penggunaan ADT ini memungkinkan penyimpanan data dalam urutan Last-In-First-Out (LIFO), memisahkan logika stack dari bagian program lain yang membutuhkan struktur data tersebut, serta memungkinkan penggunaan kembali fungsi-fungsi tersebut di berbagai bagian program tanpa perlu memperhatikan detail-detail dari implementasinya yang tersimpan di dalamnya.

3 Program Utama

Pada program aplikasi WayangWave, alur *program* dijalankan dan dimulai dari ‘main.c’. Program ini kemudian akan memerintahkan untuk input beberapa *COMMAND*, yaitu *START*, *LOAD*, dan *HELP* sebagai masukan pertama dari aplikasi WayangWave. *COMMAND START* berfungsi untuk membuka aplikasi dengan *file* konfigurasi default yang sudah berisi daftar penyanyi dan album, *COMMAND LOAD <filename>* berfungsi untuk memuat *save file* yang berisi daftar penyanyi, album, dan lagu yang diputar, sedangkan *COMMAND HELP* berfungsi untuk menampilkan daftar *command* yang dapat dieksekusi beserta deskripsinya dengan dua kondisi, apabila *file* konfigurasi telah berhasil dibaca, akan menampilkan **command utama** aplikasi, sedangkan kondisi lainnya, apabila *file* konfigurasi tidak berhasil dibaca, akan menampilkan *command START, LOAD, dan HELP* itu sendiri.

Program ini memiliki beberapa **command** utama setelah **file konfigurasi telah berhasil dibaca**:

1. **LIST DEFAULT**: Menampilkan daftar penyanyi, album dari penyanyi yang dipilih, serta daftar lagu dari album yang dipilih
2. **LIST PLAYLIST**: Menampilkan *playlist* yang ada pada pengguna.
3. **PLAY (SONG, PLAYLIST)**: ₁ Memutar **lagu** berdasarkan nama penyanyi, album, dan ID lagu, atau ₂ memutar *playlist* berdasarkan ID *playlist*.
4. **QUEUE (SONG, PLAYLIST, SWAP <x><y>, REMOVE <id>, CLEAR)** : Memanipulasi *queue* lagu dengan *command* tambah (₁ *song*, ₂ *playlist*), ₃ tukar, ₄ hapus, atau ₅ mengosongkan *queue*.
5. **SONG (NEXT, PREVIOUS)**: Mengontrol navigasi lagu dalam *queue* (putar ₁ lagu selanjutnya, ₂ lagu sebelumnya).
6. **PLAYLIST(CREATE, ADD, SWAP <id> <x><y>, REMOVE <id>, DELETE)**: Melakukan operasi pada *playlist* seperti ₁ membuat, ₂ menambahkan, ₃ menukar, ₄ menghapus lagu, atau ₅ menghapus *playlist*.
7. **STATUS**: Menampilkan informasi lagu yang sedang diputar, *queue* lagu, dan *playlist* yang sedang diputar.
8. **SAVE <filename>**: Menyimpan status aplikasi saat ini ke dalam file.
9. **QUIT**: Keluar dari aplikasi **dengan opsi untuk menyimpan data sesi saat keluar**.

Aplikasi ini memungkinkan pengguna untuk mengelola musik, membuat *playlist*, menavigasi lagu, serta menyimpan dan memuat status aplikasi. Fitur-fitur ini memberikan pengalaman pengguna yang lebih interaktif dan terorganisir saat mendengarkan musik dalam aplikasi WayangWave.

4 Algoritma-Algoritma Menarik

4.1 Enqueue First

Algoritma *enqueue first* digunakan pada *command* SONG PREVIOUS. Ketika *command* itu dijalankan, program akan mengambil lagu yang sebelumnya dijalankan dari *history* untuk dimasukkan pada *queue* pada urutan pertama. Ini berarti, kita perlu membuat fungsi *enqueue* untuk digunakan dalam memasukkan lagu ke urutan pertama *queue*. Algoritma ini unik dikarenakan pada dasarnya *enqueue* merupakan fungsi yang digunakan untuk memasukkan elemen ke dalam *queue* paling belakang dengan prinsip FIFO (*First in First Out*).

Gambar 4.1.1 Algoritma *enqueue first* sebagai algoritma menarik

```
/* *** Tambahan Procedure *** */
/* Proses: Menambahkan lagu pada queue dengan urutan pertama*/
/*I.S.: Queue kosong atau berisi lagu yang telah di-queue sebelumnya
/*F.S.: Queue telah ditambahkan dengan lagu yang ingin dimasukkan pada posisi HEAD */
void enqueueFirst(Queue *q, Song val){
    int i;
    IDX_TAIL(*q)++;
    for (i = IDX_TAIL(*q); i > IDX_HEAD(*q); i--) {
        (*q).buffer[i] = (*q).buffer[i-1];
    }
    HEAD(*q) = val;
}
```

4.2 *SenToWord*

Algoritma *SenToWord* merupakan bagian dari fungsi ADT Mesin Kata. Fungsi ini digunakan untuk mengambil satu kata sesuai dengan indeks kata yang diinginkan dan kemudian mengembalikannya untuk dapat diolah lebih lanjut. Fungsi ini berguna untuk mengolah inputan dari pengguna ataupun file *config* yang memiliki lebih dari satu kata. Algoritma ini menarik dikarenakan fungsi ini merupakan fungsi yang tidak biasa dan jarang digunakan. Biasanya, ADT Mesin Kata dibuat untuk menyimpan satu kata saja ke dalam struktur data *Word*. Dikarenakan program kita memiliki dua jenis inputan yaitu fungsi *STARTWORD* dan *STARTSENTENCE*, yang mana *STARTWORD* digunakan untuk mengambil satu per satu kata saja, sedangkan *STARTSENTENCE* digunakan untuk mengambil per satu kalimat. Fungsi *SenToWord* bertugas untuk memecah kalimat dari *STARTSENTENCE* untuk diolah oleh program.

Gambar 4.2.1 Algoritma *SenToWord* sebagai algoritma menarik

```
193
194 Word SenToWord(Word sentence, int idxKata) {
195     int count = 0;
196     Word Kata; Kata.Length = 0;
197
198     int i = 0;
199     while (i < sentence.Length && count <= idxKata) {
200         Kata.TabWord[Kata.Length] = sentence.TabWord[i];
201         if (sentence.TabWord[i] == ' ') {
202             if (count < idxKata) {
203                 Kata.Length = 0;
204             } count++;
205         }
206         if (sentence.TabWord[i] != ' ') {
207             Kata.Length++;
208         }
209         i++;
210     }
211     Kata.TabWord[Kata.Length] = '\\0';
212     return Kata;
213 }
214
```

5 Data Test

5.1 *Data Test START*

Pada saat program pertama WayangWave pertama kali dimulai, pengguna dapat menggunakan *command* *START* untuk mengkonfigurasi program WayangWave dengan file konfigurasi default. Ketika *command* dijalankan, program akan membaca file *config.txt* yang berisi konfigurasi default.

