

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

# WayangWave88Gacor

Dipersiapkan oleh:

Kelompok 01 (K03)

Jeremy Deandito / 18222112

Anindita Widya Santoso / 18222128


Bryan P. Hutagalung / 18222130

Chairul Nur Wahid / 18222132

Salsabila Azzahra / 18222139

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	<b>Sekolah Teknik Elektro dan Informatika ITB</b>	<b>Nomor Dokumen</b>		<b>Halaman</b>
		<i>IF2111-TB-03-01</i>		<i>38</i>
		<i>Revisi</i>	<i>0</i>	<i>25 Oktober 2023</i>

# Daftar Isi

1	Ringkasan	4
2	Penjelasan Tambahan Spesifikasi Tugas	4
	2.1 Enhance	4
	2.2 Multi User	4
3	Struktur Data (ADT)	5
	3.1 ADT List Statis	5
	3.2 ADT List Dinamis	5
	3.3 ADT List Linier	5
	3.4 ADT Mesin Karakter	6
	3.5 ADT Mesin Kata	6
	3.6 ADT Queue	7
	3.7 ADT Stack	7
	3.8 ADT Set	7
	3.9 ADT Map	7
	3.10 ADT User	8
	3.11 ADT Pcolor	8
4	Program Utama	8
5	Algoritma-Algoritma Menarik	9
	5.1 RNG	9
	5.2 INTERFACE	9
	5.3 LIST SONG	10
6	Data Test	10
	6.1 START	10
	6.2 LOGIN	11
	6.3 LOAD	12
	6.4 SIGN UP	13
	6.5 LIST	13
	6.5.1 LIST DEFAULT	13
	6.5.2 LIST PLAYLIST	15
	6.6 PLAY	15
	6.6.1 PLAY SONG	15
	6.6.2 PLAY PLAYLIST	16
	6.7 QUEUE	17
	6.7.1 QUEUE SONG	17
	6.7.2 QUEUE PLAYLIST	17

6.7.3 QUEUE SWAP <x> <y>	18
6.7.4 QUEUE REMOVE <n>	18
6.7.5 QUEUE CLEAR	19
6.8 SONG	20
6.8.1 SONG NEXT	20
6.8.2 SONG PREVIOUS	20
6.9 PLAYLIST	21
6.9.1 PLAYLIST CREATE	21
6.9.2 PLAYLIST ADD	21
6.9.2.1 PLAYLIST ADD SONG	21
6.9.2.2 PLAYLIST ADD ALBUM	22
6.9.3 PLAYLIST SWAP ,<id> <x> <y>	22
6.9.4 PLAYLIST REMOVE <id> <n>	23
6.9.5 PLAYLIST DELETE	23
6.10 STATUS	23
6.11 SAVE <filename>	24
6.12 QUIT	25
6.13 HELP	25
6.14 ENHANCE	26
6.15 LOGOUT	27
6.16 <INVALID COMMAND>	27
7 Test Script	28
8 Pembagian Kerja dalam Kelompok	31
9 Lampiran	32
9.1 Deskripsi Tugas Besar	32
9.2 Notulen Rapat	32
9.3 Log Activity Anggota Kelompok	32
9.4 Form Asistensi Tugas Besar	34

# 1 Ringkasan

Bondowoso menyukai Roro Jonggrang dan ingin menunjukkan Roro Jonggrang selera musik Bondowoso agar Roro menyukainya. Bondowoso tahu bahwa Roro juga suka mendengarkan musik melalui *walkman* yang Roro punya. Oleh karena itu, Bondowoso ingin membuat aplikasi pengganti perangkat lunak *walkman* Roro yang diharapkan dapat memainkan dan juga menunjukan lagu-lagu di *playlist* yang telah Bondowoso buat. Untuk mewujudkan keinginannya itu, Bondowoso meminta 5 teman ghaibnya yang ia percaya dapat melakukannya.

*WayangWave88Gacor* adalah aplikasi yang dapat melakukan pemutaran musik seperti *music streaming service* pada umumnya yang pembuatannya berbasis CLI (*Command-Line Interface*). Hal-hal yang dapat dilakukan oleh aplikasi *WayangWave88Gacor* adalah memutar lagu, menampilkan daftar lagu, membuat dan menghapus *playlist*, mengatur urutan dimainkannya lagu, dan menampilkan status dari aplikasi.

Laporan ini berisi pembahasan mengenai program-program yang telah dibuat berdasarkan spesifikasi penugasan yang telah diberikan. Pada bagian awal berisi ringkasan dari apa yang telah dikerjakan untuk membuat aplikasi. Bagian selanjutnya terdapat penjelasan tentang ADT yang kelompok kami pakai untuk membantu membuat aplikasi. Bagian selanjutnya berisi tentang penjelasan fungsi-fungsi utama. Bagian kelima berisi mengenai algoritma menarik yang terdapat dalam program yang telah kami buat.

Pemutar musik ini dibuat dengan menggunakan bahasa pemrograman C dengan memanfaatkan ADT yang telah dipelajari di mata kuliah IF2111 dengan beberapa tambahan ADT yang kami buat sendiri. ADT yang digunakan adalah ADT List Linier, ADT List Dinamis, ADT List Statis, ADT Mesin Karakter, ADT Mesin Kata, ADT Queue, ADT Stack, ADT Set, ADT Map, ADT User, dan ADT Pcolor.

## 2 Penjelasan Tambahan Spesifikasi Tugas

### 2.1 Enhance

Fitur *Enhance* dalam *WayangWave88Gacor* memberikan *user* kesempatan untuk memperkaya pengalaman mendengarkan musik mereka dengan menambahkan lagu-lagu secara acak ke dalam *playlist* yang telah mereka pilih sebelumnya.

### 2.2 Multi User

Ketika pengguna ingin menggunakan aplikasi, mereka perlu melakukan *login* ke akun yang sudah terdaftar atau melakukan *sign up* jika belum memiliki akun. Oleh karena itu, aplikasi *WayangWave88Gacor* dapat digunakan oleh lebih dari satu akun. Setiap akun memiliki *queue*, *history*, dan *playlist* yang berbeda. *WayangWave88Gacor* memungkinkan pengguna untuk beralih antar akun dengan melakukan *logout* saat ingin mengganti akun, dan kemudian melakukan *login* kembali ke akun yang sudah terdaftar.

## 3 Struktur Data (ADT)

### 3.1 ADT List Statis

Modul ADT List Statis digunakan untuk menyimpan list artis dan lagu yang sedang diputar. ADT ini berisi fungsi untuk membuat sebuah list yang panjang listnya sudah ditentukan dari awal sehingga modul ini dapat dimanfaatkan untuk membatasi data yang tidak mungkin bertambah lagi seperti data artis dan lagu yang sedang diputar.

ADT ini diimplementasikan dengan variasi list implisit dan rata kiri. Operasi-operasi primitif yang tersedia pada ADT List Statis adalah antara lain:

1. CreateEmptyStatic yang berfungsi untuk membuat List Statis kosong
2. IsListEmpty yang berfungsi untuk mengecek apakah List Statis kosong
3. GetList yang akan mengembalikan data pada list indeks ke-i
4. LengthList yang berfungsi untuk mengembalikan jumlah data pada list
5. InsertLastStatic yang berfungsi untuk memasukkan data ke bagian paling akhir List Statis

### 3.2 ADT List Dinamis

Modul ADT List Dinamis digunakan untuk user, file, dan playlist. ADT ini berisi fungsi untuk membuat sebuah list yang panjang listnya berubah-ubah sesuai dengan masukkan yang diinginkan. Modul ini dapat digunakan bagi data yang fleksibel, seperti user, file, dan playlist karena ketiga data memiliki masukkan yang tidak dapat ditentukan dan berubah-ubah seiring berjalannya program.

Operasi-operasi primitif yang tersedia pada ADT List Dinamis dapat dikatakan memiliki fungsi yang kurang lebih sama dengan yang ada pada List Statis, hanya saja digunakan oleh List Dinamis. adalah antara lain: CreateEmptyDynamic, DealokasiDynamic, IsListEmptyDynamic, GetDynamic, IsListFullDynamic, LengthListDynamic, InsertLastDynamic.

Beberapa fungsi yang berbeda dengan yang ada di List Statis adalah:

1. DealokasiDynamic yang berfungsi untuk melakukan pengembalian memori ke sistem
2. IsListFullDynamic yang berfungsi untuk mengecek apakah List Dinamis penuh

### 3.3 ADT List Linier

Modul ADT List Linier merupakan tipe data bentukan list berkait yang hanya memiliki *pointer* ke alamat pertama dan elemen berikutnya. ADT ini memiliki tipe data list dengan *address* sebagai penunjuk ke satu *node* bertipe ElmtList yang terdiri atas *infotype* dan *address*.

Operasi-operasi primitif yang tersedia pada ADT List Linier memiliki fungsi yang kurang lebih sama dengan List Statik dan List Dinamik, tetapi berlaku pada List Linier. Operasi yang ada adalah antara lain: IsEmptyLinier, CreateEmptyLinier, Alokasi, Dealokasi, SearchAddress, SearchLinier, InsVFirst, InsVLast, DelVFirst, DelVLast, InsertFirst, InsertAfter, InsertLast, DelFirst, DelP, DelLast, DelAfter, NbElmt, InversLinierList.

