**LAPORAN TUGAS BESAR**

**IF211**1 **Algoritma dan Struktur Data STI**

WayangWave

Dipersiapkan oleh:

Kelompok 4

1. Alessandro Jusack Hasian 18222025
2. Billy Samuel Setiawan 18222039
3. Micky Valentino 18222093
4. Sitanggang Jerry J.P 18220082

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

|  | **Sekolah Teknik Elektro dan Informatika ITB** | **Nomor Dokumen** | | **Halaman** |
| --- | --- | --- | --- | --- |
| *IF2111-TB-K01-04* | | *<jml hlm>* |
| *Revisi* | *<no revisi>* | *<Tgl release>* |

**Daftar Isi**

[1 Ringkasan 2](#_heading=h.gjdgxs)

[2 Penjelasan Tambahan Spesifikasi Tugas 4](#_heading=)

[3 Struktur Data (ADT) 4](#_heading=)

[3.1 LinkedList 4](#_heading=)

[3.2 List 4](#_heading=)

[3.3 Map 5](#_heading=h.k8xppoz49os2)

[3.4 Mesin Karakter dan Kata 5](#_heading=h.wd0svyk3ln1u)

[3.5 Queue 5](#_heading=h.tllm5lzb7xk)

[3.6 Set 6](#_heading=h.xn15tib6u3c5)

[3.7 Stack 6](#_heading=h.eapmyelcaeam)

[4 Program Utama 6](#_heading=)

[5 Algoritma-Algoritma Menarik 6](#_heading=)

[5.1 <Algoritma 1> 7](#_heading=)

[5.2 <Algoritma 2> 7](#_heading=)

[6 Data Test 7](#_heading=)

[6.1 Data Test START 7](#_heading=)

[6.2 Data Test LOAD 7](#_heading=)

[6.3 Data Test List 7](#_heading=h.pns9sp74yut4)

[6.4 Data Test Play 8](#_heading=h.c3jwtvajwht)

[6.5 Data Test Queue 8](#_heading=h.emylnddmo2to)

[6.6 Data Test Song 8](#_heading=h.o85fwi8vetzv)

[6.7 Data Test Playlist 8](#_heading=h.kpas11mlsug)

[6.8 Data Test Status 9](#_heading=h.s7vt928j2idt)

[6.9 Data Test Save 9](#_heading=h.3kz4los6xr3h)

[6.10 Data Test Quit 9](#_heading=h.z9rsjlr7rvkd)

[6.11 Data Test Help 9](#_heading=h.3c2qfsce09ln)

[6.12 Data Test <INVALID\_COMMAND> 9](#_heading=h.cd2xyhbfo5s5)

[7 Test Script 9](#_heading=)

[8 Pembagian Kerja dalam Kelompok 9](#_heading=)

[9 Lampiran 10](#_heading=h.2jxsxqh)

[9.1 Deskripsi Tugas Besar 10](#_heading=)

[9.2 Notulen Rapat 12](#_heading=)

[9.3 Log Activity Anggota Kelompok 17](#_heading=)

# **Ringkasan**

Pada tugas besar Algoritma dan Struktur Data kali ini, kami diminta untuk membuat program tentang aplikasi yang dapat menggantikan perangkat lunak di *walkman* Roro, yaitu WayangWave. Aplikasi ini memiliki simulasi berbasis CLI (*command-line interface*) dengan menggunakan bahasa C serta menggunakan ADT yang sudah dipelajari di mata kuliah Algoritma dan Stuktur Data. WayangWave merupakan aplikasi yang dapat mensimulasikan service pemutaran musik dengan fitur - fitur utama berupa memutar lagu, menampilkan daftar lagu, membuat dan menghapus playlist, mengatur urutan dimainkannya lagu dan menampilkan status dari aplikasi. WayangWave ini memiliki beberapa fungsi utama yang menunjang fitur - fitur tersebut, yaitu:

1. Start Menu

Program WayangWave dimulai dengan welcome page dan bisa menjalankan fungsi START yang memulai WayangWave tanpa adanya data antrean, riwayat lagu, dan playlist yang dibuat peserta, LOAD yang memulai WayangWave dengan adanya data antrean dan riwayat lagu serta playlist yang telah dibuat oleh peserta. WayangWave juga bisa diberikan perintah HELP untuk memberikan penjelasan tentang START dan LOAD.

1. Main Page

Setelah melakukan START atau LOAD, user memasuki sesi sehingga dapat melakukan fitur utama dari WayangWave adalah memutar lagu ataupun playlist, menambah antrean lagu atau playlist, memutar lagu selanjutnya atau lagu sebelumnya, membuat playlist, . Aplikasi WayangWave juga dapat mencetak penyanyi album lagu yang ada di WayangWave maupun di playlist yang dibuat oleh pengguna. WayangWave dapat menampilkan lagu yang sedang diputar dan antrean lagu yang ada.

1. Penyimpanan

User dapat menyimpan antrean lagu, riwayat lagu, dan playlist yang telah dibuat oleh user menggunakan fungsi SAVE dengan nama file yang dituju. File tersebut akan dibuat baru dan akan menyimpan database penyanyi, album, dan judul lagu, serta antrean lagu, riwayat lagu, dan playlist yang dibuat oleh pengguna. File ini akan disimpan dalam bentuk.txt dan di simpan dalam folder save.

# **Penjelasan Tambahan Spesifikasi Tugas**

Pada status, terdapat penjelasan bahwa current playlist hanya ada saat tidak ada lagu selain dari playlist, sehingga kami berasumsi bahwa saat kita play playlist, setiap kita menambah lagu dengan queue song ataupun queue playlist dan memulai lagu dengan play song ataupun play playlist, maka current playlist akan hilang. Selain itu, berdasarkan asistensi 1 yang terdapat pada lampiran, jika dilakukan play playlist, kami tidak menambahkan semua lagu playlist ke dalam stack secara langsung tetapi sesuai dengan urutan (dengan song next atau song previous).

# **Struktur Data (ADT)**

Untuk mengerjakan aplikasi WayangWave, kelompok kami memerlukan berbagai ADT untuk dipakai secara bersamaan. Kelompok kami menggunakan ADT yang telah dipelajari dalam perkuliahan, seperti Linked List untuk playlist yang dibuat pengguna, List untuk menyimpan penyanyi - penyanyi yang ada, Map untuk menyimpan album - album yang ada, Set untuk menyimpan lagu - lagu yang ada, Mesin Karakter untuk membaca input dari user, Mesin Kata untuk membaca input dari user, Queue untuk menyimpan antrean lagu WayangWave, dan Stack untuk menyimpan riwayat lagu WayangWave.