Pada program, terdapat state yang menandakan sesi dari program WayangWave. Apabila ingin memasuki sesi, pengguna perlu melakukan konfigurasi dengan *START* atau *LOAD*. Setelah *command* berhasil dijalankan state sesi akan berubah sehingga program masuk ke sesi

yang dapat memasukkan *command-command* baru untuk memulai lagu atau membuat playlist. *Command* START hanya dapat diakses ketika belum memasuki sesi. Apabila sudah memasuki sesi, maka pengguna harus memberikan *command* lain dan program akan mengeluarkan pemberitahuan bahwa *command* tidak bisa dieksekusi.

Gambar 5.1.1 Pemanggilan START

```
===== Selamat Datang di =====  
  
          W a y a n g   W a v e  
  
===== ♪ Enjoy The Music ♪ =====  
  
>> START;  
  
File konfigurasi aplikasi berhasil dibaca. WayangWave berhasil dijalankan.  
  
>> █
```

5.2 Data Test LOAD

Pengguna program dapat memanggil command LOAD <filename> untuk merepresentasikan suatu *save file* yang ingin dibuka. Selain menggunakan *command* START, pengguna program juga bisa memulai program dengan *command* LOAD. Ketika pengguna memilih LOAD, program akan mengambil *state* yang sudah pernah disimpan oleh pengguna. Jika pengguna program memberikan suatu nama file yang tidak valid, maka program gagal dijalankan.

Gambar 5.2.1 Pemanggilan LOAD untuk file yang invalid

```
===== Selamat Datang di =====  
  
wayang wave  
  
===== Enjoy The Music =====  
  
>> LOAD INVALIDFILE.TXT;  
  
Save file tidak ditemukan. WayangWave gagal dijalankan.
```

Gambar 5.2.2 Pemanggilan LOAD untuk file yang valid

```
===== Selamat Datang di =====  
  
wayang wave  
  
===== Enjoy The Music =====  
  
>> LOAD EXAMPLE.TXT;  
  
Save file berhasil dibaca. WayangWave berhasil dijalankan.
```

5.3 *Data Test LIST DEFAULT*

Pengguna program dapat memanggil command LIST DEFAULT untuk melihat list penyanyi yang ada. Setelah list ditampilkan, pengguna dapat memilih untuk melihat album dari suatu penyanyi dan setelahnya pengguna dapat memilih untuk melihat lagu dalam suatu album.

Gambar 5.3.1 Pemanggilan LIST DEFAULT

```
>> 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
```

5.4 *Data Test LIST PLAYLIST*

Pengguna program dapat memanggil command LIST DEFAULT untuk melihat list playlist yang dimiliki pengguna. Jika pengguna tidak mempunyai playlist, maka akan menampilkan “Kamu tidak memiliki playlist.” pada layar.

Gambar 5.4.1 Tampilan ketika playlist tidak ada

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

Gambar 5.4.2 Tampilan ketika playlist ada

```
>> LIST PLAYLIST;
Daftar playlist yang kamu miliki:
1. HAHA
2. KAKA
```

5.5 Data Test *PLAY SONG*

Pengguna program dapat memanggil command *PLAY SONG* untuk memainkan lagu berdasarkan masukan nama penyanyi, nama album, dan id lagu.

Gambar 5.5.1 Tampilan pemilihan lagu

```
>> 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 Nama 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;

Queue berhasil dikosongkan.
Memutar lagu "Pink Venom" oleh "BLACKPINK".
```

5.6 Data Test *PLAY PLAYLIST*

Pengguna program dapat memanggil command *PLAY PLAYLIST* untuk memainkan lagu berdasarkan masukan ID playlist, jika playlist tidak memiliki lagu maka akan menampilkan “Playlist kosong. Tidak ada lagu yang dapat diputar.”

Gambar 5.6.1 Tampilan pemilihan playlist

```
>> PLAY PLAYLIST;
Daftar Playlist Pengguna :
1. HJHJ
2. JKDSJS
3. JSOS

Masukkan ID Playlist yang dipilih : 1;

Memutar playlist "HJHJ".

>> STATUS;

Now Playing :
BLACKPINK - Pink Venom - BORN PINK

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

Gambar 5.6.2 Tampilan ketika playlist kosong

```
>> PLAY PLAYLIST;
Daftar Playlist Pengguna :
1. HDHD
2. JKDSJS
3. JSDS

Masukkan ID Playlist yang dipilih : 1;

Playlist kosong. Tidak ada lagu yang dapat diputar.
```

5.7 Data Test QUEUE SONG

Pengguna program ini dapat memanggil *command* QUEUE SONG untuk menambahkan lagu ke dalam *queue*. *Command* ini menampilkan daftar penyanyi, daftar album, daftar lagu dalam album. *Command* ini juga menerima tiga input, yaitu nama penyanyi, nama album, dan id dari lagu yang ingin dimasukkan ke dalam daftar *queue*. Apabila masukan penyanyi, album, dan id lagu tidak sesuai dengan daftar yang ada, akan mengirimkan pesan kesalahan dan keluar dari *command* sehingga pengguna perlu memasukkan kembali *command* QUEUE SONG.

Gambar 5.7.1 Tampilan ketika masukan penyanyi salah

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

Masukkan Nama Penyanyi yang dipilih : BLACKPONK;

Penyanyi BLACKPONK tidak ada dalam daftar. Silakan coba lagi.
```

Gambar 5.7.2 Tampilan ketika masukan nama album salah

```
>> 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 Nama Album yang dipilih : BOORN PINK;

Album BOORN PINK tidak ada dalam daftar. Silakan coba lagi.
```

Gambar 5.7.3 Tampilan ketika masukan id lagu salah

```
>> 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 Nama Album yang dipilih : THE ALBUM;

Daftar Lagu Album THE ALBUM 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 : 32;

ID Lagu 32 tidak ada dalam daftar. Silakan coba lagi.
```


Gambar 5.7.4 Tampilan ketika berhasil menambahkan lagu ke dalam daftar queue

```
>> 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 Nama 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;

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

>> STATUS;

Now Playing :
No songs have been played yet. Please search for a song to begin playback.
Queue :
1. BLACKPINK - Pink Venom - BORN PINK
```

5.8 Data Test QUEUE PLAYLIST

Pengguna program ini dapat memanggil *command* QUEUE PLAYLIST untuk menambahkan lagu yang ada di dalam *playlist* ke dalam *queue*. *Command* ini menampilkan daftar playlist yang tersedia dan menerima satu *input*, yaitu id *playlist*. Apabila masukan id tidak sesuai dengan daftar yang ada atau daftar kosong, akan dikirimkan pesan kesalahan dan keluar dari *command* sehingga pengguna perlu memasukkan kembali *command* QUEUE PLAYLIST.