Beberapa fungsi yang menarik adalah:

1. Alokasi yang berfungsi untuk mengirimkan *address* hasil alokasi
2. Ins(V)/Del(V) berfungsi untuk menyisipkan atau menghilangkan *value* yang ada
3. InversLinierList berfungsi untuk membalik elemen tanpa alokasi/dealokasi

### 3.4 ADT Mesin Karakter

Modul ADT Mesin Karakter digunakan untuk menerima ataupun membaca sebuah masukan berupa string yang nantinya akan dibaca huruf per huruf. Tidak hanya itu, pembacaan ini menggunakan sebuah pita yang nantinya dapat mendeteksi apakah pita membaca sebuah pengecualian karakter, seperti Mark, Blank, ataupun Newline dan menghasilkan EOP.

Operasi-operasi primitif ADT Mesin Karakter yang digunakan dalam pembuatan program adalah antara lain: StartMark, StartBlank, StartNewLine, AdvMark, AdvBlank, AdvNewLine, IsEOPMark, IsEOPBlank, IsEOPNewLine.

Operasi yang ada pada Mesin Karakter ini memiliki banyak redundansi karena terdapat banyak *case* yang bisa terjadi, yaitu Mark, Blank, dan NewLine. Fungsi dari operasi-operasi ini sendiri adalah:

1. Start berfungsi untuk menyiapkan mesin dan pitanya
2. Adv berfungsi untuk memajukan pita satu karakter
3. IsEOP berfungsi untuk mengecek apakah karakter yang sedang dibaca merupakan tanda yang menandakan End Of Process dari suatu operasi

### 3.5 ADT Mesin Kata

Modul ADT Mesin Kata digunakan untuk untuk menduplikasi ataupun menyimpan huruf-huruf dari bacaan mesin karakter menjadi sebuah struktur baru bernama Word. Selain itu, mesin kata dapat menghiraukan beberapa pengecualian ketika pembacaan masukan/file, seperti Mark, Blank, dan juga Newline. Mesin kata juga berguna untuk mengubah dari Word ke string atau integer, ataupun sebaliknya, serta membandingkan, menggabungkan, dan memisahkan antara Word maupun string.

Operasi-operasi primitif ADT Mesin Kata yang digunakan dalam pembuatan program adalah antara lain: IgnoreMarks, IgnoreBlanks, IgnoreNewlines, StartWordMark, StartWordNewline, AdvWordMark, AdvWordBlank, AdvWordNewline, CopyWordMark, CopyWordBlank, CopyWordNewline, DisplayWord, CompareWord, CompareStringWord, WordToInt, StringToWord, SplitWordMark, SplitWordBlank, SplitWordLeftMark, SplitWordLeftBlank, ConcatWord, ReadNameLine, ReadCountWord, ReadCountLine.

Operasi-operasi yang ada pun kurang lebih sama dengan Mesin Karakter, namun jika Mesin Karakter memiliki parameter karakter, di Mesin Kata yang dicek adalah Kata. Fungsi-fungsi yang baru adalah:

1. Ignore yang berfungsi untuk mengabaikan satu atau beberapa parameter (Mark, Blank, dan NewLine) sesuai dengan nama fungsinya
2. CopyWord yang berfungsi untuk mengakuisisi kata dan menyimpannya dalam currentWord
3. DisplayWord yang berfungsi untuk menunjukkan/melakukan “print string” terhadap kata yang sedang ada pada currentWord
4. Compare yang berfungsi untuk membandingkan kata satu dengan kata lainnya
5. Split yang berfungsi untuk memisahkan masukkan menjadi beberapa kata berdasarkan spasi ataupun mark
6. Concat yang berfungsi untuk menggabungkan kata satu dengan kata lainnya
7. Read yang berfungsi untuk membaca kata ataupun jumlah karakter suatu kata

### 3.6 ADT Queue

Modul ADT Queue digunakan untuk merepresentasikan urutan lagu yang sedang dimainkan dan akan dimainkan setelah lagu urutan pertama. ADT Queue cocok untuk representasi urutan lagu dikarenakan modul Queue memanfaatkan prinsip FIFO, yakni *First In, First Out*. Operasi-operasi primitif ADT Queue yang digunakan dalam pembuatan program adalah antara lain:

1. CreateEmptyQueue yang berfungsi untuk membuat *queue* baru yang kosong
2. IsEmptyQueue yang berfungsi untuk mengecek apakah *queue* kosong
3. IsFullQueue yang berfungsi untuk mengecek apakah *queue* penuh
4. IsMemberQueue yang berfungsi untuk mengecek apakah suatu indeks merupakan bagian dari *queue*
5. LengthQueue yang berfungsi untuk mengecek banyaknya elemen *queue*
6. Enqueue yang berfungsi untuk menambahkan elemen baru ke *tail queue*
7. Dequeue yang berfungsi untuk menghapuskan elemen *head queue*

### 3.7 ADT Stack

Modul ADT Stack digunakan untuk menyimpan *history* yang dimiliki oleh *user*. ADT ini dibuat dengan sistem LIFO (Last In First Out), sehingga elemen yang ada pada *top of stack* merupakan elemen terbaru yang dimasukkan dan penyisipan maupun penghapusan akan selalu dilakukan pada elemen puncak.

Operasi-operasi primitif ADT Stack yang digunakan dalam pembuatan program adalah antara lain: CreateEmptyStack, IsEmptyStack, IsFullStack, Push, Pop. Operasi yang ada memiliki kemiripan dengan operasi ADT Queue namun terjadi pada Stack. Fungsi-fungsi yang baru dan berbeda adalah:

1. Push yang berfungsi untuk menambahkan elemen ke TOP Stack
2. Pop yang berfungsi untuk menghapuskan elemen yang ada pada TOP Stack

### 3.8 ADT Set

Modul ADT Set digunakan untuk menyimpan nama-nama album yang terdapat pada setiap artis tanpa terjadi pengulangan.

Operasi-operasi primitif ADT Set yang digunakan dalam pembuatan program adalah antara lain: CreateEmptySet, IsEmptySet, IsFullSet, InsertLastSet, IsMemberSet. ADT Set memiliki operasi primitif yang mirip dengan ADT lainnya.

### 3.9 ADT Map

Modul ADT Map digunakan untuk *album\_artist* dan *song\_album* pada saat membaca konfigurasi. ADT ini berisi *Key* dan *Value*. Pada Map *album\_artist*, album sebagai *key* dan *artist* sebagai *value*. Pada Map *song\_album*, *song* sebagai *key* dan album sebagai *value*.

Operasi-operasi primitif ADT Map yang digunakan antara lain: CreateEmptyMap, IsEmptyMap, IsFullMap, ValueMap, InsertLastMap, IsMemberMap. ADT ini memiliki fungsi baru yaitu ValueMap yang berfungsi untuk mengembalikan nilai *value key*.

### 3.10 ADT User

Modul ADT User digunakan untuk menyimpan nama lagu-lagu dalam *queue* dan *history*, serta nama-nama *playlist* beserta nama lagu-lagunya. ADT ini berisi struktur berupa ADT Queue dengan nama variabel Queue, ADT Stack dengan nama variabel History, ADT List Dinamis dengan nama variabel Playlist, dan ADT List Linier dengan nama variabel PlaylistSong.Song

Operasi-operasi primitif ADT User yang digunakan dalam pembuatan program adalah antara lain:

1. CreateEmptyUser yang berfungsi untuk membuat suatu struktur *queue*, *history*, *playlist*, dan lagu *playlist* untuk banyak *user*.
2. InsertLastUser yang berfungsi untuk menambahkan jumlah elemen user yang memiliki struktur *queue*, *history*, *playlist*, dan lagu *playlist*.

### 3.11 ADT Pcolor

Modul ADT PColor digunakan untuk mengatur tampilan warna pada *output* teks dalam program. ADT ini menyediakan beberapa definisi warna dan fungsi-fungsi untuk mencetak karakter dengan warna-warna tertentu. Terdapat tiga fungsi yang dapat digunakan untuk mencetak karakter dengan warna tertentu ke terminal.

1. void print\_red(char c) yang berfungsi untuk mencetak karakter c dengan warna merah.
2. void print\_green(char c) yang berfungsi untuk mencetak karakter c dengan warna hijau.
3. void print\_blue(char c) yang berfungsi untuk mencetak karakter c dengan warna biru.

## 4 Program Utama

Program utama atau *main* merupakan suatu file yang berfungsi sebagai penyatu dari seluruh ADT yang telah dibuat, yaitu LIST STATIS, LIST DINAMIS, MESIN KARAKTER, MESIN KATA, QUEUE, STACK, SET, MAP, LIST LINIER, USER dan PCOLOR. Tidak hanya itu, program utama juga berfungsi sebagai driver untuk seluruh fungsi/fitur yang telah dibuat pula, yaitu START, LOAD, LIST, PLAY, QUEUE, SONG, PLAYLIST, STATUS, SAVE, QUIT, HELP, ENHANCE, MULTI USER serta INTERFACE. Ketika *user* salah memasukkan *command* dari fungsi/fitur, program utama juga akan mengeluarkan *invalid command*. Keberjalanan program utama menggunakan parameter tiga buah boolean, yaitu run, menu, dan sesi, serta menggunakan konsep banyak kemungkinan *if else* untuk setiap *command* fungsi/fiturnya.