## LinkedList

LinkedList digunakan untuk merepresentasikan playlist yang dibuat oleh pengguna. Dalam LinkedList terdapat nama dari playlist yang disimpan dalam variabel “nama” bertipe Word serta address lagu - lagu dalam sebuah playlist yang disimpan dalam addressPlaylist. Alasan kami memilih ADT ini adalah playlist merupakan suatu kumpulan lagu yang kontigu serta kita bisa menambahkan lagu secara bebas karena LinkedList tidak memiliki batasan memori. Implementasi LinkedList terdapat pada file listlinier.c dan memiliki header di listlinier.h

## List

List digunakan untuk merepresentasikan penyanyi yang ada dalam data yang disediakan serta daftar playlist yang dibuat oleh pengguna. Dalam penyanyi, kami menggunakan list statik karena database yang kami buat tidak akan bertambah sehingga tidak perlu mengalokasi memori tambahan. Pada penyanyi, terdapat jumlah penyanyi yang ada yang disimpan dalam variabel “JumlahPenyanyi” bertipe integer serta penyanyi yang disimpan dalam variabel Penyanyi yang bertipe array of PenyanyiType yang merupakan struktur List Statik. Dalam Playlist, kami menggunakan list dinamis karena pengguna dapat menambahkan playlist sehingga tidak ada batasan memori yang pasti untuk mengalokasi semua playlist. Pada Playlist Alasan kami memilih ADT ini adalah Penyanyi merupakan sebuah data yang kontigu dan pada aplikasi ini tetap. Pada Playlist, pengguna dapat menambahkan playlist tanpa batas dan memiliki keterurutan sehingga list dinamis merupakan ADT yang tepat. Implementasi lists terdapat pada file list.c dan memiliki header di list.h

## Map

Map digunakan untuk merepresentasikan album - album yang dimiliki oleh seorang penyanyi. Dalam map, terdapat jumlah album yang dimiliki oleh satu penyanyi yang disimpan dalam variabel “JumlahAlbum” bertipe integer dan Isi album yang merupakan struktur Map berupa keytype bertipe Word yang merupakan nama album yang disimpan dalam variabel “NamaAlbum” dan SetSong (valuetype) yang berisi daftar lagu. Implementasi Map terdapat pada file Map.c dan memiliki header di Map.h.

## Mesin Karakter dan Kata

Mesin Karakter dan kata digunakan untuk membaca file konfigurasi default maupun save dari file.

## Queue

Queue digunakan untuk merepresentasikan antrean lagu pada Wayang Wave. Queue yang kami buat bertipe QueueSongType yang berisi 3 variabel dengan tipe Word yaitu, nama penyanyi yang disimpan dalam variabel “penyanyi”, nama album yang disimpan dalam variabel “album”, dan judul lagu yang disimpan dalam variabel “judul\_lagu” dalam antrean. Kelompok kami menggunakan ADT queue untuk merepresentasikan antrean lagu karena antrean lagu memiliki sifat FIFO (First In First Out) yang merupakan sifat yang dimiliki oleh ADT queue. Implementasi queue terdapat pada file queue.c dan memiliki header di queue.h

## Set

Set digunakan untuk merepresentasikan lagu - lagu yang berada dalam sebuah album. Set yang kami buat berisikan jumlah lagu yang ada dalam 1 album yang disimpan dalam variabel “JumlahLagu” serta judul dari setiap lagu yang ada dalam 1 album yang bertipe Word dan disimpan dalam variabel “Songs”. Kelompok kami menggunakan ADT set untuk merepresentasikan lagu - lagu dalam 1 album karena Setiap lagu dalam 1 album unik dan tidak bisa duplikasi judulnya. Implementasi setterdapat pada file set.c dan memiliki header di set.h

## Stack

Stack digunakan untuk merepresentasikan riwayat lagu yang telah diputar. Isi dari Stack ini sama seperti queue, yaitu QueueSongType. Kami memilih Stack untuk merepresentasikan riwayat lagu karena riwayat lagu memiliki sifat LIFO (Last In First Out) yang merupakan sifat yang dimiliki ADT stack. Implementasi stack terdapat pada file stack.c dan memiliki header di stack.h

# **Program Utama**

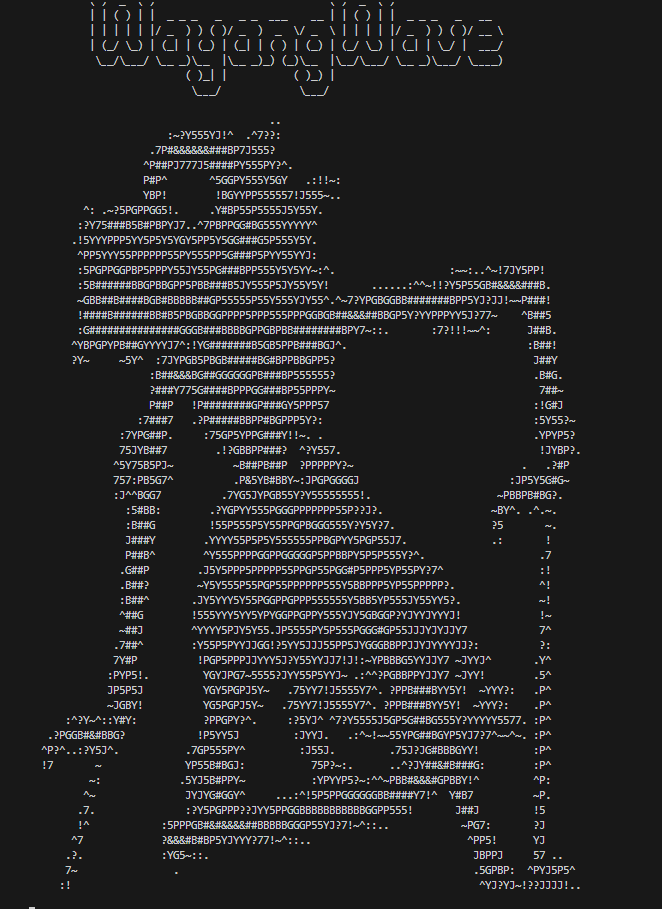
Program utama yang ada pada file “main.c” akan memasukkan semua file header dari ADT yang telah dibuat sebelumnya. Saat akan dijalankan, program utama akan menampilkan interface yang menerima input pengguna untuk memilih bagian program mana yang akan dijalankan. Setelah itu, program utama akan menampilkan START dan LOAD. Jika pengguna melalukan input START, program akan membaca file yang berisi jumlah penyanyi, jumlah album, dsb. Sementara, apabila pengguna memilih LOAD, program akan membaca file save yang telah disimpan sebelumnya. Jika program tidak menemukan file save, program akan mengeluarkan output yang menunjukkan jika file konfigurasi gagal berjalan.

# **Algoritma-Algoritma Menarik**

Isi dengan penjelasan mengenai algoritma-algoritma menarik yang Anda temukan atau Anda buat dalam Tugas Besar, jika ada. Berikan penjelasan singkat, di mana algoritma tersebut digunakan dan mengapa Anda anggap algoritma tersebut menarik.

## Prosedur art\_WayangWave()

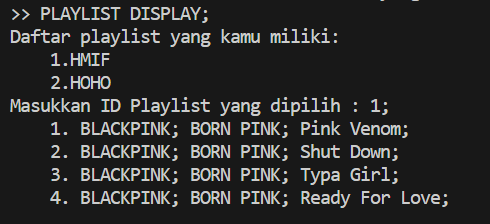
Prosedur ini dipanggil setiap kali program WayangWave dijalankan sebagai tampilan awal yang me-welcome user.