Gambar 5.8.1 Tampilan ketika daftar playlist kosong atau tidak sesuai dengan daftar playlist

```
===== ♪ Enjoy The Music ♪ =====

>> START;

File konfigurasi aplikasi berhasil dibaca. WayangWave berhasil dijalankan.

>> QUEUE PLAYLIST;
Daftar Playlist Pengguna :

Masukkan ID Playlist: 1;
Tidak ada playlist dengan ID tersebut
```

Gambar 5.8.2 Tampilan ketika daftar playlist sesuai dengan masukan dan berhasil di-queue

```

>> STATUS;

Now Playing :
BLACKPINK - Shut Down - BORN PINK

Queue :
1. BLACKPINK - Typa Girl - BORN PINK
2. BLACKPINK - Ready For Love - BORN PINK

>> QUEUE PLAYLIST;
Daftar Playlist Pengguna :
1. BP TOP3
2. PLAYLIST CREATE MIXED

Masukkan ID Playlist: 1;
Berhasil menambahkan playlist "BP TOP3" ke queue.

>> STATUS;

Now Playing :
BLACKPINK - Shut Down - BORN PINK

Queue :
1. BLACKPINK - Typa Girl - BORN PINK
2. BLACKPINK - Ready For Love - BORN PINK
3. BLACKPINK - Pink Venom - BORN PINK
4. BLACKPINK - Shut Down - BORN PINK
5. BLACKPINK - Typa Girl - BORN PINK
6. BLACKPINK - Ready For Love - BORN PINK

```

5.9 Data Test *QUEUE SWAP*

Pengguna program ini dapat memanggil *command* *QUEUE SWAP* untuk menukar urutan lagu yang berada dalam *queue*. Terdapat tiga kondisi, apabila pengguna tidak memasukkan parameter, memasukkan parameter dengan id yang sama, atau *queue* kosong maka akan dicetak pesan kesalahan. Parameter yang dimasukkan adalah id yang mau ditukar urutannya.

Gambar 5.9.1 Pemanggilan *QUEUE SWAP* benar dan berhasil

```

>> STATUS;

Now Playing :
BLACKPINK - Shut Down - BORN PINK

Queue :
1. BLACKPINK - Typa Girl - BORN PINK
2. BLACKPINK - Ready For Love - BORN PINK

>> QUEUE SWAP 1 2;

Lagu "Typa Girl" berhasil ditukar dengan "Ready For Love"

>> STATUS;

Now Playing :
BLACKPINK - Shut Down - BORN PINK

Queue :
1. BLACKPINK - Ready For Love - BORN PINK
2. BLACKPINK - Typa Girl - BORN PINK

>> █

```

Gambar 5.9.2 Pemanggilan QUEUE SWAP dengan tidak memasukkan parameter

```

===== Selamat Datang di =====

wayang wave

===== ♪ Enjoy The Music ♪ =====

>> START;

File konfigurasi aplikasi berhasil dibaca. WayangWave berhasil dijalankan.

>> QUEUE SWAP;

Argumen input kurang, tidak melakukan swap.

>> █

```

Gambar 5.9.2 Pemanggilan QUEUE SWAP dengan parameter id yang sama

```

===== Selamat Datang di =====

wayang wave

===== ♪ Enjoy The Music ♪ =====

>> START;

File konfigurasi aplikasi berhasil dibaca. WayangWave berhasil dijalankan.

>> QUEUE SWAP 1 1;

Argumen x dan y sama... Tidak melakukan perubahan pada queue.

>> █

```

Gambar 5.0.2 Pemanggilan QUEUE SWAP dengan queue kosong

```

wayang wave

===== ♪ Enjoy The Music ♪ =====

>> START;

File konfigurasi aplikasi berhasil dibaca. WayangWave berhasil dijalankan.

>> STATUS;

Now Playing :
No songs have been played yet. Please search for a song to begin playback.
Queue :
Your queue is empty.

>> QUEUE SWAP 1 2;

Queue kosong. Tidak bisa melakukan swap.

>> █

```

5.10 Data Test *QUEUE REMOVE*

Pengguna program ini dapat memanggil *command* *QUEUE REMOVE* untuk menghapus lagu yang ada pada urutan atau id yang dimasukkan. Apabila *queue* kosong dan parameter kurang maka akan dicetak pesan kesalahan.

*Gambar 5.10.1 Pemanggilan *QUEUE REMOVE**

```
>> LOAD EXAMPLE;

Save file berhasil dibaca. WayangWave berhasil dijalankan.

>> STATUS;

Now Playing :
BLACKPINK - Shut Down - BORN PINK

Queue :
1. BLACKPINK - Typa Girl - BORN PINK
2. BLACKPINK - Ready For Love - BORN PINK

>> QUEUE REMOVE 1;

Lagu "Typa Girl" oleh "BLACKPINK" telah dihapus dari queue!

>> STATUS;

Now Playing :
BLACKPINK - Shut Down - BORN PINK

Queue :
1. BLACKPINK - Ready For Love - BORN PINK

>> █
```

*Gambar 5.10.2 Pemanggilan *QUEUE REMOVE* dengan parameter kurang*

```
===== Selamat Datang di =====

wayang wave

===== ♪ Enjoy The Music ♪ =====

>> START;

File konfigurasi aplikasi berhasil dibaca. WayangWave berhasil dijalankan.

>> QUEUE REMOVE;

Argumen 'id' belum diberikan!

>> █
```

*Gambar 5.10.3 Pemanggilan *QUEUE REMOVE* dengan queue kosong*

```
===== Selamat Datang di =====

wayang wave

===== ♪ Enjoy The Music ♪ =====

>> START;

File konfigurasi aplikasi berhasil dibaca. WayangWave berhasil dijalankan.

>> QUEUE REMOVE 1;

Queue kosong. Tidak bisa melakukan remove

>> █
```

5.11 Data Test *QUEUE CLEAR*

Pengguna program ini dapat memanggil *command* *QUEUE CLEAR* untuk mengosongkan daftar *queue* lagu.

Gambar 5.11.1 Pemanggilan *QUEUE CLEAR*

```
>> LOAD EXAMPLE;

Save file berhasil dibaca. WayangWave berhasil dijalankan.

>> STATUS;

Now Playing :
BLACKPINK - Shut Down - BORN PINK

Queue :
1. BLACKPINK - Typa Girl - BORN PINK
2. BLACKPINK - Ready For Love - BORN PINK

>> QUEUE CLEAR;

Queue berhasil dikosongkan.

>> STATUS;

Now Playing :
BLACKPINK - Shut Down - BORN PINK

Queue :
Your queue is empty.

>> |
```

5.12 Data Test *SONG NEXT*

Pengguna program dapat memanggil *command* *SONG NEXT* untuk melanjutkan pemutaran lagu yang ada pada *queue* dengan urutan pertama. Apabila *queue* kosong dan tidak ada lagu yang dimainkan, pesan kesalahan akan ditampilkan. Apabila terdapat lagu yang dimainkan dan *queue* kosong, lagu yang sedang dimainkan akan dimainkan kembali. Apabila terdapat lagu dalam *queue*, baik ada maupun tidak ada lagu yang diputar, lagu yang berada di urutan pertama pada daftar *queue* akan dimainkan.