Pada awal program utama dijalankan *run* akan bernilai *true*. Kemudian, jika *user* memasukkan *command* fungsi START atau LOAD, maka *menu* akan bernilai *true*. Lalu, jika *user* memasukkan *command* fungsi LOGIN, maka sesi akan bernilai *true* dan setelah itu seluruh fungsi lain, selain START, LOAD, LOGIN, SIGN UP bisa dijalankan. Jika QUIT dijalankan, maka seluruh boolean *run*, *menu*, dan sesi akan langsung bernilai *false* dan keluar dari program.



## 5 Algoritma-Algoritma Menarik

### 5.1 RNG

Fungsi Random Number Generator atau RNG merupakan algoritma untuk mencari angka *random* di antara 0 sampai suatu batasan. Batasan ini akan menjadi parameter dari fungsi rng yang kemudian akan menghasilkan angka *random*. Fungsi ini menggunakan *seed* dari penghitungan *random time* dan juga *math*. Setelah itu, fungsi RNG ini akan diimplementasikan dalam *command* fitur bonus ENHANCE yang akan menghasilkan angka *random* untuk jumlah lagu yang akan ditambahkan ke dalam *playlist* dan juga akan menghasilkan angka *random* untuk pemilihan lagu yang akan ditambahkan ke dalam *playlist*.

```
unsigned int rng(unsigned int max)
{
    time_t t;
    srand((unsigned int)time(&t));
    static unsigned int seed = 0;
    seed = (seed << 3) + (unsigned int)time(NULL) + rand() + (unsigned int)pow(seed, 2) + (unsigned int)sqrt(seed);

    return (seed % (max)) + 1;
}
```

Gambar 5.1 Tampilan fungsi Random Number Generator

### 5.2 INTERFACE

Pada algoritma *interface*, terdapat beberapa fungsi yang digunakan. Fungsi utama yang digunakan adalah DelayLoad dan Image. Fungsi DelayLoad berfungsi untuk menampilkan gambar atau *interface* dengan ada perbedaan waktu / penundaan di setiap fasenya. Kemudian fungsi Image berfungsi untuk membaca file txt yang berisi suatu ASCII yang nantinya akan ditampilkan dengan warna kuning pada terminal.

```
// ===== DELAY LOAD =====

void DelayLoad(int seconds)
{
    int milli_seconds = 500 * seconds;

    clock_t start_time = clock();

    while (clock() < start_time + milli_seconds);
}

// ===== IMAGE =====

void Image(FILE *ascii)
{
    char read_string[MAX_LEN];
    while(fgets(read_string, sizeof(read_string), ascii) != NULL)
        printf("%s", YELLOW, read_string);
    printf("\n");
}
```

Gambar 5.2.1 Tampilan algoritma pada fungsi INTERFACE

### 5.3 LIST SONG

```
printf("Ingin melihat lagu yang ada? (Y/N) : ");
StartWordMark();
if (CompareStringWord(currentWord, "Y"))
{
    printf("\nMasukkan ID Playlist yang dipilih : ");
    StartWordMark();
    int ID_Playlist = WordToInt(currentWord);

    if (ID_Playlist > 0 && ID_Playlist <= LengthDynamic(multi.Elements[idx_user].Playlist))
    {
        if (NbElmt(multi.Elements[idx_user].PlaylistSong[ID_Playlist-1].Song) > 0)
        {
            Word Dash = StringToWord(" - ");
            address currentSong = First((multi).Elements[idx_user].PlaylistSong[ID_Playlist-1].Song);
            int index = 0;
            for (int i = 0; i < NbElmt(multi.Elements[idx_user].PlaylistSong[ID_Playlist-1].Song); i++)
            {
                Word InfoSong = Info(currentSong);
                Word NamaArtis = SplitWordLeftMark(InfoSong);
                Word Pilihan = ConcatWord(NamaArtis, Dash);
                Word NamaAlbum = SplitWordLeftMark(SplitWordMark(InfoSong));
                Pilihan = ConcatWord(Pilihan, NamaAlbum);
                Pilihan = ConcatWord(Pilihan, Dash);
                Word NamaLagu = SplitWordMark(SplitWordMark(InfoSong));
                Pilihan = ConcatWord(Pilihan, NamaLagu);
                printf("\td. ", index+1);
                DisplayWord(Pilihan);
                index++;
                printf("\n");
                currentSong = Next(currentSong);
            }
        }
        else
        {
            printf("%s\nTidak ada lagu di playlist ini.\n", RED);
        }
    }
}
```

Gambar 5.3 Tampilan algoritma LIST SONG

Pada fungsi *LIST PLAYLIST*, kami menambahkan sebuah algoritma berupa algoritma *list song*. Algoritma ini berfungsi untuk menampilkan lagu-lagu yang terdapat pada sebuah playlist yang telah dibuat oleh user. Setelah fungsi *LIST PLAYLIST* menampilkan nama-nama playlist yang dimiliki oleh user secara terurut menaik, user akan diminta untuk memberikan masukkan. Apabila masukkan berupa “Y”, maka user akan diminta untuk memberikan masukkan ID playlist yang ingin ditampilkan isinya dan isi lagu akan ditampilkan sesuai dengan masukkan ID. Apabila playlist kosong, maka akan menampilkan “Tidak ada lagu di playlist ini. Apabila masukkan berupa “N” maka program tidak akan meminta masukkan ID playlist serta tidak menampilkan isi lagu dari playlist.

## 6 Data Test

### 6.1 START

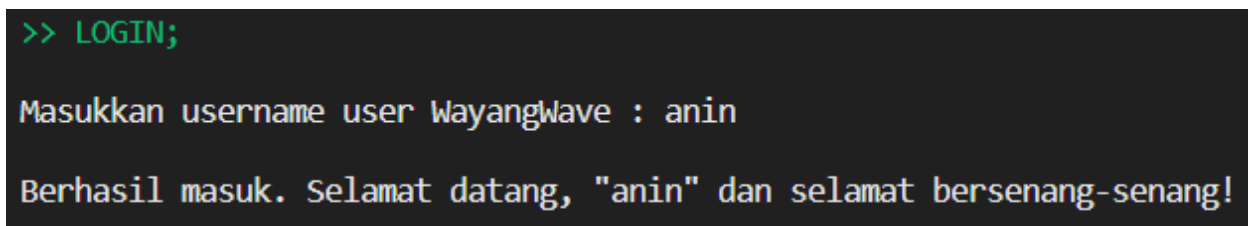
*Command* START digunakan untuk membaca konfigurasi dan memulai program *WayangWave88Gacor*.



Gambar 6.1.1 Tampilan awal ketika *user* memanggil *command* START

### 6.2 LOGIN

*Command* LOGIN digunakan untuk masuk ke dalam aplikasi. Jika *user* berusaha untuk menjalankan *command* aplikasi tanpa melakukan LOGIN, *command* *user* tidak akan terdeteksi.



Gambar 6.2.1 Tampilan saat *command* LOGIN dipanggil dengan *username* yang sudah terdaftar

```
>> LOGIN;

Masukkan username user WayangWave : jeremynyebelin

Tidak ada user dengan username "jeremynyebelin"

Berikut daftar username yang terdaftar :
1. bryan
2. anin
3. jeremy
4. irul
5. icha

Masukkan username user WayangWave : |
```

Gambar 6.2.2 Tampilan saat *command* LOGIN dipanggil, tetapi *username* yang dimasukkan belum terdaftar

### 6.3 LOAD

*Command* LOAD digunakan untuk membaca konfigurasi file txt yang sudah ada. Saat *command* START sudah dipanggil terlebih dahulu, *command* LOAD tidak akan bisa diakses

```
=====
LOADINGSAVE DATA

#####
##:#####
##:#####
##:#####
##:#####
##:#####
#####

HAVE FUN

=====

Save file berhasil dibaca. WayangWave88Gacor berhasil dijalankan.
=====
```

Gambar 6.3.1 Tampilan saat *command* LOAD dipanggil

```
>> LOAD hehe.txt;

Command tidak bisa dieksekusi!
```

Gambar 6.3.2 Tampilan saat *command* LOAD dipanggil dan file yang dipanggil tidak ada

## 6.4 SIGN UP

*Command* SIGN UP digunakan untuk mendaftarkan *username* yang belum ada pada file konfigurasi. Jika *username* yang didaftarkan sudah ada pada sistem, sistem akan menolak pendaftaran *username* tersebut.

```
>> SIGN UP;

Masukkan username user baru WayangWave : anin

Username "anin" sudah didaftarkan, silakan gunakan username lain!
```

Gambar 6.4.1 Tampilan saat *command* SIGN UP dipanggil, tetapi *username* yang didaftarkan merupakan *username* yang sudah ada

```
>> SIGN UP;

Masukkan username user baru WayangWave : aninbaik

Username "aninbaik" berhasil didaftarkan, silakan login kembali!
```

Gambar 6.4.2 Tampilan saat *command* SIGN UP dipanggil dengan *username* yang belum didaftarkan sebelumnya

## 6.5 LIST

*Command* LIST digunakan untuk menampilkan list yang ada pada aplikasi. Terdapat dua fitur pada *command* ini, yaitu LIST DEFAULT dan LIST PLAYLIST.