Gambar 1. Art WayangWave

## Command PLAYLIST DISPLAY

Command ini merupakan bagian dari command playlist yang berfungsi untuk menampilkan daftar playlist yang ada dan juga menampilkan lagu-lagu dalam playlist yang dipilih.



Gambar 2. Playlist Display

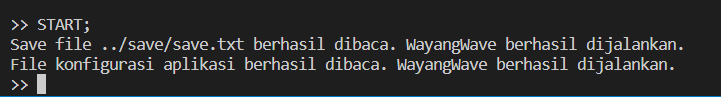
## Fungsi SentenceToWord

Fungsi ini menangkap suatu baris dari ADT WORD yang dalam berbentuk baris (include blank) kemudian memilih kata ke-index berapa yang diinginkan. Fungsi kemudian mereturn suatu Word. Fungsi terdapat dalam mesinkata.c

# **Data Test**

## Data Test START

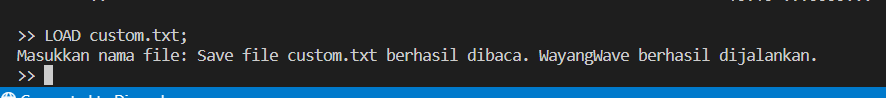
Pada data test Start, kami mengetes meng-compile semua fungsi dan ADT yang kami buat serta membaca file default yang ada di folder save dengan nama “save.txt”. Pembacaan file dilakukan dan di masukkan ke dalam ADT sesuai dengan formatnya. Hasil yang seharusnya diberikan adalah data berhasil dibaca, serta dimasukkan ke dalam Penyanyi, Album, dan Judul lagu sesuai dengan format data yang diberikan.



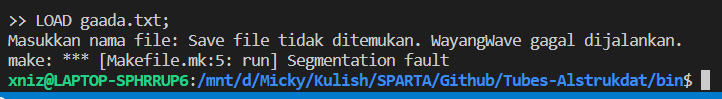
Gambar 3. Start

## Data Test LOAD

Pada data test Load, kami mengetes pembacaan file yang di save pengguna sebelum sesi dimulai. nama file unik sesuai dengan penamaan save file pengguna. Hasil yang diharapkan adalah fungsi dapat membuka file yang di save pengguna, membaca dan menuliskan kata - kata yang ada di file kedalam ADT penyanyi, album, judul lagu, antrean, riwayat lagu, dan playlist sesuai dengan format data yang diberikan.



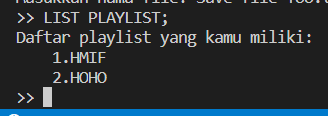
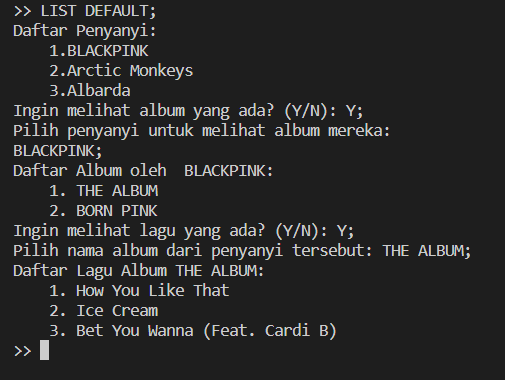
Gambar 4. Load berhasil



Gambar 5. Load Gagal

## Data Test List

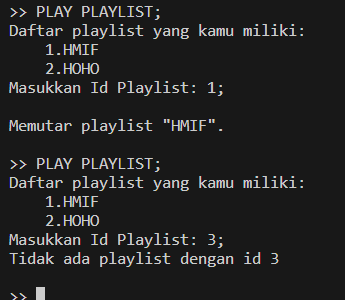
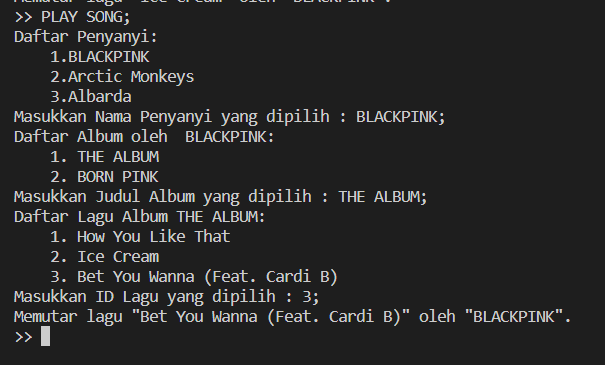
Pada data test list, terdapat 2 pengetesan, yaitu List Default dan List Playlist. Pada list default, kami mengetes pembacaan struktur data List, Map, dan Set yang merupakan penyanyi, album, dan judul lagu sesuai dengan inputan user, sedangkan pada List Playlist, kami mengetes pembacaan struktur data List dinamis dan Linked List yang merupakan Daftar playlist yang dibuat pengguna dan lagu - lagu yang terdapat pada playlist tersebut sesuai dengan inputan pengguna. Hasil yang seharusnya diberikan adalah penampilan penyanyi, album, dan judul atau daftar playlist dan lagu - lagu dalam playlist sesuai inputan user.



Gambar 6. List Default Gambar 7. List Playlist

## Data Test Play

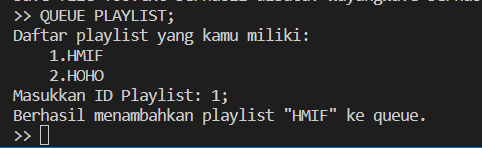
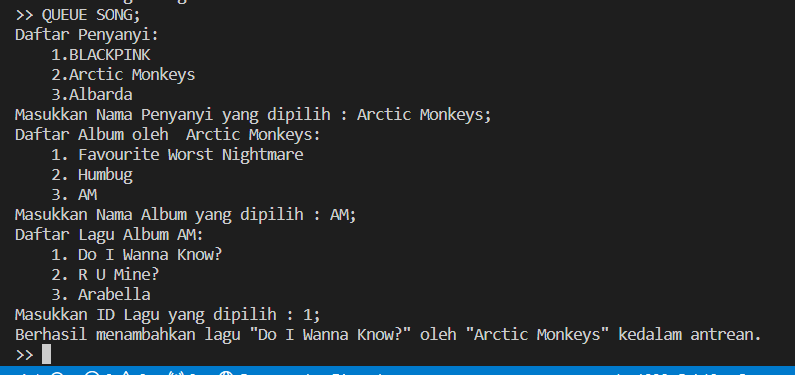
Pada data test play, terdapat 2 pengetesan yang kami lakukan, yaitu play song dan play playlist. Pada play song, kami mengetes pembacaan struktur List statik yang merupakan daftar penyanyi, Map yang merupakan daftar album, dan Set yang merupakan daftar lagu. Setelah pembacaan, kami melakukan pengetesan struktur Queue yang merupakan antrean lagu dengan menggunakan primitif enqueue untuk menambahkan lagu dan dequeue untuk menghapus lagu - lagu yang sebelumnya diputar. Kami juga melakukan pengetesan pada struktur Stack yang merupakan riwayat lagu dengan melakukan pop untuk menghapus riwayat lagu. Hasil yang seharusnya diberikan adalah semua antrean dan riwayat lagu sebelum play song atau play playlist dihapus dan semua song yang di play ditambahkan ke current playing dan antrean lagu.