Gambar 5.12.1 Tampilan ketika *queue* kosong dan tidak ada lagu yang dimainkan

```
===== ♪ Enjoy The Music ♪ =====

>> START;

File konfigurasi aplikasi berhasil dibaca. WayangWave berhasil dijalankan.

>> SONG NEXT;

Queue kosong. Tidak ada lagu yang dimainkan.
```

Gambar 5.12.2 Tampilan ketika berhasil menjalankan lagu dari daftar queue dengan urutan pertama

```
>> LOAD EXAMPLE;

Save file berhasil dibaca. WayangWave berhasil dijalankan.

>> STATUS;

Now Playing :
BLACKPINK - Shut Down - BORN PINK

Queue :
1. BLACKPINK - Typa Girl - BORN PINK
2. BLACKPINK - Ready For Love - BORN PINK

>> SONG NEXT;

Memutar lagu selanjutnya
"Typa Girl" oleh "BLACKPINK"

>> STATUS;

Now Playing :
BLACKPINK - Typa Girl - BORN PINK

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

Gambar 5.12.3 Tampilan ketika terdapat lagu yang dimainkan namun queue kosong

```
>> STATUS;

Now Playing :
BLACKPINK - Ready For Love - BORN PINK

Queue :
Your queue is empty.

>> SONG NEXT;

Queue kosong, memutar kembali lagu
"Ready For Love" oleh "BLACKPINK"
```

5.13 Data Test SONG PREVIOUS

Pengguna program dapat memanggil *command* SONG PREVIOUS untuk melanjutkan pemutaran lagu yang ada di riwayat dengan urutan yang paling terakhir atau lagu yang sebelumnya dimainkan. Apabila riwayat kosong dan tidak ada lagu yang dimainkan, pesan kesalahan akan ditampilkan. Apabila terdapat lagu yang dimainkan dan riwayat kosong, lagu yang sedang dimainkan akan dimainkan kembali. Apabila terdapat lagu dalam riwayat, baik ada maupun tidak ada lagu yang diputar, lagu yang berada di urutan terakhir pada daftar riwayat akan dimainkan atau memutar lagu sebelumnya.

Gambar 5.13.1 Tampilan ketika riwayat kosong dan tidak ada lagu yang dimainkan

```
===== ♪ Enjoy The Music ♪ =====

>> START;

File konfigurasi aplikasi berhasil dibaca. WayangWave berhasil dijalankan.

>> SONG PREVIOUS;

Riwayat lagu kosong. Tidak ada lagu yang dimainkan

>> █
```

Gambar 5.13.2 Tampilan ketika berhasil menjalankan lagu dari daftar riwayat dengan urutan terakhir atau lagu yang diputar sebelumnya

```
>> LOAD EXAMPLE;

Save file berhasil dibaca. WayangWave berhasil dijalankan.

>> STATUS;

Now Playing :
BLACKPINK - Shut Down - BORN PINK

Queue :
1. BLACKPINK - Typa Girl - BORN PINK
2. BLACKPINK - Ready For Love - BORN PINK

>> SONG NEXT;

Memutar lagu selanjutnya
"Typa Girl" oleh "BLACKPINK"

>> SONG PREVIOUS;

Memutar lagu sebelumnya
"Shut Down" oleh "BLACKPINK"
```

Gambar 5.13.3 Tampilan ketika terdapat lagu yang dimainkan namun riwayat kosong

```
>> SONG PREVIOUS;
Memutar lagu sebelumnya
"Shut Down" oleh "BLACKPINK"

>> SONG PREVIOUS;
Memutar lagu sebelumnya
"Pink Venom" oleh "BLACKPINK"

>> SONG PREVIOUS;

Riwayat lagu kosong. Memutar kembali lagu
"Pink Venom" oleh "BLACKPINK"

>> █
```

5.14 Data Test **PLAYLIST CREATE**

Pengguna program dapat memanggil *command* PLAYLIST CREATE untuk membuat playlist yang baru. Playlist berhasil dibuat jika pengguna menginput nama playlist lebih dari 3 karakter, selanjutnya playlist tersebut akan terdaftar di daftar playlist.

Gambar 5.14.1 Tampilan ketika berhasil membuat playlist

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

Gambar 5.14.2 Tampilan ketika gagal membuat playlist

```
>> PLAYLIST CREATE;
Masukkan nama playlist yang ingin dibuat : G;
Minimal terdapat 3 karakter selain whitespace dalam nama playlist. Silakan coba lagi.
```

5.15 Data Test **PLAYLIST ADD SONG**

Pengguna program dapat memanggil *command* PLAYLIST ADD SONG untuk menambahkan lagu pada suatu playlist yang sebelumnya sudah terdaftar pada daftar playlist. Pesan *error* akan ditampilkan jika inputan pengguna tidak *valid* pada setiap permintaan inputan.

Gambar 5.15.1 Tampilan ketika menambahkan lagu ke playlist

```
>> 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 : 2;

Daftar Playlist Pengguna :
1. GIVARI

Masukkan ID Playlist yang dipilih : 1;

Lagu dengan judul "Shut Down" pada album BORN PINK oleh penyanyi
BLACKPINK berhasil ditambahkan ke dalam playlist GIVARI.
```

5.16 Data Test **PLAYLIST ADD ALBUM**

Pengguna program dapat memanggil *command* **PLAYLIST ADD ALBUM** untuk menambahkan album pada suatu playlist yang sebelumnya sudah terdaftar pada daftar playlist. Pesan *error* akan ditampilkan jika inputan pengguna tidak *valid* pada setiap permintaan inputan.

Gambar 5.16.1 Tampilan ketika menambahkan album ke playlist

```
>> 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. GIVARI

Masukkan ID Playlist yang dipilih : 1;

Album dengan judul "Humbug" berhasil ditambahkan ke dalam playlist pengguna "GIVARI".
```

5.17 Data Test **PLAYLIST SWAP**

Pengguna program dapat memanggil *command* **PLAYLIST SWAP** untuk menukar urutan lagu pada playlist yang terdaftar.

Gambar 5.17.1 Tampilan ketika menukar urutan lagu

```
>> PLAYLIST SWAP 1 3 5;
Berhasil menukar lagu dengan nama "Bet You Wanna (Feat. Cardi B)" dengan "Crying Lightning" di playlist "GIVARI"
```


5.18 Data Test **PLAYLIST REMOVE**

Pengguna program dapat memanggil *command* PLAYLIST REMOVE untuk menghapus lagu pada playlist yang terdaftar.

Gambar 5.18.1 Tampilan ketika menghapus lagu

```
>> PLAYLIST REMOVE 1 6;  
Lagu "Ice Cream (with Selena Gomez)" oleh "BLACKPINK" telah dihapus dari playlist "GIVARI"!
```

5.19 Data Test **PLAYLIST DELETE**

Pengguna program dapat memanggil *command* PLAYLIST DELETE untuk menghapus suatu playlist yang terdaftar dalam daftar playlist. Pesan *error* akan ditampilkan jika inputan pengguna tidak *valid* pada setiap permintaan inputan.

Gambar 5.19.1 Tampilan ketika playlist dihapus

```
>> PLAYLIST DELETE;  
Daftar Playlist Pengguna :  
1. GIVARI  
  
Masukkan ID Playlist yang dipilih : 1;  
  