### 6.5.1 LIST DEFAULT

*Command* LIST DEFAULT dapat menampilkan list artis, list album yang dimiliki artis, serta list lagu yang ada pada album artis. Jika *input* yang diberikan oleh *user* tidak sesuai dengan data yang ada, *user* akan diminta untuk memasukkan *input* kembali.

```

>> LIST DEFAULT;

Daftar Penyanyi :
    1. The Weeknd
    2. Ariana Grande
    3. SZA
    4. One Direction

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

Pilih penyanyi untuk melihat album mereka : One Direction;

Daftar Album oleh One Direction :
    1. Midnight Memories (Deluxe)
    2. FOUR (Deluxe)

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

Pilih album untuk melihat lagu yang ada di album : FOUR (Deluxe);

Daftar lagu di FOUR (Deluxe) :
    1. Steal My Girl
    2. 18
    3. Night Changes
    4. No Control

```

Gambar 6.5.1.1 Tampilan saat *command* LIST DEFAULT dipanggil dan diberikan *input* yang sesuai dengan permintaan

```

>> LIST DEFAULT;

Daftar Penyanyi :
    1. The Weeknd
    2. Ariana Grande
    3. SZA
    4. One Direction

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

=====

>> 

```

Gambar 6.5.1.2 Tampilan saat *command* LIST DEFAULT dipanggil dan diberikan *input* “N” yang langsung menghentikan *command*

```
>> LIST DEFAULT;

Daftar Penyanyi :
  1. The Weeknd
  2. Ariana Grande
  3. SZA
  4. One Direction

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

Pilih penyanyi untuk melihat album mereka : Justin Bieber;

Nama artis tidak ditemukan. Silakan coba lagi!
```

Gambar 6.5.1.3 Tampilan saat *command* LIST DEFAULT dipanggil dan diberikan *input* yang tidak sesuai

## 6.5.2 LIST PLAYLIST

*Command* LIST PLAYLIST dapat menampilkan semua *playlist* yang ada pada akun *user*.

```
>> LIST PLAYLIST;

Daftar playlist yang kamu miliki :
  1. Playlist anin #1
  2. Playlist anin #2

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

Masukkan ID Playlist yang dipilih : 2;
  1. Ariana Grande - My Everything - One Last Time
  2. Ariana Grande - Yours Truly - Almost Is Never Enough
  3. SZA - SOS - Nobody Gets Me
```

Gambar 6.5.2.1 Tampilan saat *command* LIST PLAYLIST dipanggil

## 6.6 PLAY

*Command* PLAY digunakan untuk memutar lagu maupun playlist yang ada. Terdapat dua fitur pada *command* ini, yaitu PLAY SONG dan PLAY PLAYLIST.

### 6.6.1 PLAY SONG

*Command* PLAY SONG digunakan untuk memutar lagu berdasarkan masukan nama penyanyi, nama album, serta id lagu. Saat *command* berhasil dijalankan, *queue* dan *history* lagu akan langsung kosong.

```
>> PLAY SONG;

Daftar Penyanyi :
  1. The Weeknd
  2. Ariana Grande
  3. SZA
  4. One Direction

Masukkan Nama Penyanyi yang dipilih : SZA;

Daftar Album oleh SZA :
  1. SOS
  2. Ctrl

Masukkan Judul Album yang dipilih : SOS;

Daftar Lagu Album SOS oleh SZA :
  1. Kill Bill
  2. Snooze
  3. Nobody Gets Me
  4. Open Arms (feat. Travis Scott)
  5. Good Days

Masukkan ID Lagu yang dipilih : 3;

Berhasil menambahkan lagu "Nobody Gets Me" oleh "SZA" ke queue.
```

Gambar 6.6.1.1 Tampilan saat *command* PLAY SONG dipanggil dan diberikan *input* yang sesuai dengan permintaan

## 6.6.2 PLAY PLAYLIST

*Command* PLAY PLAYLIST digunakan untuk memutar lagu-lagu yang ada pada *playlist*. Lagu pertama pada *playlist* akan menjadi *current song*, sedangkan lagu-lagu sisanya akan masuk ke dalam *queue* dan *history* dalam bentuk urutan yang di-*reverse*.

```
>> PLAY PLAYLIST;

Masukkan ID Playlist: 2;

Memutar playlist "Playlist anin #2".
```

Gambar 6.6.2.1 Tampilan saat *command* PLAY PLAYLIST dipanggil dan diberikan *input* ID Playlist yang sesuai

```
>> PLAY PLAYLIST;

Masukkan ID Playlist: 100;

Playlist tidak ditemukan.
```

Gambar 6.6.2.2 Tampilan saat *command* PLAY PLAYLIST dipanggil dan diberikan *input* ID Playlist yang tidak sesuai (tidak ada di sistem)



## 6.7 QUEUE

*Command* QUEUE digunakan untuk memanipulasi *queue* lagu yang ada. Terdapat lima fitur pada *command* ini, yaitu QUEUE SONG, QUEUE PLAYLIST, QUEUE SWAP, QUEUE REMOVE, dan QUEUE CLEAR.

### 6.7.1 QUEUE SONG

*Command* QUEUE SONG digunakan untuk menambahkan satu lagu ke dalam *queue*.

```
>> QUEUE SONG;

Daftar Penyanyi :
  1. The Weeknd
  2. Ariana Grande
  3. SZA
  4. One Direction

Masukkan Nama Penyanyi yang dipilih : Ariana Grande;

Daftar Album oleh Ariana Grande :
  1. Yours Truly
  2. My Everything
  3. Dangerous Woman

Masukkan Judul Album yang dipilih : My Everything;

Daftar Lagu Album My Everything oleh Ariana Grande :
  1. Problem
  2. One Last Time
  3. Break Free
  4. Love Me Harder

Masukkan ID Lagu yang dipilih : 2;

Berhasil menambahkan lagu "One Last Time" oleh "Ariana Grande" ke queue.
```

Gambar 6.7.1.1 Tampilan saat *command* QUEUE SONG dipanggil dan diberikan *input* yang sesuai dengan permintaan

### 6.7.2 QUEUE PLAYLIST

*Command* QUEUE PLAYLIST dapat menambahkan playlist yang ada ke dalam queue dengan menginput nomor playlist tersebut.

```
>> QUEUE PLAYLIST;

Masukkan ID Playlist yang dipilih : 2;

Berhasil menambahkan playlist "Playlist anin #2" ke queue.
```

Gambar 6.7.2.1 Tampilan saat *command* QUEUE PLAYLIST dipanggil dan diberikan *input* ID Playlist yang sesuai

```
>> QUEUE PLAYLIST;

Daftar Playlist Pengguna :
  1. Playlist anin #1
  2. Playlist anin #2

Masukkan ID Playlist yang dipilih : 2;

Berhasil menambahkan playlist "Playlist anin #2" ke queue.
```

Gambar 6.7.2.2 Tampilan saat *command* QUEUE PLAYLIST dipanggil dan diberikan *input* ID Playlist yang tidak sesuai (tidak ada di sistem)

### 6.7.3 QUEUE SWAP <x> <y>

*Command* QUEUE SWAP <x> <y> dapat menukar lagu dengan indeks x dan lagu dengan indeks y.

```
>> QUEUE SWAP 1 2;

Lagu "Nobody Gets Me" berhasil ditukar dengan "Good Days"
```

Gambar 6.7.3.1 Tampilan saat *command* QUEUE SWAP <x> <y> dipanggil dan diberikan *input* x dan y yang sesuai

```
>> QUEUE SWAP 9 2;

Lagu dengan urutan ke 9 tidak terdapat dalam queue!
```

Gambar 6.7.3.2 Tampilan saat *command* QUEUE SWAP <x> <y> dipanggil dan diberikan *input* x atau y yang tidak sesuai (salah satu indeks tidak terdapat dalam *queue*)

```
>> QUEUE SWAP 10 11;

Lagu dengan urutan ke 10 dan 11 tidak terdapat dalam queue!
```

Gambar 6.7.3.3 Tampilan saat *command* QUEUE SWAP <x> <y> dipanggil dan diberikan *input* x dan y yang tidak sesuai (kedua indeks tidak terdapat dalam *queue*)

### 6.7.4 QUEUE REMOVE <n>

*Command* QUEUE REMOVE <n> dapat menghapus lagu pada *queue* yang ada di urutan ke-n.

Queue :

1. SZA - SOS - Nobody Gets Me
2. One Direction - Midnight Memories (Deluxe) - You & I
3. Ariana Grande - My Everything - One Last Time
4. Ariana Grande - Yours Truly - Almost Is Never Enough

Gambar 6.7.4.1 Tampilan awal *queue*

```
>> QUEUE REMOVE 3;
```

Lagu "Ariana Grande" oleh "One Last Time" telah dihapus dari queue!

Gambar 6.7.4.2 Tampilan saat *command* QUEUE REMOVE <n> dipanggil dan *input* n sesuai

Queue :

1. SZA - SOS - Nobody Gets Me
2. One Direction - Midnight Memories (Deluxe) - You & I
3. Ariana Grande - Yours Truly - Almost Is Never Enough

Gambar 6.7.4.3 Tampilan akhir *queue*

```
>> QUEUE REMOVE 99;
```

Lagu dengan urutan ke 99 tidak terdapat dalam queue!

Gambar 6.7.4.4 Tampilan saat *command* QUEUE REMOVE <n> dipanggil dan *input* n tidak sesuai (tidak ada dalam *queue*)

## 6.7.5 QUEUE CLEAR

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

```
>> QUEUE CLEAR;
```

Queue berhasil dikosongkan.