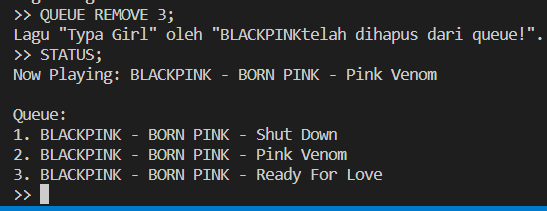
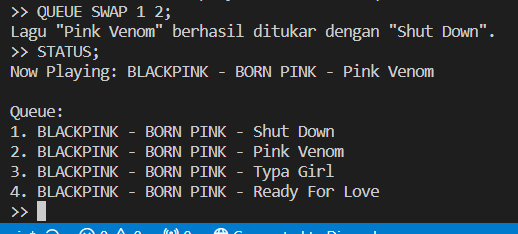
Gambar 8. Play Song Gambar 9. Play Playlist

## Data Test Queue

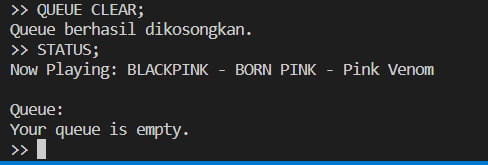
Pada data test queue, terdapat 5 pengetesan yang kami lakukan, yaitu queue song, queue playlist, queue swap, queue remove, dan queue clear. Secara umum fungsi ini mengetes bagaimana ADT queue bekerja, seperti queue swap untuk menukar antrean lagu, queue remove untuk menghapus lagu dari antrean, queue clear untuk menghapus semua antrean lagu. Pada queue song dan queue playlist, kami juga mengetes struktur data dari penyanyi, album, dan lagu, yaitu list statik, map, dan set serta list dinamis dan linked list yang merupakan struktur data daftar playlist dan lagu dalam playlist. Hasil yang seharusnya dihasilkan adalah penambahan lagu atau lagu - lagu dalam playlist ke dalam antrean dengan format rata kiri, menukar urutan antrean sesuai dengan inputan user, menghapus lagu dari antrean sesuai inputan user, dan membersihkan antrean dengan menghapus semua lagu dalam antrean.



Gambar 10. Queue Song Gambar 11. Queue Playlist



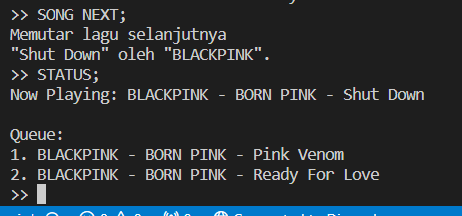
Gambar 12. Queue Swap Gambar 13. Queue Remove



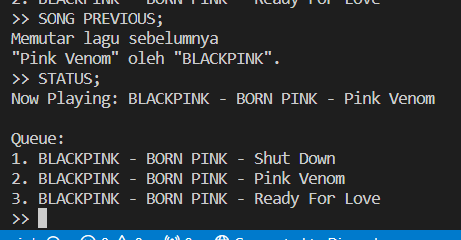
Gambar 14. Queue Clear

## Data Test Song

Pada data test song, terdapat 2 pengetesan, yaitu song next dan song previous. Kedua pengetesan tersebut mengetes fungsi - fungsi yang bekerja di ADT queue, seperti enqueue, dequeue, Fungsi ini juga mengetes fungsi - fungsi yang bekerja di ADT stack, yaitu pop dan push. Hasil yang seharusnya dihasilkan adalah pemutaran lagu berikutnya atau pemutaran lagu sebelumnya dengan current play song menjadi antrean dengan urutan pertama dan riwayat previous song hilang.