Playlist ID 1 dengan judul "GIVARI" berhasil dihapus.
```

5.20 Data Test **STATUS**

Pengguna program dapat memanggil *command* STATUS untuk menampilkan lagu yang sedang dimainkan beserta *queue song* yang ada dan dari playlist mana lagu tersebut diputar.

Gambar 5.20.1 Tampilan ketika menampilkan status

```
>> STATUS;  
Now Playing :  
BLACKPINK - How You Like That - THE ALBUM  
  
Queue :  
1. BLACKPINK - How You Like That - THE ALBUM  
2. BLACKPINK - Ice Cream (with Selena Gomez) - THE ALBUM  
3. BLACKPINK - Bet You Wanna (Feat. Cardi B) - THE ALBUM  
4. Arctic Monkeys - My Propeller - Humbug  
5. Arctic Monkeys - Crying Lightning - Humbug  
6. Arctic Monkeys - Secret Door - Humbug  
7. Arctic Monkeys - Pretty Visitors - Humbug
```

5.21 Data Test **SAVE**

Pengguna program dapat memanggil *command* SAVE untuk menyimpan *state* aplikasi terbaru. SAVE dapat dilakukan dengan dua cara yaitu langsung menginput SAVE <filename> pada *command* atau SAVE setelah melakukan inputan QUIT pada *command* program. Pada kasus jika <filename> tidak diberikan, maka akan dilakukan *autosave* sehingga dihasilkan *file*

simpanan *state* terakhir dengan nama “auto_save.txt”. Pada kasus <filename> yang diberikan tidak mengandung ekstensi “.txt” di belakangnya, maka SAVE akan secara otomatis menambahkan “.txt” di belakangnya.

Gambar 5.21.1 Pemanggilan SAVE <filename> secara langsung

```
===== Selamat Datang di =====
          W a y a n g   W a v e
===== Enjoy The Music =====

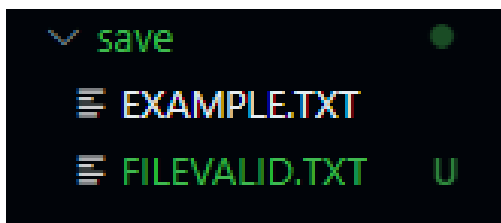
>> START;

File konfigurasi aplikasi berhasil dibaca. WayangWave berhasil dijalankan.

>> SAVE FILEVALID.TXT;
Save file berhasil disimpan.

>> |
```

Gambar 5.21.2 Hasil pemanggilan SAVE <filename> secara langsung



Gambar 5.21.3 Pemanggilan SAVE dari QUIT

```
===== Enjoy The Music =====

>> START;

File konfigurasi aplikasi berhasil dibaca. WayangWave berhasil dijalankan.

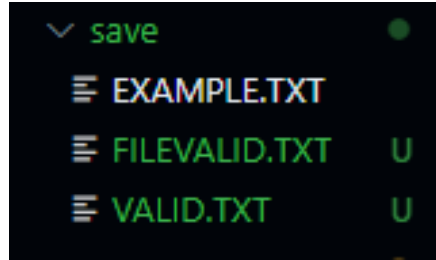
>> QUIT;
Apakah kamu ingin menyimpan data sesi sekarang? (Y/N) Y;
Masukkan nama file untuk penyimpanan: VALID.TXT;
Save file berhasil disimpan.

===== Kamu Keluar dari Wayang Wave =====

\\      =o)
(o>     /\
_(( ) Sampai Jumpa  \_V_
//              //

===== Keep Vibing =====
dama@LAPTOP-NNOC7JIU:/mnt/d/OneDrive - Institut Teknologi Bandung/01-Akademis/02-Assignments/Semester 3/IF2111 - Alstrukdat/Tubes/tubes_al
```

Gambar 5.21.4 Hasil pemanggilan SAVE dari Quit



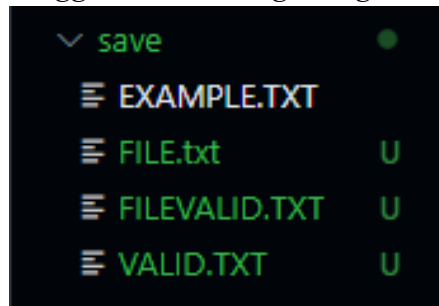
Gambar 5.21.5 Pemanggilan SAVE dengan argumen <filename> tanpa “.txt”

```
>> START;

File konfigurasi aplikasi berhasil dibaca. WayangWave berhasil dijalankan.

>> SAVE FILE;
Save file berhasil disimpan.
```

Gambar 5.21.6 Hasil pemanggilan SAVE dengan argumen <filename> tanpa “.txt”



Gambar 5.21.7 Hasil pemanggilan SAVE tanpa argumen <filename>

```
>> SAVE;

Argumen filename belum diberikan!
Autosaving dengan nama auto_save.txt...

Save file berhasil disimpan.

>> |
```

5.22 Data Test QUIT

Pengguna program dapat memanggil *command* QUIT untuk keluar dari sesi aplikasi. Namun, sebelum keluar dari aplikasi, program akan menanyakan apakah pengguna ingin melakukan SAVE terhadap aplikasi yang telah dijalankan. Jika pengguna program memilih melakukan SAVE, maka perubahan pada aplikasi akan disimpan pada suatu *file* yang diinginkan pengguna. Jika pengguna program memilih untuk tidak melakukan SAVE, maka perubahan pada aplikasi tidak akan disimpan.

Gambar 5.22.1 Pemanggilan QUIT dengan SAVE

```

===== ♪ Enjoy The Music ♪ =====

>> START;

File konfigurasi aplikasi berhasil dibaca. WayangWave berhasil dijalankan.

>> QUIT;
Apakah kamu ingin menyimpan data sesi sekarang? (Y/N) Y;
Masukkan nama file untuk penyimpanan: HEHE;
Save file berhasil disimpan.

===== Kamu Keluar dari Wayang Wave =====

  \\\              ~o)
 (o>             /\
_(()_Sampai Jumpa  \V_
  //              //

===== ♪ Keep Vibing ♪ =====
dama@LAPTOP-NNOC7JIU:/mnt/d/OneDrive - Institut Teknologi Bandung/01-Akademis/02-Assignments/Semester 3/IF2111 - Alstrukdat/Tubes/tubes_alstrukdatSTI_2023/bin$

```

Gambar 5.22.2 Pemanggilan QUIT tanpa SAVE

```

===== ♪ Enjoy The Music ♪ =====

>> START;

File konfigurasi aplikasi berhasil dibaca. WayangWave berhasil dijalankan.