Gambar 6.7.5.1 Tampilan saat *command* QUEUE CLEAR dipanggil

Queue :

Your queue is empty.

Gambar 6.7.5.2 Tampilan *queue* setelah QUEUE CLEAR dipanggil

## 6.8 SONG

*Command* SONG digunakan untuk menavigasi lagu yang ada pada queue saat ini. Terdapat dua fitur dari *command* SONG, yaitu SONG NEXT dan SONG PREVIOUS.

### 6.8.1 SONG NEXT

*Command* SONG NEXT dapat memutar lagu selanjutnya yang ada pada queue song saat ini. Jika tidak ada *queue song*, maka akan memutar ulang lagu yang sama.

```
>> SONG NEXT;
```

```
Memutar lagu selanjutnya "Kill Bill" oleh "SZA"
```

Gambar 6.8.1.1 Tampilan saat *command* SONG NEXT dipanggil ketika terdapat *queue*

```
>> SONG NEXT;
```

```
Queue kosong, memutar kembali lagu "Kill Bill" oleh "SZA"
```

Gambar 6.8.1.2 Tampilan saat *command* SONG NEXT dipanggil ketika tidak ada *queue*

### 6.8.2 SONG PREVIOUS

*Command* SONG PREVIOUS dapat memutar lagu sebelum lagu yang sedang diputar. Jika tidak ada riwayat lagu sebelumnya, maka akan memutar ulang lagu yang sama.

```
>> SONG PREVIOUS;
```

```
Memutar lagu sebelumnya "Nobody Gets Me" oleh "SZA"
```

Gambar 6.8.2.1 Tampilan saat *command* SONG PREVIOUS dipanggil jika terdapat riwayat lagu

```
>> SONG PREVIOUS;
```

```
Riwayat lagu kosong, memutar kembali lagu "Nobody Gets Me" oleh "SZA"
```

Gambar 6.8.2.2 Tampilan saat *command* SONG NEXT dipanggil jika tidak terdapat riwayat lagu

## 6.9 PLAYLIST

*Command* PLAYLIST digunakan untuk melakukan *basic command playlist*. Terdapat lima fitur pada *command* ini, yaitu PLAYLIST CREATE, PLAYLIST ADD, PLAYLIST SWAP, PLAYLIST REMOVE, dan PLAYLIST DELETE

### 6.9.1 PLAYLIST CREATE

*Command* PLAYLIST CREATE digunakan untuk membuat playlist baru.

```
>> PLAYLIST CREATE;

Masukkan nama playlist yang ingin dibuat : semuaorangbingunggabingungkaloudahdisurga;

Playlist "semuaorangbingunggabingungkaloudahdisurga" berhasil dibuat!

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

Gambar 6.9.1.1 Tampilan saat *command* PLAYLIST CREATE dipanggil

```
>> PLAYLIST CREATE;

Masukkan nama playlist yang ingin dibuat :      n;

Minimal terdapat 3 karakter selain spasi dalam nama playlist. Silakan coba lagi!
```

Gambar 6.9.1.2 Tampilan saat *command* PLAYLIST CREATE dipanggil dan diberikan *input* nama *playlist* yang tidak sesuai ketentuan

### 6.9.2 PLAYLIST ADD

*Command* PLAYLIST ADD digunakan untuk menambahkan lagu pada suatu *playlist* yang telah ada sebelumnya.

#### 6.9.2.1 PLAYLIST ADD SONG

*Command* PLAYLIST ADD SONG digunakan untuk menambahkan satu lagu ke dalam *playlist*

```

>> PLAYLIST ADD SONG;

Daftar Penyanyi :
1. The Weeknd
2. Ariana Grande
3. SZA
4. One Direction

Masukkan Nama Penyanyi yang dipilih : The Weeknd;

Daftar Album oleh The Weeknd :
1. Starboy
2. Dawn FM
3. After Hours

Masukkan Judul Album yang dipilih : Starboy;

Daftar Lagu Album Starboy oleh The Weeknd :
1. Starboy (feat. Daft Punk)
2. Reminder
3. Die For You

Masukkan ID Lagu yang dipilih : 3;

Daftar Playlist Pengguna :
1. Playlist anin #1
2. Playlist anin #2

Masukkan ID Playlist yang dipilih : 2;

Lagu dengan judul "Die For You" pada album Starboy oleh penyanyi The Weeknd berhasil ditambahkan ke dalam playlist Playlist anin #2.

```

Gambar 6.9.2.1.1 Tampilan saat *command* PLAYLIST ADD SONG dipanggil dan diberikan *input* yang sesuai

### 6.9.2.2 PLAYLIST ADD ALBUM

*Command* PLAYLIST ADD ALBUM digunakan untuk menambahkan semua lagu yang ada pada suatu album ke dalam *playlist*.

```

>> PLAYLIST ADD ALBUM;

Daftar Penyanyi :
1. The Weeknd
2. Ariana Grande
3. SZA
4. One Direction

Masukkan Nama Penyanyi yang dipilih : SZA;

Daftar Album oleh SZA :
1. SOS
2. Ctrl

Masukkan Judul Album yang dipilih : SOS;

Daftar Playlist Pengguna :
1. Playlist anin #1
2. Playlist anin #2

Masukkan ID Playlist yang dipilih : 2;

Beberapa lagu dari album "SOS" berhasil ditambahkan ke dalam playlist pengguna "Playlist anin #2".

```

Gambar 6.9.2.2.1 Tampilan saat *command* PLAYLIST ADD ALBUM dipanggil dan diberikan *input* yang sesuai

### 6.9.3 PLAYLIST SWAP ,<id> <x> <y>

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

```
>> PLAYLIST SWAP 2 1 3;
```

Berhasil menukar lagu dengan nama "One Last Time" dengan "Nobody Gets Me" di playlist "Playlist anin #2"

Gambar 6.9.3 Tampilan saat *command* PLAYLIST SWAP dipanggil dan diberikan *input* yang sesuai

#### 6.9.4 PLAYLIST REMOVE <id> <n>

*Command* PLAYLIST REMOVE digunakan untuk menghapus lagu dengan urutan n pada playlist dengan urutan ke n pada playlist dengan index id sesuai dengan urutan index pada kumpulan playlist yang dimiliki oleh user.

```
>> PLAYLIST REMOVE 1 2;
```

Lagu "Tattooed Heart" oleh "Ariana Grande" telah dihapus dari playlist "Playlist anin #1"!

Gambar 6.9.4 Tampilan saat *command* PLAYLIST REMOVE dipanggil dan diberikan *input* yang sesuai

#### 6.9.5 PLAYLIST DELETE

*Command* PLAYLIST DELETE digunakan untuk menghapus playlist dengan ID Playlist yang dipilih. Apabila sebuah playlist dihapus, maka lagu-lagu yang terdapat di dalam playlist yang hendak dihapus juga akan dihapus.

```
>> PLAYLIST DELETE;
```

Daftar Playlist Pengguna :  
1. Playlist anin #1  
2. Playlist anin #2

Masukkan ID Playlist yang dipilih : 2;

Playlist ID 2 dengan judul "Playlist anin #2" berhasil dihapus.

Gambar 6.9.5 Tampilan saat *command* PLAYLIST DELETE dipanggil dan diberikan *input* yang sesuai

### 6.10 STATUS

*Command* STATUS digunakan untuk menampilkan lagu yang sedang diputar dan ada atau tidaknya *queue*. Jika ada *queue*, akan ditampilkan *list queue* lagu

```
>> STATUS;

Now Playing :
SZA - SOS - Nobody Gets Me

Queue :
Your queue is empty.
```

Gambar 6.10.1 Tampilan saat *command* STATUS dipanggil saat tidak ada *queue* lagu

```
>> STATUS;

Now Playing :
SZA - SOS - Nobody Gets Me

Queue :
1. SZA - SOS - Kill Bill
2. The Weeknd - After Hours - Save Your Tears
3. The Weeknd - Starboy - Die For You
4. Ariana Grande - Yours Truly - Almost Is Never Enough
5. One Direction - Midnight Memories (Deluxe) - Strong
```

Gambar 6.10.2 Tampilan saat *command* STATUS dipanggil saat ada *queue* lagu

```
>> STATUS;

Current Playlist :
Night Ride

Now Playing :
One Direction - Midnight Memories (Deluxe) - Story of My Life

Queue :
1. One Direction - FOUR (Deluxe) - Night Changes
2. SZA - SOS - Kill Bill
3. SZA - SOS - Nobody Gets Me
```

Gambar 6.10.3 Tampilan saat *command* STATUS dipanggil saat *queue* hanya terdiri dari satu *playlist* tertentu

## 6.11 SAVE <filename>

*Command* SAVE digunakan untuk menyimpan informasi yang ada selama sesi berjalan seperti artis, album, lagu, *user*, lagu yang sedang dimainkan, *queue*, riwayat, dan *playlist* dalam bentuk “.txt”. Untuk menjalankan *command* ini, pengguna harus masuk ke menu terlebih dahulu dengan menjalankan *command* START/LOAD. Jika sudah masuk ke menu maka pengguna dapat menjalankan *command* SAVE dengan menulis SAVE diikuti dengan nama file yang diinginkan beserta “.txt” .