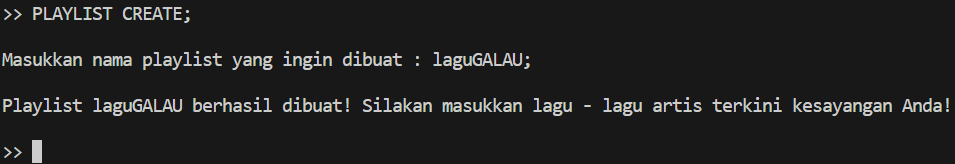
Gambar 15. Song Next



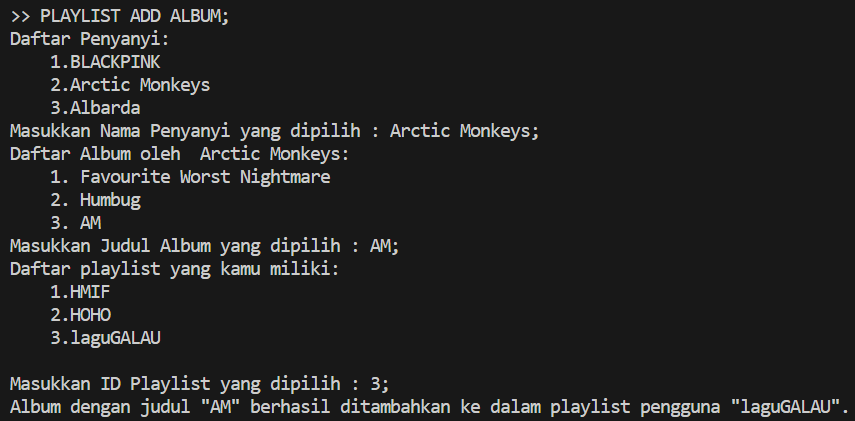
Gambar 16. Song Previous

## Data Test Playlist

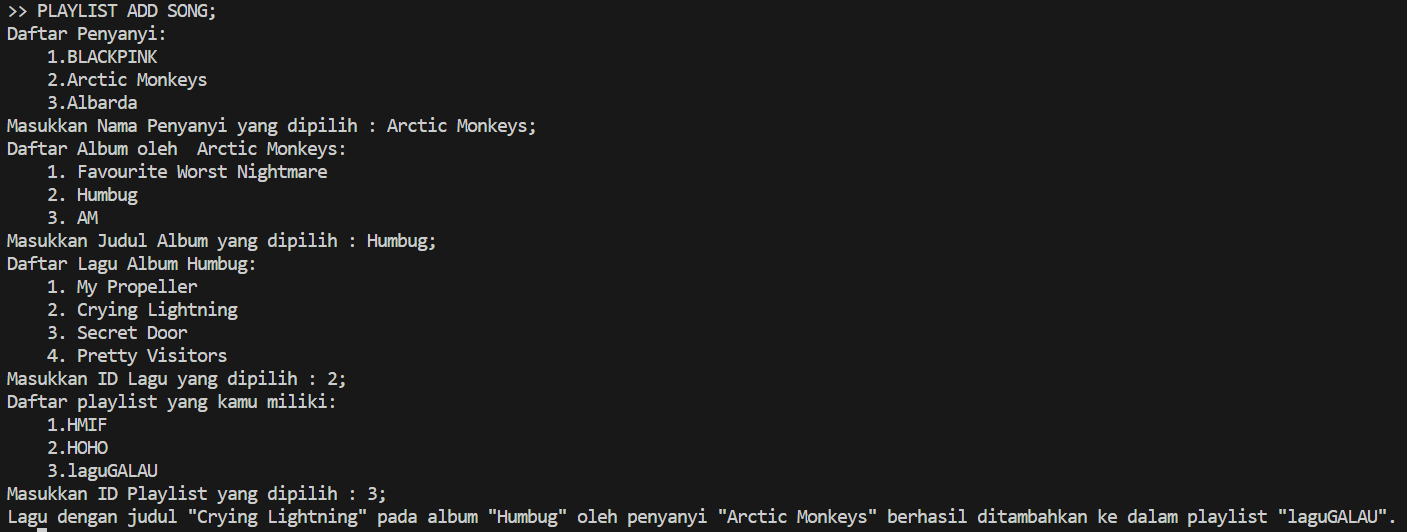
Pada data test playlist, kami mengetes lima command yaitu create, add, swap, remove, dan delete. Command playlist create melakukan pembuatan playlist dengan syarat nama playlist harus lebih dari tiga karakter bukan spasi. Playlist add dibagi dua menjadi playlist add song yang berfungsi untuk menambahkan lagu ke dalam playlist yang dipilih dengan syarat lagu tersebut awalnya bukan merupakan anggota dari playlist yang dituju. Sedangkan playlist add album berfungsi untuk menambahkan lagu-lagu dalam sebuah album ke playlist yang dipilih dengan syarat yang sama seperti sebelumnya. Lalu command playlist swap digunakan untuk menukar dua lagu dalam sebuah playlist berdasarkan indeksnya. Command playlist remove bekerja untuk menghapus sebuah lagu dalam suatu playlist berdasarkan indeks dan playlist delete berfungsi menghapus sebuah playlist dari daftar playlist yang ada berdasarkan indeks playlist.



Gambar 17. Playlist Create



Gambar 18. Playlist Add Album



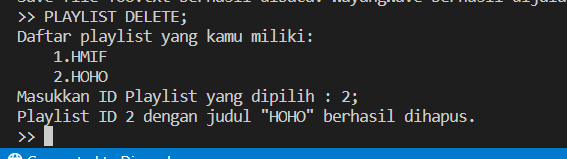
Gambar 19. Playlist Add Song



Gambar 20. Playlist Swap



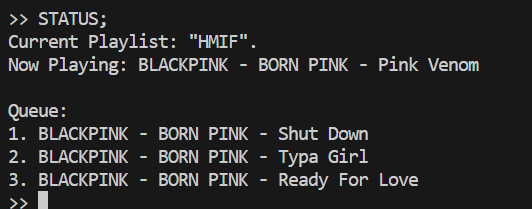
Gambar 21. Playlist Remove



Gambar 22. Playlist Delete

## Data Test Status

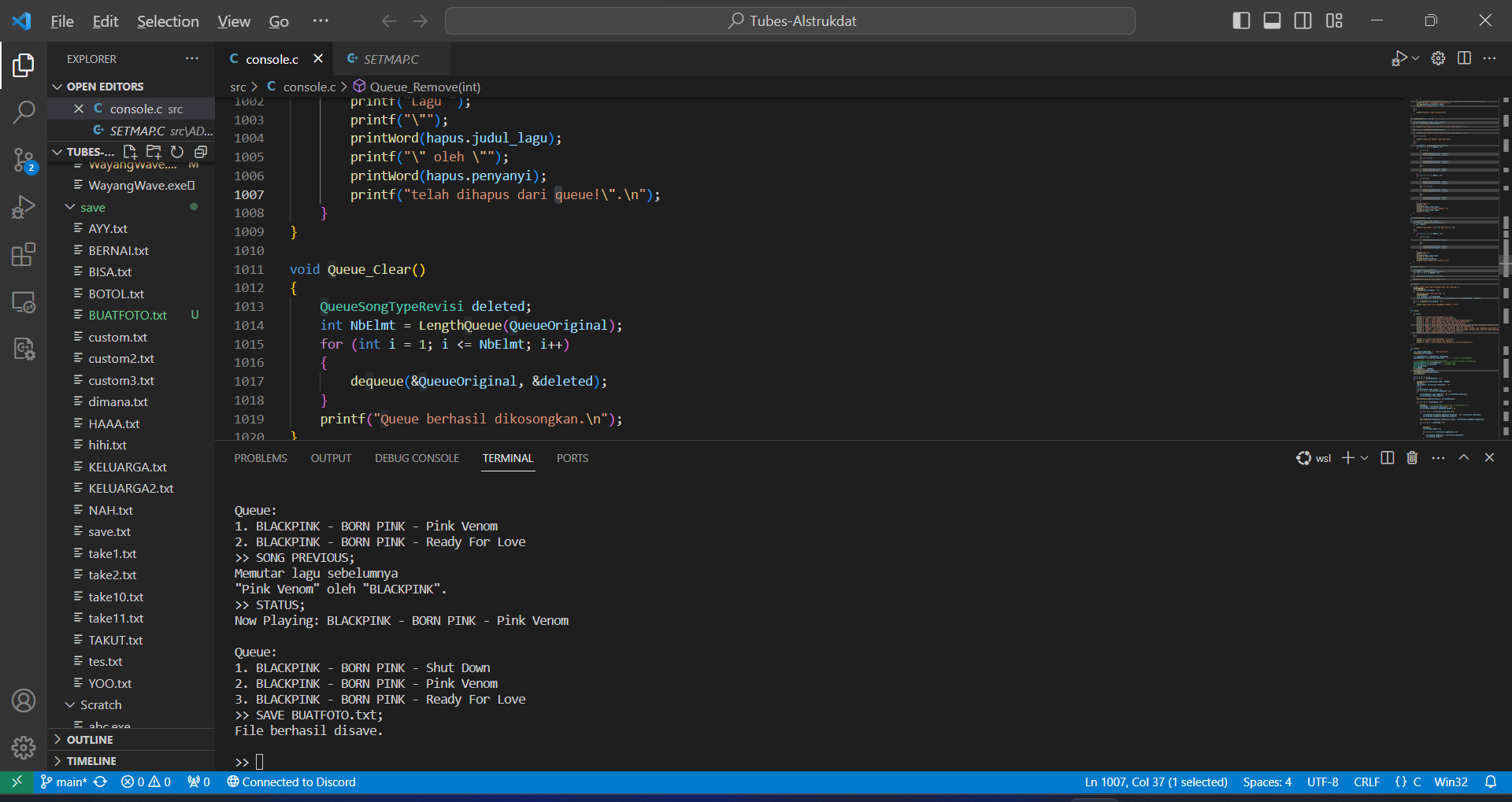
Pada data test status, kami mengetes current playlist yang hanya muncul saat play playlist satu kali, current play song yang ada, dan antrean yang ada. Hasil yang seharusnya dihasilkan adalah penampilan playlist yang diputar jika melakukan play playlist, lagu yang sedang diputar dan antrean lagu yang ada.