>> QUIT;
Apakah kamu ingin menyimpan data sesi sekarang? (Y/N) N;

===== Kamu Keluar dari Wayang Wave =====

  \\\              ~o)
 (o>             /\
_(()_Sampai Jumpa  \V_
  //              //

===== ♪ Keep Vibing ♪ =====
dama@LAPTOP-NNOC7JIU:/mnt/d/OneDrive - Institut Teknologi Bandung/01-Akademis/02-Assignments/Semester 3/IF2111 - Alstrukdat/Tubes/tubes_alstrukdatSTI_2023/bin$

```

5.23 Data Test HELP

Pengguna program dapat memanggil *command* HELP untuk menunjukkan daftar *command* yang dapat digunakan di program beserta fungsinya. *Command* HELP bisa dipanggil sebelum atau setelah *command* START atau LOAD.

Gambar 5.23.1 Pemanggilan HELP sebelum START atau LOAD

```

>> HELP;

===.+:*+.::=.+:*+.::=.+:*+.::=[ MENU HELP WAYANGWAVE ]==.+:*+.::=.+:*+.::=.+:*+:
|      1. START -> Untuk masuk sesi baru      |
|      2. LOAD  -> Untuk memulai sesi berdasarkan file konfigurasi  |
|_____|

```

Gambar 5.23.2 Pemanggilan HELP setelah START atau LOAD

```
>> HELP;
```

```
===.+*+.==.+*+.==.+*+=.+*+=.+*+.==.+*+.==.+*+.==.+*+. [ MENU HELP WAYANGWAVE ]==.+*+=.+*+.==.+*+=.+*+.==.+*+.==.+*+.==.+*+.==.+*+.
```

```
1. LIST -> Untuk menampilkan daftar lagu, playlist, album, penyanyi
```

```
2. PLAY -> Untuk memulai suatu lagu baru atau lagu dari playlist yang dipilih
```

```
3. QUEUE -> Untuk memanipulasi queue lagu dengan command tambah song, tambah playlist, tukar, hapus, atau mengosongkan
```

```
4. SONG -> Untuk mengontrol navigasi lagu dalam queue (putar lagu selanjutnya atau lagu sebelumnya)
```

```
5. PLAYLIST -> Untuk melakukan operasi pada playlis, seperti membuat, menambahkan, menukar, menghapus lagu, atau menghapus playlist
```

```
6. STATUS -> Untuk menampilkan lagu yang sedang dimainkan beserta Queue song yang ada dan dari playlist mana lagu itu diputar
```