```
>> SAVE haha.txt;  
Save file berhasil disimpan
```

Gambar 6.11.1 Tampilan saat *command* SAVE dipanggil

## 6.12 QUIT

*Command* QUIT digunakan untuk keluar dari program. Setelah *command* ini dijalankan, pengguna dapat memilih apakah ingin menyimpan data pada sesi saat itu atau tidak. Jika pengguna memilih untuk menyimpan, pengguna diminta untuk menginput nama dari file yang akan disimpan.

```
Save file tidak disimpan  
Kamu keluar dari WayangWave88Gacor. Dadah ^_^/
```

Gambar 6.12.1 Tampilan saat *command* QUIT dipanggil dan menginput “N”

```
Save file berhasil disimpan  
Kamu keluar dari WayangWave88Gacor. Dadah ^_^/
```

Gambar 6.12.2 Tampilan saat *command* QUIT dipanggil dan menginput “Y” serta nama file

## 6.13 HELP

*Command* HELP digunakan untuk menampilkan daftar *command* yang ada beserta deskripsinya kegunaan setiap *command*. Jika *user* belum memasuki aplikasi, daftar *command* yang tersedia ada START, LOAD, HELP, dan QUIT. Jika sudah memasuki aplikasi, tetapi belum melakukan LOGIN, daftar *command* yang tersedia ada LOGIN, SIGN UP, HELP, SAVE, dan QUIT.

```
>> HELP;  
  
===== [ Menu Help WayangWave ] =====  
1. START -> Untuk masuk sesi baru  
2. LOAD -> Untuk memulai sesi berdasarkan file konfigurasi  
3. HELP -> Untuk menampilkan bantuan  
4. QUIT -> Untuk keluar dari aplikasi
```

Gambar 6.13.1 Tampilan saat *command* HELP dipanggil sebelum memasuki aplikasi (belum memanggil START, LOAD, ataupun LOGIN)

```
>> HELP;
```

```
===== [ Menu Help WayangWave ] =====  
1. LOGIN -> Untuk masuk ke akun yang sudah ada  
2. SIGN UP -> Untuk mendaftarkan akun  
3. HELP -> Untuk menampilkan bantuan  
4. SAVE -> Untuk menyimpan state aplikasi terbaru  
5. QUIT -> Untuk keluar dari aplikasi
```

Gambar 6.13.2 Tampilan saat *command* HELP dipanggil setelah memasuki aplikasi, tetapi belum memanggil LOGIN

```
>> HELP;
```

```
===== [ Menu Help WayangWave ] =====  
1. LIST -> Untuk menampilkan daftar playlist, list penyanyi, list album, dan list lagu  
2. PLAY -> Untuk memutar lagu atau playlist yang dipilih  
3. QUEUE -> Untuk memanipulasi queue lagu  
4. SONG -> Untuk melakukan navigasi lagu yang ada pada queue  
5. PLAYLIST -> Untuk melakukan basic command playlist seperti create, add, swap, remove, dan delete  
6. STATUS -> Untuk menampilkan lagu yang sedang dimainkan, queue lagu yang ada, dan playlist lagu diputar  
7. SAVE -> Untuk menyimpan state aplikasi terbaru  
8. HELP -> Untuk menampilkan bantuan  
9. QUIT -> Untuk keluar dari aplikasi  
10. LOGOUT -> Untuk keluar dari sesi dan kembali ke menu utama
```

Gambar 6.13.3 Tampilan saat *command* HELP dipanggil setelah memasuki aplikasi dan sudah memanggil LOGIN

## 6.14 ENHANCE

*Command* ENHANCE digunakan untuk menambahkan lagu-lagu *random* ke dalam *playlist*.

```
>> ENHANCE;
```

```
Berikut daftar playlist yang dimiliki :  
1. Playlist bryan
```

```
Silakan masukkan ID playlist yang ingin dienhance : 1;
```

```
Berhasil menambahkan lagu Story of My Life oleh One Direction ke dalam ID playlist 1.  
Berhasil menambahkan lagu Moonlight oleh Ariana Grande ke dalam ID playlist 1.
```

Gambar 6.14 Tampilan saat *command* ENHANCE dipanggil

## 6.15 LOGOUT

*Command* LOGOUT digunakan untuk keluar dari akun.

```
>> LOGOUT;  
  
Berhasil keluar. Sampai jumpa lagi, "anin" dan selamat beraktivitas kembali!:)
```

Gambar 6.15 Tampilan saat *command* LOGOUT dipanggil

## 6.16 <INVALID COMMAND>

Jika *command* yang di-input tidak sesuai atau merupakan *invalid command*, aplikasi tidak dapat mengeksekusi *command* dan akan memberikan *return* pesan kesalahan.

```
>> LOAD custom.txt;  
  
Command tidak bisa dieksekusi!
```

Gambar 6.16.1 Tampilan saat *command* LOAD dipanggil setelah *command* START sudah dipanggil (LOAD dipanggil setelah masuk ke sesi)

```
>> SIGN UP;  
  
Command tidak bisa dieksekusi!
```

Gambar 6.16.2 Tampilan saat *command* SIGN UP dipanggil sebelum *command* START/LOAD dipanggil (SIGN UP dipanggil sebelum masuk ke sesi)

```
>> NEXT SONG;  
  