Gambar 23. Status

## Data Test Save

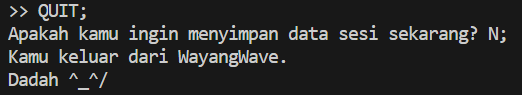
Pada data test save, kami mengetes pembuatan file dengan nama sesuai dengan input user, pembacaan dari program dan menuliskannya ke file tersebut dengan format sesuai spesifikasi. Hasil yang seharusnya dihasilkan adalah terdapat file baru dengan nama sesuai dengan inputan user dan berisikan penyanyi, album, lagu sesuai dengan database, antrean, riwayat lagu yang berasal dari fungsi-fungsi yang sudah dijalankan user, dan semua playlist yang telah dibuat user.



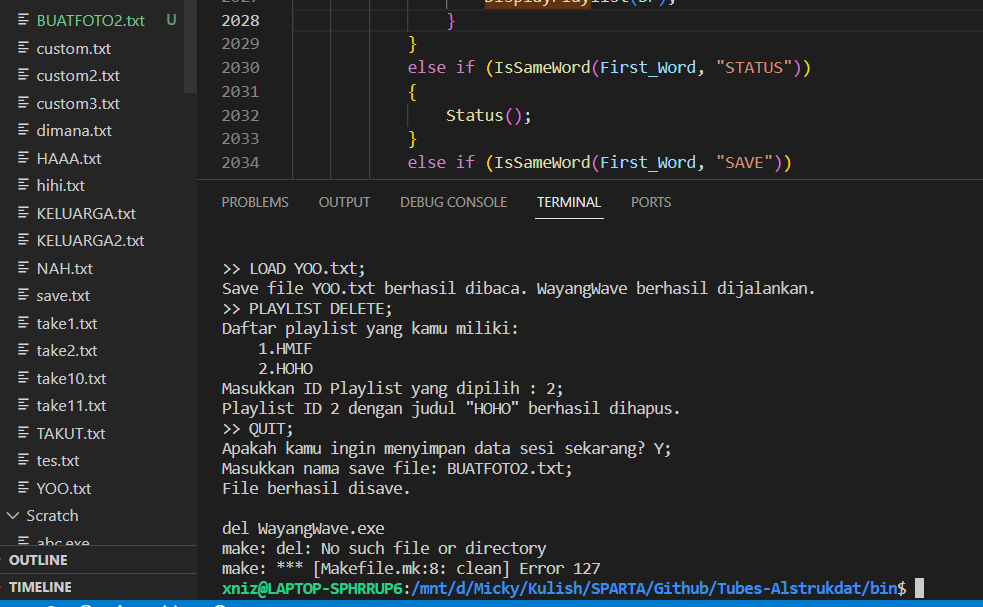
Gambar 24. Save

## Data Test Quit

Pada data test quit, kami mengetes sintaks untuk mengeluarkan user dari aplikasi WayangWave. Hasil yang seharusnya dihasilkan adalah pengeluaran user dari aplikasi WayangWave jika sesuai inputan user.



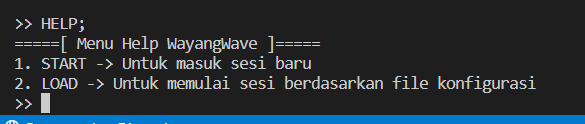
Gambar 25. Quit tanpa save



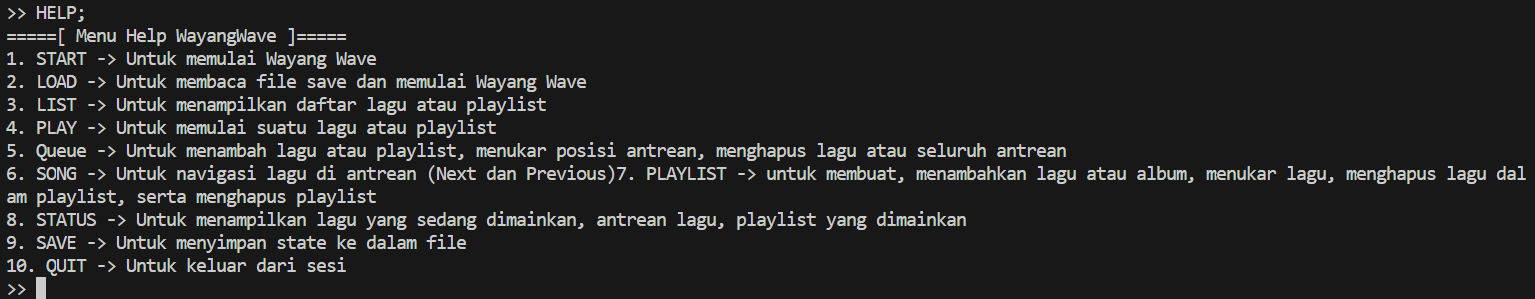
Gambar 26. Quit dengan Save

## Data Test Help

Pada data test help, kami mengetes pencetakan pemberitahuan fitur - fitur yang ada dan kegunaannya sesuai dengan sesi user. Hasil yang seharusnya dihasilkan adalah pencetakan fitur - fitur sesuai sesi user (sebelum masuk sesi dan setelah masuk sesi).



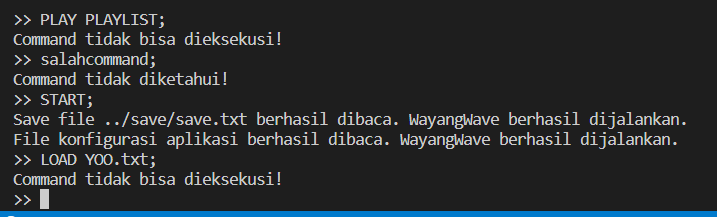
Gambar 27. Help sebelum sesi



Gambar 28. Help setelah sesi

## Data Test <INVALID\_COMMAND>

Pada data test invalid command, kami mengetes pengeluaran teks berisi error atau invalid sesuai kasus. Hasil yang seharusnya dihasilkan adalah pencetakan teks error sesuai kesalahan yang user lakukan (tidak sesuai sesi atau tidak ada dalam fungsi WayangWave)



Gambar 29. Invalid Command

# **Test Script**

Isi dengan skenario test yang dimungkinkan untuk semua fitur yang ada. Bisa dibuat dalam bentuk tabel sebagai berikut:

| **No.** | **Fitur yang Dites** | **Tujuan Testing** | **Langkah-Langkah Testing** | **Input Data Test** | **Hasil yang Diharapkan** | **Hasil yang Keluar** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | START | Untuk mengetahui apakah WayangWave dapat dijalankan atau tidak | Melakukan input command START | Gambar 3 | Program berhasil membaca konfigurasi default (save.txt) | Sesuai yang diharapkan |
| 2 | LOAD | Untuk mengetahui apakah dapat membaca file serta mengalokasikan variabel tersebut dengan benar. | Melakukan input command LOAD <namafile>.txt | Gambar 4 | Program berhasil membaca file konfigurasi yang diinginkan | Sesuai yang diharapkan |
| 3 | List Default | Untuk mengetahui apakah data yang di-load program sesuai dengan file konfigurasi | Melakukan START lalu LIST DEFAULT | Gambar 6 | Program dapat menampilkan isi data sesuai file konfigurasi | Sesuai yang diharapkan |
| 4 | List Playlist | Untuk mengetahui apakah playlist telah disimpan dengan baik | Melakukan START lalu LIST PLAYLIST | Gambar 7 | Program dapat menampilkan playlist yang disimpan dengan baik | Sesuai yang diharapkan |
| 5 | PLAY SONG | Untuk mengetahui apakah dapat mengakses lagu serta memodifikasi queue/riwayat | Melakukan START lalu PLAY SONG | Gambar 8 | Program dapat memainkan lagu dari data serta menempatkan ke queue/riwayat | Sesuai yang diharapkan |
| 6 | PLAY PLAYLIST | Untuk mengetahui apakah lagu yang disimpan dalam playlist dapat diakses serta memodifikasi queue/riwayat | Melakukan START lalu CREATE PLAYLIST , PLAYLIST ADD SONG, kemudian PLAY PLAYLIST | Gambar 9 | Program dapat memainkan lagu dari playlist serta menempatkan ke queue/riwayat | Sesuai yang diharapkan |
| 7 | QUEUE SONG | Untuk menambahkan lagu pada antrean | Melakukan START lalu memilih penyanyi, album dan id lagu | Gambar 10 | Lagu ditambahkan ke antrean | Sesuai yang diharapkan |
| 8 | QUEUE PLAYLIST | Untuk menambahkan playlist pada antrean | Melakukan START lalu memiliki id playlist | Gambar 11 | Program dapat menambahkan isi playlist ke dalam queue | Sesuai yang diharapkan |
| 9 | QUEUE SWAP | Untuk mengetahui apakah lagu dalam antrean dapat ditukar | Melakukan START kemudian QUEUE SONG 2x , kemudian QUEUE SWAP | Gambar 12 | Program dapat menukar lagu sesuai indeks lagu yang dimasukkan | Sesuai yang diharapkan |
| 10 | QUEUE REMOVE | Untuk mengecek apakah dapat menghilangkan sebuah lagu dalam antrean | Melakukan START kemudian QUEUE SONG, kemudian QUEUE REMOVE | Gambar 13 | Lagu dalam suatu antrean hilang | Sesuai yang diharapkan |
| 11 | QUEUE CLEAR | Untuk mengecek apakah dapat menghilangkan semua lagu dalam antrean | Melakukan START kemudian QUEUE SONG | Gambar 14 | Seluruh lagu dalam suatu antrean hilang | Sesuai yang diharapkan |
| 12 | SONG NEXT | Untuk mengetahui apakah dapat memainkan lagu pada queue berikutnya | Melakukan START, kemudian QUEUE SONG, kemudian SONG NEXT | Gambar 15 | Memainkan lagu selanjutnya | Sesuai yang diharapkan |
| 13 | SONG PREVIOUS | Untuk mengetahui apakah dapat memainkan lagu pada riwayat terbaru / mengulang current song | Melakukan START kemudian bentuk play apapun, kemudian SONG PREVIOUS | Gambar 16 | Memainkan lagu sebelumnya/lagu sekarang jika tidak ada riwayat | Sesuai yang diharapkan |
| 14 | PLAYLIST CREATE | Untuk membuat playlist baru | Melakukan START lalu PLAYLIST CREATE, lalu memasukkan nama playlist | Gambar 17 | Playlist bertambah sesuai nama | Sesuai yang diharapkan |
| 15 | PLAYLIST ADD ALBUM | Untuk menambahkan semua lagu dalam album ke playlist yang sudah ada | Melakukan START lalu PLAYLIST CREATE lalu memasukkan nama playlist, lalu PLAYLIST ADD ALBUM, kemudian memasukkan nama penyanyi, dan memasukkan nama album | Gambar 18 | Semua lagu dalam album masuk ke dalam playlist | Sesuai yang diharapkan |
| 16 | PLAYLIST ADD SONG | Untuk mengetahui apakah dapat meng-add 1 lagu pada suatu playlist | Melakukan START lalu PLAYLIST CREATE, kemudian PLAYLIST ADD SONG | Gambar 19 | Sebuah lagu ditambahkan pada suatu playlist | Sesuai yang diharapkan |
| 17 | PLAYLIST SWAP | Untuk menukar lagu yang ada di playlist | Melakukan START lalu PLAYLIST CREATE, kemudian PLAYLIST ADD ALBUM lalu PLAYLIST SWAP 1 1 2. | Gambar 20 | lagu di playlsit 1 dengan id 1 ditukar dengan id 2 | Sesuai yang diharapkan |
| 18 | PLAYLIST REMOVE | Untuk mengetahui apakah dapat menghapus sebuah lagu dalam playlist | Melakukan START lalu PLAYLIST CREATE, kemudian PLAYLIST REMOVE | Gambar 21 | Lagu dalam suatu playlist berhasil dihilangkan | Sesuai yang diharapkan |
| 19 | PLAYLIST DELETE | Untuk menghapus 1 playlist | Melakukan LOAD YOO.txt lalu PLAYLIST DELETE 2 | Gambar 22 | Menghapus playlist dengan indeks inputan (2) | Sesuai yang diharapkan |
| 20 | STATUS | Untuk mengetahui apakah fungsi-fungsi yang dijalankan dapat berjalan sesuai tujuan. | Melakukan LOAD YOO.txt, lalu QUEUE PLAYLIST, lalu STATUS | Gambar 23 | Menampilkan current play song dan semua antrean lagu | Sesuai yang diharapkan |
| 21 | SAVE | Untuk mengetahui apakah dapat menyimpan data dalam program ke dalam suatu file | Melakukan START, menyimpan beberapa lagu dari PLAY PLAYLIST DAN QUEUE SONG, lalu SAVE BUATFOTO.txt | Gambar 24 | Menyimpan file yang dibuat di folder save dan memiliki isi dengan aktivitas user | Sesuai yang diharapkan |
| 23 | QUIT | Untuk keluar dari aplikasi WayangWave | Melakukan LOAD YOO.txt lalu QUIT | Gambar 25  Gambar 26 | Keluar dari wayang wave | Sesuai yang diharapkan |
| 22 | HELP | Untuk mengetahui fungsi apa saja yang tersedia | Melakukan input command HELP | Gambar 27  Gambar 28 | Menampilkan bantuan tentang fungsi - fungsi | Sesuai yang diharapkan |