```
7. SAVE -> Untuk menyimpan state aplikasi terbaru ke dalam suatu file
```

```
8. QUIT -> Untuk keluar dari sesi
```

6 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	START	Memeriksa apakah file config default berhasil dibaca	Menginput <i>command</i> START pada saat pertama kali memulai program WayangWave	START	Gambar 5.1.1	Sesuai dengan hasil yang diharapkan
2	LOAD	Memeriksa apakah <i>file</i> berhasil dibaca	Memberikan <i>command</i> LOAD diikuti nama <i>file</i> yang <i>valid</i>	LOAD <filename>	Gambar 5.2.1 Gambar 5.2.2	Sesuai dengan hasil yang diharapkan
3	LIST DEFAULT	Menampilka n list penyanyi yang ada dan dapat memilih untuk melihat album dari penyanyi yang dipilih. Kemudian melihat lagu yang ada dari album yang dipilih.	Mengetik LIST DEFAULT pada <i>command</i> setelah START atau LOAD, lalu terdapat konfirmasi jika ingin melihat album dan lagu di dalam album.	LIST DEFAULT	Gambar 5.3.1	Sesuai dengan hasil yang diharapkan
4	LIST PLAYLIST	Menampilka n list playlist yang dimiliki pengguna.	Mengetik LIST DEFAULT pada <i>command</i> setelah START atau LOAD.	LIST PLAYLIST	Gambar 5.4.1 Gambar 5.4.2	Sesuai dengan hasil yang diharapkan
5	PLAY SONG	Memainkan lagu berdasarkan nama penyanyi,	Mengetik LIST DEFAULT pada <i>command</i> setelah START atau LOAD, lalu muncul list penyanyi untuk input	PLAY SONG	Gambar 5.5.1	Sesuai dengan hasil yang diharapkan

		album, dan ID lagu.	nama penyanyi, list album untuk input nama album, list lagu suatu album untuk input ID lagu.			
6	PLAY PLAYLIST	Memainkan lagu berdasarkan ID playlist.	Mengetik LIST DEFAULT pada <i>command</i> setelah START atau LOAD, lalu muncul list playlist yang dimiliki dan diminta input ID playlist.	PLAY PLAYLIST	Gambar 5.6.1 Gambar 5.6.2	Sesuai dengan hasil yang diharapkan
7	QUEUE SONG	Memeriksa apakah lagu yang dipilih berhasil ditambahkan ke queue	Mengetik QUEUE SONG pada <i>command</i> setelah START atau LOAD, lalu memilih penyanyi, judul album, ID lagu yang ingin ditambahkan ke queue	QUEUE SONG	Gambar 5.7.1 Gambar 5.7.2 Gambar 5.7.3 Gambar 5.7.4	Sesuai dengan hasil yang diharapkan
8	QUEUE PLAYLIST	Memeriksa apakah playlist yang dipilih berhasil ditambahkan ke queue	Mengetik QUEUE PLAYLIST pada <i>command</i> setelah START atau LOAD, lalu memilih ID playlist yang ingin ditambahkan ke queue	QUEUE PLAYLIST	Gambar 5.8.1 Gambar 5.8.2	Sesuai dengan hasil yang diharapkan
9	QUEUE SWAP	Memeriksa apakah urutan lagu berhasil ditukar di queue	Mengetik QUEUE SWAP pada <i>command</i> setelah START atau LOAD, lalu diikuti urutan lagu yang ingin ditukar pada queue	QUEUE SWAP <x> <y>	Gambar 5.9.1 Gambar 5.9.2 Gambar 5.9.3 Gambar 5.9.4	Sesuai dengan hasil yang diharapkan
10	QUEUE REMOVE	Memeriksa apakah lagu yang dipilih dalam queue berhasil dihapus	Mengetik QUEUE REMOVE pada <i>command</i> setelah START atau LOAD, lalu diikuti urutan lagu yang ingin dihapus pada queue	QUEUE REMOVE <id>	Gambar 5.10.1 Gambar 5.10.2 Gambar 5.10.3	Sesuai dengan hasil yang diharapkan
11	QUEUE CLEAR	Memeriksa apakah queue berhasil dikosongkan	Mengetik QUEUE CLEAR pada <i>command</i> setelah START atau LOAD	QUEUE CLEAR	Gambar 5.11.1	Sesuai dengan hasil yang diharapkan
12	SONG NEXT	Memainkan lagu yang tersimpan di <i>queue</i> dengan urutan pertama	Mengetik SONG NEXT pada <i>command</i> setelah START atau LOAD, QUEUE SONG atau PLAY SONG. Apabila terdapat lagu yang dimainkan sesuai dengan STATUS ('Now Playing') dan otomatis <i>queue</i> kosong (belum memulai <i>command</i> QUEUE SONG), SONG NEXT	SONG NEXT	Gambar 5.12.1 Gambar 5.12.2 Gambar 5.12.3	Sesuai dengan hasil yang diharapkan

			dapat dijalankan dengan tetap memutar lagu yang sedang dimainkan. Apabila hanya dengan QUEUE SONG, SONG NEXT dapat dijalankan dengan memutar lagu berdasarkan urutan pada <i>queue</i> .			
13	SONG PREV	Memainkan lagu yang tersimpan di riwayat dengan urutan terakhir atau lagu yang sebelumnya dimainkan	Mengetik SONG PREV pada <i>command</i> setelah START atau LOAD, QUEUE SONG kemudian SONG NEXT, atau PLAY SONG. Apabila terdapat lagu yang dimainkan, otomatis <i>queue</i> dan riwayat kosong, maka akan memutar lagu yang sedang dimainkan. Apabila hanya QUEUE SONG dan SONG NEXT, maka <i>command</i> SONG PREV akan memutar lagu berdasarkan urutan pada riwayat (dari yang paling terakhir dimainkan).	SONG PREV	Gambar 5.13.1 Gambar 5.13.2 Gambar 5.13.3	Sesuai dengan hasil yang diharapkan
14	PLAYLIST CREATE	Memeriksa apakah playlist berhasil dibuat	Mengetik PLAYLIST CREATE pada <i>command</i> setelah START atau LOAD, lalu input nama playlist yang ingin dibuat (minimal 3 karakter)	PLAYLIST CREATE	Gambar 5.14.1 Gambar 5.14.2	Sesuai dengan hasil yang diharapkan
15	PLAYLIST ADD SONG	Memeriksa apakah lagu yang dipilih berhasil ditambahkan ke playlist	Mengetik PLAYLIST ADD SONG pada <i>command</i> setelah START atau LOAD, lalu memilih penyanyi, judul album, ID lagu yang ingin ditambahkan ke playlist tujuan	PLAYLIST ADD SONG	Gambar 5.15.1	Sesuai dengan hasil yang diharapkan
16	PLAYLIST ADD ALBUM	Memeriksa apakah album yang dipilih berhasil ditambahkan ke playlist	Mengetik PLAYLIST ADD ALBUM pada <i>command</i> setelah START atau LOAD, lalu memilih penyanyi dan judul album yang ingin ditambahkan ke playlist tujuan	PLAYLIST ADD ALBUM	Gambar 5.16.1	Sesuai dengan hasil yang diharapkan
17	PLAYLIST SWAP	Memeriksa apakah urutan lagu berhasil	Mengetik PLAYLIST SWAP pada <i>command</i> setelah START atau LOAD, lalu diikuti	PLAYLIST SWAP <id> <x> <y>	Gambar 5.17.1	Sesuai dengan hasil yang diharapkan

		ditukar di playlist	inputan ID playlist dan urutan lagu yang ingin ditukar pada playlist			
18	PLAYLIST REMOVE	Memeriksa apakah lagu yang dipilih dalam playlist berhasil dihapus	Mengetik PLAYLIST REMOVE pada <i>command</i> setelah START atau LOAD, lalu diikuti inputan ID playlist dan urutan lagu yang ingin dihapus pada playlist	PLAYLIST REMOVE <id> <n>	Gambar 5.18.1	Sesuai dengan hasil yang diharapkan
19	PLAYLIST DELETE	Memeriksa apakah playlist yang dipilih berhasil dihapus	Mengetik PLAYLIST DELETE pada <i>command</i> setelah START atau LOAD, lalu berikan ID playlist yang ingin dihapus	PLAYLIST DELETE	Gambar 5.19.1	Sesuai dengan hasil yang diharapkan
20	STATUS	Memeriksa kevalidan antara status yang ditampilkan dengan lagu yang diputar	Mengetik STATUS pada <i>command</i> setelah START atau LOAD	STATUS	Gambar 5.20.1	Sesuai dengan hasil yang diharapkan
21	SAVE	Melakukan <i>save file</i>	Mengetik SAVE <filename> pada <i>command</i> setelah START atau LOAD	SAVE <filename>	Gambar 5.21.1 Gambar 5.21.2 Gambar 5.21.3 Gambar 5.21.4 Gambar 5.21.5 Gambar 5.21.6	Sesuai dengan hasil yang diharapkan
22	QUIT	Memberikan pilihan untuk melakukan <i>save file</i> atau keluar dari program	Mengetik QUIT pada <i>command</i> setelah START atau LOAD	QUIT	Gambar 5.22.1 Gambar 5.22.2	Sesuai dengan hasil yang diharapkan
23	HELP	Menampilkan daftar <i>command</i> dan fungsinya yang dapat dijalankan pada program	Mengetik HELP pada <i>command</i> setelah atau sebelum START atau LOAD.	HELP	Gambar 5.23.1 Gambar 5.23.2	Sesuai dengan hasil yang diharapkan

7 Pembagian Kerja dalam Kelompok

<i>Nama Anggota Kelompok - NIM</i>	<i>Pembagian Kerja</i>
Thalita Zahra Sutejo - 18222023	<ul style="list-style-type: none"> - Mengerjakan pembuatan algoritma, <i>data test</i> dan <i>test script</i> pada laporan dalam aplikasi WayangWave bagian <i>queue (next, previous, swap, remove, clear)</i> dan <i>song (previous, next)</i>. - Membuat ADT Lagu dan tambahan ADT <i>Queue</i> - Menyiapkan laporan dan mengerjakan laporan bagian ringkasan, program utama, stuktur data queue, struktur data lagu, dan notulensi.