Command tidak diketahui!
```

Gambar 6.16.2 Tampilan saat *invalid command* dipanggil (tidak ada *command* NEXT SONG pada sistem)

## 7 Test Script

No	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	START	Memastikan program dapat dijalankan	Mengetikkan <i>command</i> START	START	User berhasil masuk ke aplikasi	Gambar 6.1.1
2	LOGIN	Memastikan <i>username</i> yang dimasukkan terdaftar	Mengetikkan <i>command</i> LOGIN diikuti <i>username</i>	LOGIN, <i>username</i>	User masuk ke akun	Gambar 6.2.1 Gambar 6.2.2
3	LOAD <n>	Memastikan program dapat membaca file konfigurasi	Mengetikkan <i>command</i> LOAD diikuti nama file txt	LOAD, nama file txt	File konfigurasi dibaca	Gambar 6.3.1 Gambar 6.3.2
4	SIGN UP	Memastikan <i>username</i> yang dimasukkan belum pernah terdaftar	Mengetikkan <i>command</i> SIGN UP diikuti <i>username</i> baru	SIGN UP, <i>username</i> baru	User berhasil mendaftarkan <i>username</i> baru	Gambar 6.4.1 Gambar 6.4.2
5	LIST DEFAULT	Memastikan fungsi menunjukkan daftar lagu album artis yang terdaftar	Mengetikkan <i>command</i> LIST DEFAULT diikuti nama artis dan albumnya	LIST DEFAULT, Y/N, artis, album	Muncul daftar lagu milik album artis tertentu	Gambar 6.5.1.1 Gambar 6.5.1.2 Gambar 6.5.1.3
6	LIST PLAYLIST	Memastikan fungsi menunjukkan daftar <i>playlist</i> yang dimiliki <i>user</i>	Mengetikkan <i>command</i> LIST PLAYLIST	LIST PLAYLIST	Muncul daftar <i>playlist</i> yang dimiliki	Gambar 6.5.2.1
7	PLAY SONG	Memastikan lagu yang di-play langsung menjadi <i>now playing</i>	Mengetikkan <i>command</i> PLAY SONG, artis, album, dan ID lagu	PLAY SONG, artis, album, ID lagu	<i>Now playing</i> berubah, <i>queue</i> dan <i>history</i> kosong	Gambar 6.6.1.1
8	PLAY PLAYLIST	Memastikan jika <i>playlist</i> yang di-play sesuai, akan masuk ke <i>now playing</i> , <i>queue</i> , dan <i>history</i>	Mengetikkan <i>command</i> PLAY PLAYLIST, ID Playlist	PLAY PLAYLIST, ID Playlist	Lagu pertama <i>playlist</i> menjadi <i>now playing</i> , lagu sisanya masuk ke <i>queue</i> dan di-reverse ke <i>history</i>	Gambar 6.6.2.1 Gambar 6.6.2.2
9	QUEUE SONG	Memastikan jika lagu yang dipilih sesuai, masuk ke <i>queue</i>	Mengetikkan <i>command</i> QUEUE SONG, artis, album, dan ID lagu	QUEUE SONG, artis, album, ID lagu	Lagu yang dipilih masuk ke dalam <i>queue</i>	Gambar 6.7.1.1
10	QUEUE PLAYLIST	Memastikan jika <i>playlist</i> yang dipilih sesuai,	Mengetikkan <i>command</i> QUEUE PLAYLIST, ID Playlist	QUEUE PLAYLIST, ID Playlist	<i>Playlist</i> yang dipilih masuk ke dalam <i>queue</i>	Gambar 6.7.2.1 Gambar 6.7.2.2

		akan masuk ke <i>queue</i>				
11	QUEUE SWAP <x> <y>	Memastikan jika <i>input</i> x dan y sesuai, urutannya akan tertukar	Mengetikkan <i>command</i> QUEUE SWAP, urutan <i>queue</i> yang ingin ditukar	QUEUE SWAP, urutan yang ingin ditukar	Urutan lagu ke-x dan ke-y pada <i>queue</i> ditukar	Gambar 6.7.3.1 Gambar 6.7.3.2 Gambar 6.7.3.3
12	QUEUE REMOVE <n>	Memastikan jika <i>input</i> n sesuai, lagu pada urutan n di <i>queue</i> akan hilang	Mengetikkan <i>command</i> QUEUE REMOVE, urutan <i>queue</i> yang ingin dihilangkan	QUEUE REMOVE, urutan yang ingin dihilangkan	Lagu ke-n pada <i>queue</i> hilang	Gambar 6.7.4.1 Gambar 6.7.4.2 Gambar 6.7.4.3 Gambar 6.7.4.4
13	QUEUE CLEAR	Memastikan <i>queue</i> kosong saat <i>command</i> dipanggil	Mengetikkan <i>command</i> QUEUE CLEAR	QUEUE CLEAR	<i>Queue</i> kosong	Gambar 6.7.5.1 Gambar 6.7.5.2
14	SONG NEXT	Memastikan memutar lagu selanjutnya yang ada di <i>queue</i>	Mengetikkan <i>command</i> SONG NEXT	SONG NEXT	Lagu yang diputar berubah ke lagu selanjutnya yang ada pada <i>queue</i>	Gambar 6.8.1.1 Gambar 6.8.1.2
15	SONG PREVIOUS	Memastikan dapat memutar lagu yang diputar sebelumnya	Mengetikkan <i>command</i> SONG PREVIOUS	SONG PREVIOUS	Lagu yang diputar berubah ke lagu sebelumnya yang ada pada <i>queue</i>	Gambar 6.8.2.1 Gambar 6.8.2.2
16	PLAYLIST CREATE	Memastikan jika nama <i>playlist</i> belum ada, akan dibuat <i>playlist</i> dengan nama tersebut	Mengetikkan <i>command</i> PLAYLIST CREATE	PLAYLIST CREATE	Terbentuknya <i>playlist</i> baru	Gambar 6.9.1.1 Gambar 6.9.1.2
17	PLAYLIST ADD SONG	Memastikan lagu yang dipilih masuk ke <i>playlist</i>	Mengetikkan <i>command</i> PLAYLIST ADD SONG, artis, album, ID lagu, dan ID Playlist	PLAYLIST ADD SONG, artis, album, ID lagu, ID Playlist	Lagu terpilih masuk ke dalam <i>playlist</i>	Gambar 6.9.2.1.1
18	PLAYLIST ADD ALBUM	Memastikan semua lagu pada album yang dipilih masuk ke <i>playlist</i>	Mengetikkan <i>command</i> PLAYLIST ADD ALBUM, artis, album, dan ID Playlist	PLAYLIST ADD ALBUM, artis, album, ID Playlist	Semua lagu pada album terpilih masuk ke dalam <i>playlist</i>	Gambar 6.9.2.2.1
19	PLAYLIST SWAP	Memastikan jika <i>input</i> ID, x, dan	Mengetikkan <i>command</i>	PLAYLIST SWAP, ID	Urutan lagu ke-x dan ke-y pada	Gambar 6.9.3

	<id> <x> <y>	y sesuai, lagu x dan y pada <i>playlist</i> ID akan tertukar	PLAYLIST SWAP, ID Playlist, urutan lagu pada <i>playlist</i> yang ingin ditukar	Playlist, urutan lagu yang ingin ditukar	<i>playlist</i> ID Playlist ditukar	
20	PLAYLIST REMOVE <id> <n>	Memastikan jika <i>input</i> ID dan n sesuai, lagu n pada <i>playlist</i> ID akan hilang	Mengetikkan <i>command</i> PLAYLIST REMOVE, ID Playlist, urutan lagu pada <i>playlist</i> yang ingin dihapus	PLAYLIST REMOVE, ID Playlist, urutan lagu yang ingin dihapus	Lagu pada <i>playlist</i> ID Playlist urutan ke-n hilang	Gambar 6.9.4
21	PLAYLIST DELETE <n>	Memastikan <i>playlist</i> hilang/terhapuskan saat <i>command</i> dipanggil	Mengetikkan <i>command</i> PLAYLIST DELETE, ID Playlist yang ingin dihapus	PLAYLIST DELETE, ID Playlist	<i>Playlist</i> terpilih hilang dari daftar <i>playlist</i>	Gambar 6.9.5
22	STATUS	Memastikan saat <i>command</i> dipanggil, dimunculkan <i>now playing</i> , <i>queue</i> , dan <i>playlist</i> lagu	Mengetikkan <i>command</i> STATUS	STATUS	Menunjukkan <i>now playing</i> , <i>queue</i> , dan <i>playlist</i> asal lagu diputar	Gambar 6.10.1 Gambar 6.10.2 Gambar 6.10.3
23	SAVE <filename>	Memastikan saat <i>command</i> dipanggil, <i>state</i> aplikasi disimpan	Mengetikkan <i>command</i> SAVE, nama file	SAVE nama file.txt	Menyimpan <i>state</i> aplikasi	Gambar 6.11.1
24	QUIT	Memastikan saat <i>command</i> dipanggil, <i>user</i> keluar dari aplikasi	Mengetikkan <i>command</i> QUIT	QUIT, Y/N, nama file	Keluar dari program serta menyimpan data jika memilih disimpan	Gambar 6.12.1 Gambar 6.12.2
25	HELP	Memastikan fungsi menampilkan daftar <i>command</i> yang ada sesuai <i>state</i> aplikasi	Mengetikkan <i>command</i> HELP	HELP	Menunjukkan daftar <i>command</i> sesuai <i>state</i> aplikasi	Gambar 6.13.1 Gambar 6.13.2 Gambar 6.13.3
26	ENHANCE	Memastikan <i>command</i> dipanggil, <i>user</i> mendapatkan rekomendasi lagu-lagu <i>random</i>	Mengetikkan <i>command</i> ENHANCE, ID Playlist yang ingin ditambahkan lagu-lagu baru	ENHANCE, ID Playlist	Terdapat lagu-lagu baru pada <i>playlist</i> yang dipilih	Gambar 6.14
27	LOGOUT	Memastikan jika <i>command</i> dipanggil, <i>user</i> keluar dari akun	Mengetikkan <i>command</i> LOGOUT	LOGOUT	<i>User</i> keluar dari akun	Gambar 6.15

28	<INVALID COMMAND>	Memastikan jika <i>command</i> yang di- <i>input</i> tidak sesuai, <i>command</i> tidak dapat dieksekusi	Mengetikkan <i>command</i> yang tidak ada pada sistem	<command asal yang tidak ada pada sistem>	<i>Command</i> tidak dapat dieksekusi, memunculkan pesan kesalahan	Gambar 6.16.1 Gambar 6.16.2 Gambar 6.16.3
----	----------------------	--	---	---	--	---

## 8 Pembagian Kerja dalam Kelompok

NIM	Nama	Pembagian Kerja
18222112	Jeremy Deandito	<ul style="list-style-type: none"> <li>- ADT Queue</li> <li>- Fitur List, Playlist</li> <li>- Fitur menampilkan lagu-lagu dari sebuah playlist</li> </ul>
18222128	Anindita Widya Santoso	<ul style="list-style-type: none"> <li>- ADT List Statis, List Dinamis</li> <li>- Fitur Play, Queue, Help</li> <li>- Melakukan <i>formatting</i> laporan</li> <li>- Menyusun laporan dan README</li> <li>- Menyusun Form Asistensi Tugas Besar</li> </ul>
18222130	Bryan P. Hutagalung	<ul style="list-style-type: none"> <li>- ADT Mesin Kata, Mesin Karakter, User</li> <li>- Membuat File Main</li> <li>- Fitur Start, Load</li> <li>- Fitur BONUS Enhance dan Multi User</li> <li>- Membuat Interface</li> </ul>
18222132	Chairul Nur Wahid	<ul style="list-style-type: none"> <li>- ADT Set, Map, Linked List</li> <li>- Fitur Save, Quit</li> </ul>
18222139	Salsabila Azzahra	<ul style="list-style-type: none"> <li>- ADT Stack</li> <li>- Fitur Song, Status</li> <li>- Menyusun laporan</li> </ul>