# **Pembagian Kerja dalam Kelompok**

| No | Nama Anggota | Pembagian Tugas |
| --- | --- | --- |
| 1 | Alessandro Jusack Hasian ( 18222025 ) | Membuat ADT mesin kata/mesin karakter serta fungsi yang berhubungan dengan adt tersebut. Serta membuat seluruh fungsi start, load, save, list default. Juga membuat beberapa fungsi untuk adt stack, queue, set, map dan list statik.  Debugging dan testing. Serta membuat laporan.  Revisi fungsi. |
| 2 | Billy Samuel Setiawan ( 18222039 ) | Membuat ADT list dinamis dan list linier, serta fungsi yang berhubungan dengan adt tersebut. Serta membuat seluruh fungsi yang berhubungan dengan playlist. Membuat laporan. Membuat fungsi dasar untuk list linier/list dinamis. Debugging dan testing.  Revisi fungsi, file header, dan laporan. |
| 3 | Micky Valentino ( 18222093 ) | Membuat ADT queue, stack serta fungsi yang berhubungan dengan adt tersebut. Serta membuat seluruh fungsi yang berhubungan dengan history,queue, play song, help, quit, invalid command.Membuat cara compile program. Debugging dan testing. Serta membuat laporan.  Revisi fungsi. |
| 4 | Sitanggang Jerry J. P. ( 18220082 ) | Membuat adt list statik, beserta fungsi primitive. Juga membuat beberapa bagian laporan. Membuat lampiran 1 dan 2. |
| 5 | Rahmat Pujiatno ( 18219062 ) | - |

# **Lampiran**

## ***Deskripsi Tugas Besar***

**Latar Belakang**

**Spesifikasi Umum**

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

**System Mechanic**

1. **About the System**

WayangWave merupakan sebuah aplikasi yang bisa mensimulasikan service

pemutaran musik. WayangWave ini memiliki 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

1. **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.

1. **Command**

Pemain dapat memasukkan command - command berikut :

1. **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.

1. **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](https://docs.google.com/document/d/1PCSWm3fqjT4dnMEdRDeMqJP6V5_1I62nitk-Z_1_p2g/edit#heading=h.hh57cbvsu4kb).

1. **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.

1. **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.

1. **QUEUE**

QUEUE merupakan command yang digunakan untuk memanipulasi queue lagu.

1. **SONG**

SONG merupakan command yang digunakan untuk navigasi lagu yang ada pada queue lagu saat ini.

1. **PLAYLIST**

Command ini digunakan untuk melakukan basic command untuk playlist

yaitu **CREATE**, **ADD**, **SWAP**, **REMOVE** dan **DELETE**.

1. **STATUS**

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

1. **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.

1. **QUIT**

QUIT merupakan command yang digunakan untuk keluar dari aplikasi WayangWave.

1. **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.

1. **<INVALID COMMAND>**

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

## ***Notulen Rapat***

**Form Asistensi Tugas Besar**

**IF2111/Algoritma dan Struktur Data STI**

**Sem. 1 2023/2024**

No. Kelompok /Kelas : 4 / 01

Nama Kelompok :

Anggota Kelompok (Nama/NIM) : 1. Rahmat Pujiatno (18219062)

2. Sitanggang Jerry J.P. (18220082)

3. Alessandro Jusack Hasian (18222025)

4. Billy Samuel Setiawan (18222039)

5. Micky Valentino (18222093)

Asisten Pembimbing : Riandy Hasan

Asistensi I

| **Tanggal : 29 Oktober 2023** | **Catatan Asistensi:**  **List Pertanyaan :**  ADTnya boleh dari praktikum?  Kalo play song riwayatnya dihapus?  kalo play playlist riwayatnya jadi kebalikan queue?  kalo Song next, queue kosong, lagu yang sedang diputar dimasukkin riwayat ga?  kalo song previous unik(?)  playlist boleh double ga  penyanyi dapet datanya darimana  pembagian tugas lebih baik gimana?  **Jawaban :**   1. Adt yang wajib bisa dari prak aja, cuman bisa disesuaikan sesuai kebutuhan ( Mau ditambahin juga boleh). 2. Setelah play song, riwayat di stack & queue dihapus 3. Konfigurasi dibuat sendiri ( yang penting ikutin format yang udah ada di spek. 4. History bisa aja duplikat, urutan harus jelas dan sesuai. **Stack Dinamis**. Max 100, pas demo ga bakal dicek diatas 100. 5. Fungsi repeat lagu tetap bakal kecatat di history 6. **Song Previous** : kalo lagu pertama di previous, bakal ngestack lagu itu terus. 7. Playlist ga bakal duplikat. ( **belum confirm : sementara boleh sama sesuai spek**) 8. Play playlist bakal ditumpuk 9. Semua lagu yg dari play playlist otomatis masuk ke history (**tapi mending nanti sewaktu diplay masuk ke historynya** karena kalau langsung semua sebelum diplay rawan duplikat) 10. Rekomendasi pembagian tugas : Bagi per adt, buat driver, bagi per command.(Pengalaman asisten).   Ntar command dibagi berdasarkan tingkat kesulitan.   1. Masukin langsung penyanyi - lagu - album dijadiin 1 ADT baru kalo mau biar ga usah search   **Referensi repo :**  <https://github.com/riandyhasan/Tubes-Alstrukdat> |
| --- | --- |
| **Tempat : Google Meets ( Online)** |
| **Kehadiran Anggota Kelompok:**  1  Rahmat Pujiatno    2  Sitanggang Jerry Josia Partogi    3  Alessandro Jusack Hasian    4  Billy Samuel Setiawan    5  Micky Valentino |
|  | **Tanda Tangan Asisten:** |

Asistensi II

| **Tanggal : 15 November 2023** | **Catatan Asistensi:**  Driver udah dibikin?  mesinkata udah ( bla bla bla). Intinya belum yakin drivernya udah bener apa engga.  Drivernya udah ok sih ( kata asisten)  **Playlist**  bagian playlist (linked list). ADT udah, dalam penggunaannya ditambahin nama doang. supaya nunjuk ke lagu. Playlist Add lagi ongoing.  Kendala : aman aman aja  Tinggal yang album.  **Objective start -> bikin playlist, queue sama record -> save -> load -> ubah2 urutan lagu**  **QnA**  Q : Kendala : di file itu mark nya apa? Dikasih titik apa gimana? Mark untuk akhir file?  A : \0. dikasih end line tiap akhir  Q : Secara overall, ada kendala?  A : Masih ada problem dalam compile bareng-bareng. Gabungin semuanya ke satu adt.h  Q : Cara kerja bat  A : main.c inc console.c inc console.h inc adt.h    Q : multiple declared main  A : pilih yang dipake doang. Hati hati ada yang ga diinginin malah ikut ke compile jadinya ntar error. Bagusnya :  **Reference**  <https://github.com/riandyhasan/Tubes-Alstrukdat/blob/main/src/adt.h>  **Note** 1 ngebaca line  1 ngebaca kalimat  \n |
| --- | --- |
| **Tempat : Google Meets ( Online)** |
| **Kehadiran Anggota Kelompok:**  1  Sitanggang Jerry Josia Partogi    2  Alessandro Jusack Hasian    3  Billy Samuel Setiawan    4  Micky Valentino |
|  | **Tanda Tangan Asisten:** |

## ***Log Activity Anggota Kelompok***

| **Tanggal** | **Aktivitas** |
| --- | --- |
| 25 Oktober 2023 | * Pembuatan grup line |
| 26 Oktober 2023 | * Pembuatan Repository Github |
| 27 Oktober 2023 | * Meet Perdana |
| 29 Oktober 2023 | * Asistensi ke - 1 |
|  |  |
|  |  |
| 15 November 2023 | * Asistensi ke - 2 |
| 24 November 2023 | * Debugging * Testing * Finalisasi Laporan * Formatting dan Pengumpulan Tugas Besar |