Alfandito Rais Akbar - 18222037	<ul style="list-style-type: none"> - Mengerjakan <i>COMMAND</i> aplikasi WayangWave bagian <i>start</i> dan file <i>main</i>. - Implementasi ADT Mesin Kata dan ADT Mesin Karakter untuk <i>Command</i> Input dan membaca file. - Mengerjakan laporan bagian struktur data mesin kata, struktur data mesin karakter, <i>data test</i>, dan <i>test script</i> start.
Givari Al Fachri - 18222045	<ul style="list-style-type: none"> - Mengerjakan <i>COMMAND</i> aplikasi WayangWave bagian <i>playlist</i> dan <i>status</i> - Membuat ADT Lagu (Program) - Membuat driver ADT linked list dan driver ADT list dinamis - Mengerjakan laporan bagian struktur data linked list, struktur data list dinamis, data test, dan data script
Dama Dhananjaya Dalimam - 18222047	<ul style="list-style-type: none"> - Membuat <i>repository github</i> dan menambahkan ADT yang diperlukan untuk aplikasi WayangWave - Mengerjakan <i>COMMAND</i> aplikasi

	WayangWave dan laporan bagian <i>load, save, quit</i> <ul style="list-style-type: none"> - Testing di WSL - Pembuatan <i>shell script</i> dan <i>batch file</i> untuk <i>compile</i> dan <i>run</i> program
Satria Wisnu Wibowo - 18222087	<ul style="list-style-type: none"> - Mengerjakan <i>COMMAND</i> aplikasi bagian <i>list</i> dan <i>play</i> - Mengerjakan laporan bagian struktur data list statis, struktur data map, struktur data set, data test list, dan data test play.

8 Lampiran

8.1 Deskripsi Tugas Besar

Bondowoso, seorang remaja yang unik, lahir dari kecintaannya pada lagu-lagu dari boombox kesayangannya, mengalami kesulitan dalam menjalin persahabatan karena penampilannya yang menakutkan. Satu-satunya cinta dalam hidupnya, Roro Jonggrang, hanya tergila-gila pada lagu hip-hop dari walkman-nya. Dengan keahlian musik yang luar biasa, Bondowoso berusaha menarik hati Roro melalui WayangWave, sebuah aplikasi yang dibuat bersama teman-teman kasat matanya. Dengan niat baik, mereka berusaha memperkenalkan Roro pada keberagaman musik, menghasilkan kisah romantis yang menggambarkan semangat dan kreativitas di tengah perbedaan.

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.

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 <id> <n>

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.







8.2 Notulen Asistensi

8.2.1 Notulensi Asistensi 1

Tanggal : 30 Oktober 2023	Catatan Asistensi: Dijelaskannya aspek-aspek penting dan penjelasan tambahan berdasarkan spesifikasi yang telah diberikan terkait Tugas Besar IF2111 Algoritma dan Struktur Data. Pertanyaan 1: Bagaimana teknis penyimpanan struktur <i>file</i> ? Jawaban: Struktur file (<i>console.c</i>) diisi dengan fungsi-fungsi dari semua <i>command</i> dan program main akan memanggil fungsi dari <i>console.c</i> . Dengan begitu, agar mempermudah dan tidak menumpuk di program main, dibuat <i>console.c</i> . Pertanyaan 2: Terkait harus jalan di linux, ada keterangan dan spesifikasi tambahan atau tidak? Jawaban: Bikin di <i>windows</i> dulu tidak apa-apa, tetapi kelompok harus tes di linux. Apabila tiba-tiba program error, maka perlu diperbaiki. Catatan terakhir: Jangan lupa masukkan username asisten ke github (username: apin014)
Tempat : Zoom Meeting	
Kehadiran Anggota Kelompok:	
1 18222023 	
2 18222037 	
3 18222045 	
4 18222047	

 5 18222087  6	
	Tanda Tangan Asisten: 

8.2.1 Notulensi Asistensi 2

Tanggal :15 November 2023	Catatan Asistensi:
Tempat : Zoom Meeting	Asistensi 2 dimulai dengan presentasi progress yang telah dikerjakan oleh Kelompok 9. (Telah di presentasikan oleh lima anggota kelompok mengenai bagian masing-masing, khususnya <i>COMMAND</i>)
<p>Kehadiran Anggota Kelompok:</p> <p>1 18222023</p>  <p>2 18222037</p>  <p>3 18222045</p>  <p>4 18222047</p>  <p>5 18222087</p> 	<p>Pertanyaan 1: Bagaimana cara menyimpan <i>playlist</i>, karena dalam pembuatan masih sedikit bingung dengan pengimplementasiannya. Jawaban: Disimpan dengan array dinamis.</p> <p>Pertanyaan 2: Bagaimana dengan teknis tidak boleh memakai scanf? (Harus dengan STARTWORD) Jawaban: Lihat referensi <i>code</i> yang sudah ada sebagai contoh pengimplementasian yang tidak menggunakan scanf.</p> <p>Pernyataan tambahan dari asisten:</p> <ol style="list-style-type: none"> 1. Hati-hati dan perhatikan <i>input COMMAND</i> 2. Untuk Queue, ingat dan perhatikan, apabila Queue Playlist, diperhatikan urutannya berdasarkan <i>Song</i> yang ada dalam <i>Playlist</i>. Untuk Song, apabila mengakses <i>song previous</i>, lagu yang menjadi TOP dari Stack previous Song disimpan kemudian dijadikan head dari Queue, sedangkan lagu yang <i>on play (onPlaySong)</i> dimasukkan ke Stack <i>previous Song</i> dan menjadi TOP dari Stack <i>previous Song</i>.
	<p>Tanda Tangan Asisten:</p> 

8.3 Log Activity Anggota Kelompok

No.	Tanggal	NIM	Nama	Aktivitas
1	29/10/2023	18222047	Dama Dhananjaya Daliman	Membuat repository github
2	31/10/2023	18222047	Dama Dhananjaya Daliman	Push ADT List, ADT Mesin Karakter, ADT Mesin Kata, ADT Queue, ADT Stack
3	03/11/2023	18222047	Dama Dhananjaya Daliman	Push ADT Map, ADT Set, ADT Hashmap. Push <i>command</i> QUIT
4	05/11/2023	18222047	Dama Dhananjaya Daliman	Push ADT Linked List
5	10/11/2023	18222037	Alfandito Rais Akbar	Menambahkan ADT Mesin Kata dan Mesin Karakter
6	11/11/2023	18222037	Alfandito Rais Akbar	Menambahkan file konfigurasi default dan memperbaiki ADT linked list, list statis, mesin karakter, dan mesin kata
7	13/11/2023	18222037	Alfandito Rais Akbar	Memperbaharui beberapa ADT, menambahkan command START, memperbaharui main, dan memperbaiki console.c
8	17/11/2023	18222045	Givari Al Fachri	Menambahkan command-command playlist
9	19/11/2023	18222045	Givari Al Fachri	Memperbaharui command-command playlist

10	20/11/2023	18222023	Thalita Zahra Sutejo	Menambahkan command help dan song, memperbaiki circular include, dan menambahkan command-command queue.
11	20/11/2023	18222045	Givari Al Fachri	Memperbaharui ADT queue dan command status
12	20/11/2023	18222087	Satra Wisnu Wibowo	Memperbaharui console.c
13	21/11/2023	18222023	Thalita Zahra Sutejo	Menambahkan enqueue song, driver queue dan song, dan menambahkan komentar deskriptif
14	21/11/2023	18222047	Dama Dhananjaya Daliman	Menyamakan tipe dan struktur data
15	21/11/2023	18222045	Alfandito Rais Akbar	Memperbaiki ADT dan Main
16	22/11/2023	18222087	Satria Wisnu Wibowo	Mengubah playsong dan playplaylist
17	22/11/2023	18222037	Alfandito Rais Akbar	Memperbaiki console
18	22/11/2023	18222047	Dama Dhananjaya Daliman	Menambahkan command save dan memperbaiki playlist
19	22/11/2023	18222023	Thalita Zahra Sutejo	Memperbaiki queue song, queue swap, queue remove, dan help.
20	22/11/2023	18222045	Givari Al Fachri	Memperbaharui playlist
21	23/11/2023	18222087	Satria Wisnu Wibowo	Memperbaiki play

				playlist
22	23/11/2023	18222023	Thalita Zahra Sutejo	Memperbaiki console
23	23/11/2023	18222047	Dama Dhananjaya Daliman	Memperbaiki command-command queue dan memperbaharui command save dan load.
24	24/11/2023	18222047	Dama Dhananjaya Daliman	Menambahkan banner, membuat batch file, memperbaharui readme, memperbaiki concat kata dan save, menambahkan “\r” handling, dan memperbaiki command load.
25	24/11/2023	18222045	Givari Al Fachri	Menambahkan driver array dinamis dan memperbaharui console.c