## 9 Lampiran

### 9.1 Deskripsi Tugas Besar

Membuat sebuah aplikasi simulasi berbasis CLI (*command-line interface*). Sistem aplikasi ini dibuat dalam bahasa C dengan menggunakan struktur data yang sudah dipelajari di mata kuliah Algoritma dan Struktur Data STI. Struktur data yang sudah dibuat pada praktikum dapat digunakan ataupun dimodifikasi untuk mengerjakan Tugas Besar ini.

### 9.2 Notulen Rapat

<b>Rapat 1</b>	<b>Catatan Rapat:</b> <ul style="list-style-type: none"><li>- Menyatukan pikiran dan konsep Tugas Besar</li><li>- Membagi tugas per orang (bagian <i>code</i>) untuk Tugas Besar</li><li>- <i>Workspace</i> Tugas Besar Workspace Tugas Besar Algoritma Strukt...</li><li>- Membuat folder drive dan github terpusat untuk Tugas Besar</li></ul>
<b>Tanggal : 26 Oktober 2023</b>	
<b>Tempat : Lecture Room 2 ITB Jatinangor</b>	
<b>Kehadiran Anggota Kelompok:</b> 18222112 18222128 18222130 18222139	
<b>Rapat 2</b>	<b>Catatan Rapat:</b> <ul style="list-style-type: none"><li>- Membahas setiap struktur dan fungsi ADT dan penggunaannya di Tugas Besar</li><li>- Membahas setiap fungsi COMMAND dan penggunaannya di Tugas Besar</li></ul>
<b>Tanggal : 6 November 2023</b>	
<b>Tempat : Lecture Room 2 ITB Jatinangor</b>	
<b>Kehadiran Anggota Kelompok:</b> 18222112 18222128 18222130 18222139	
<b>Rapat 3</b>	<b>Catatan Rapat:</b> <ul style="list-style-type: none"><li>- Membahas <i>progress report</i></li><li>- Membagi tugas per orang (bagian laporan) untuk Tugas Besar</li></ul>
<b>Tanggal : 15 November 2023</b>	
<b>Tempat : GKU 3 ITB Jatinangor</b>	
18222112 18222128 18222130 18222132 18222139	

### 9.3 Log Activity Anggota Kelompok

Log Act	Anggota	Tanggal
Rapat perdana pembahasan rencana Tugas Besar	18222112, 18222128, 18222130, 18222139	26 Oktober 2023
Pengerjaan ADT List, HELP, dan README	18222128	30 Oktober 2023
Pengerjaan START dan Main	18222130	1 November 2023






Asistensi 1	18222112, 18222128, 18222130, 18222132, 18222139	1 November 2023
Pengerjaan ADT Queue	18222112	2 November 2023
Pengerjaan ADT Mesin Kata dan ADT Mesin Karakter	18222130	5 November 2023
Pengerjaan ADT Set dan ADT Map	18222132	6 November 2023
Pengerjaan ADT Stack	18222139	8 November 2023
Pengerjaan PLAY	18222128	12 November 2023
Pengerjaan PLAYLIST	18222112	14 November 2023
Asistensi 2	18222112, 18222128, 18222130, 18222132, 18222139	15 November 2023
Pengerjaan LIST	18222112	16 November 2023
Pengerjaan QUEUE	18222128	16 November 2023
Pengerjaan MULTI USER	18222130	17 November 2023
Pengerjaan SAVE	18222132	19 November 2023
Pengerjaan STATUS	18222139	19 November 2023
Pengerjaan Laporan	18222112, 18222128, 18222130, 18222132, 18222139	19 November 2023
Pengerjaan ENHANCE	18222130	20 November 2023
<i>Speed Run</i> CODE	18222112, 18222128, 18222130, 18222132, 18222139	20 November 2023
Asistensi 3	18222112, 18222128, 18222130, 18222132, 18222139	22 November 2023
Pengerjaan INTERFACE	18222130	22 November 2023
Fiksasi CODE	18222112, 18222130	23 November 2023
Fiksasi README	18222128	24 November 2023




## 9.4 Form Asistensi Tugas Besar

No. Kelompok/Kelas : 01/03  
 Nama Kelompok : WayangWave88Gacor  
 Anggota Kelompok (Nama/NIM) :  
 1. Jeremy Deandito / 18222112  
 2. Anindita Widya Santoso / 18222128  
 3. Bryan P. Hutagalung / 18222130  
 4. Chairul Nur Wahid / 18222132  
 5. Salsabila Azzahra / 18222139


Asisten Pembimbing : Riandy Hasan






### Asistensi I

<b>Tanggal :</b> 1 November 2023	<b>Catatan Asistensi:</b>
<b>Tempat :</b> Zoom Meeting	
<b>Kehadiran Anggota Kelompok:</b>  <div style="text-align: center;"> <p>1</p>         Jeremy Deandito        18222112     </div> <div style="text-align: center;"> <p>2</p>         Anindita Widya Santoso        18222128     </div> <div style="text-align: center;"> <p>3</p>         Bryan P. Hutagalung        18222130     </div> <div style="text-align: center;"> <p>4</p> </div>	<p><b>UMUM</b>        Inti tugasnya membuat program mirip Spotify dengan nama WayangWave. I.S, F.S diusahakan ada, tapi diutamakan fungsionalitas programnya. Comment dll menyusul aja.</p> <p><b>COMMAND</b>  <b>Start</b> sama <b>Load</b> sama-sama untuk ngebaca konfigurasi, kalo start konfigurasi penyanyi, album, dll. Sedangkan, untuk load ngeload data state yang sudah di Save, termasuk playlist nya.</p> <p><b>Play</b> beda sama nambah di queue. Jadi, kalau ngeplay song/playlist bakal langsung kosongin queue.</p> <p>Fungsi <b>Queue</b> kurang tambah song/playlist ke queue, kalau song cuman satu lagu, kalo playlist ya satu playlist. Swap muter urutan, remove ngebuang isi, dan clear ngosongin semua.</p> <p><b>Song</b> next dan previous ngubah-ngubah lagu yang diplay.</p> <p><b>Playlist</b> ada fungsi buat playlist, nambahin lagu, swap lagu.</p> <p><b>Status</b> bakal print status now playing dan queue.</p> <p><b>Save</b> bisa dicek dari konfigurasi dalam bentuk txt yang ada konfigurasi data album, penyanyi, song, queue, dan playlist. GABOLEH PAKE string compare, harus pake iterasi cek satu-satu (((tapi bakal ditanyain dulu bisa gak pake string)))</p> <p><b>Quit</b> untuk keluar</p>






 Chairul Nur Wahid 18222132 5  Salsabila Azzahra 18222139	<p><b>Invalid Command</b> untuk semisal ada command yang gak ada di list.</p> <p>txt ngikutin format yang udah dikasi, terserah isinya (lagu, album, penyanyi).</p> <p><b>ADT</b>  ADT List nyimpen data playlist, ADT mesin kata buat baca konfigurasi, ADT Queue untuk urutan lagu, ADT Stack untuk riwayat pemutaran, ADT Set &amp; Map untuk simpan lagu, album, sama penyanyi, ADT Linked List untuk nyimpen lagu dalem playlist. SETIAP ADT HARUS ADA DRIVER. Pasti akan diperlukan untuk membuat ADT baru yang nyimpen song, player, sama album</p> <p><b>BONUS</b>  Enhance ngasih lagu-lagu rekomendasi, Multi User buat login sama logout, kalau ada command ini, bakal ada nyimpen data user, savenya juga jadi berubah karena disimpan per user.</p>
	<p><b>Tanda Tangan Asisten:</b></p>  Riandy Hasan 18220058


## Asistensi II

<p><b>Tanggal :</b> 15 November 2023</p> <p><b>Tempat :</b> Zoom Meeting</p> <p><b>Kehadiran Anggota Kelompok:</b>  1    Jeremy Deandito  18222112  2</p>	<p><b>Catatan Asistensi:</b></p> <p><b>PROGRESS REPORT</b></p> <ul style="list-style-type: none"> <li>- ADT sudah semua</li> <li>- File Config udah ada yang buat nge-test &amp; load</li> <li>- Fungsi beberapa udah jadi (terutama <i>start</i>, <i>load</i>, <i>help</i>)</li> </ul> <p><b>QnA</b></p> <ol style="list-style-type: none"> <li>1. Apakah setiap run di gcc harus sepanjang itu perintahnya?  → Supaya gak panjang, bisa pake Makefile; make run atau bisa menggunakan powershell (choco)</li> </ol>
--	---

 Anindita Widya Santoso 18222128 3  Bryan P. Hutagalung 18222130 4  Chairul Nur Wahid 18222132 5  Salsabila Azzahra 18222139	<ol style="list-style-type: none"> <li>2. Pada bagian laporan, apakah perlu nulis notal &amp; C? Atau bagaimana?  → Tulis C nya. Beberapa aja, yang emang diminta di laporannya.</li> <li>3. Setiap fungsi harus ada <i>driver</i> ya kak?  → Yang wajib paling di ADT</li> </ol>
	<p><b>Tanda Tangan Asisten:</b></p>  <b>Riandy Hasan</b> 18220058

### Asistensi III

Tanggal : 22 November 2023	<b>Catatan Asistensi:</b>  <b>PROGRESS REPORT:</b> <ul style="list-style-type: none"> <li>- Udah bisa Makefile (YAY)</li> <li>- Semua fungsi udah berjalan dengan baik</li> </ul> <b>QNA:</b> <ol style="list-style-type: none"> <li>1. Jika PLAY SONG dipanggil, apakah <i>history</i> menjadi kosong? → Iya</li> <li>2. Kalau misal PLAY PLAYLIST, <i>history</i> kosong, <i>queue</i> kosong. Lalu, semisal di SONG PREVIOUS, <i>history</i>-nya isinya apa? → Konsepnya, pas di next baru lagunya masuk ke <i>history</i>. Jadi, <i>history</i> isinya kebalikan dari <i>queue</i></li> <li>3. Apa perbedaan algoritma menarik yang ada di laporan sama spesifikasi tambahan? → Biasanya diisi bonus atau misal ada edit-edit dari ADT</li> <li>4. Bagian <i>playlist</i> di spek ada emoji, jadi kita harus bisa pake emoji kah? → Enggak</li> <li>5. Berarti baru bisa QUIT program setelah LOAD/START ya? → Iya</li> </ol>
Tempat : Zoom Meeting	
Kehadiran Anggota Kelompok:	
<p>1</p>  <p>Jeremy Deandito 18222112</p> <p>2</p>  <p>Anindita Widya Santoso 18222128</p> <p>3</p>  <p>Bryan P. Hutagalung 18222130</p>	
<p>4</p>  <p>Chairul Nur Wahid 18222132</p> <p>5</p>  <p>Salsabila Azzahra 18222139</p>	

	<p><b>Tanda Tangan Asisten:</b></p>  <p><b>Riandy Hasan</b> <b>18220058</b></p>
--	---