

LAPORAN TUGAS BESAR

IF2111 Algoritma dan Struktur Data

BNMO


Dipersiapkan oleh:

Kelompok 6

Nicholas Francis Aditjandra	18221005
Sri Laksmi Purwoningtyas	18221009
Farchan Martha Adji Chandra	18221011
Raden Sjora Okalani	18221014
Fariz Putra Hanggara	18221015
Aniqa Fayyaza Akbar	18221020

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		<i>IF2111-TB1-07</i>		26
		<i>Revisi</i>		11 November 2022

Daftar Isi

1 Ringkasan	3
2 Penjelasan Tambahan Spesifikasi Tugas	4
2.1 Bonus (TicTacToe)	4
3 Struktur Data	4
3.1 ADT ArrayDin	4
3.2 ADT ArrayDinerDash	4
3.3 ADT Mesin Karakter dan Mesin Kata	5
3.4 ADT Queue	6
4 Program Utama	6
5 Data Test	7
5.1 Data Test 1	7
5.2 Data Test 2	8
5.3 Data Test 3	8
5.4 Data Test 4	9
5.5 Data Test 5	9
5.6 Data Test 6	10
5.7 Data Test 7	10
5.8 Data Test 8	11
5.9 Data Test 9	12
5.10 Data Test 10	12
5.11 Data Test 11	13
5.12 Data Test 12	13
5.13 Data Test 13	14
5.14 Data Test 14	14
6 Test Script	15
7 Pembagian Kerja dalam Kelompok	17
8 Lampiran	18
8.1 Deskripsi Tugas Besar 1	18
8.2 Notulen Rapat	21
8.3 Log Activity Anggota Kelompok	21
8.4 Hasil Asistensi	22

1 Ringkasan

Tugas Besar 1 meminta mahasiswa untuk membuat BNMO, sebuah *video game console* berbasis *command-line interface* dalam bahasa C. BNMO memiliki fitur utama, di antaranya memainkan *game*, menambahkan *game*, menghapus *game*, dan mengurutkan *game* yang ingin dimainkan. Program ini dibuat dengan struktur data yang sebelumnya telah dipelajari, seperti ADT Queue, ADT Array Dinamis, ADT Mesin Karakter, dan ADT Mesin Kata. Program ini dimulai dengan menampilkan *welcome page* dan meminta input “START” atau “LOAD”.

Setelah program membaca *input*-an pertama, pemain dapat menginput berbagai *command* yang tersedia, yaitu SAVE, CREATEGAME, LISTGAME, DELETGAME, QUEUEGAME, PLAYGAME, SKIPGAME, QUIT, dan HELP. Berikut penjelasan singkat tentang *command* yang tersedia:

- START: membaca *file* konfigurasi *default*
- LOAD: membaca *file* yang dituliskan
- SAVE: menyimpan *state* game pemain ke dalam suatu *file*
- CREATEGAME: menambahkan *game* baru pada daftar game
- LISTGAME: menampilkan daftar *game* yang tersedia
- DELETGAME: menghapus *game* dari daftar *game*
- QUEUEGAME: mendaftarkan *game* ke dalam *list*
- PLAYGAME: memainkan *game*
- SKIPGAME: melewati permainan
- QUIT: keluar dari program
- HELP: menampilkan *command* yang tersedia

Apabila pemain memasukkan *input*-an yang tidak sesuai dengan *command* yang ada, program akan menampilkan pesan “Command tidak dikenali, silahkan masukkan command yang valid.” dan setelah itu akan terus meminta *input*-an yang sesuai. Terdapat dua game yang tersedia di program ini, yaitu RNG dan Diner Dash. RNG merupakan *game* tebak angka. Pemain memiliki 10 kesempatan untuk menebak angka dari 1-100. Di akhir *game* akan ditampilkan skor. Skor berkurang 10 setiap gagal menebak. Sedangkan Diner Dash merupakan permainan pengantaran makanan berdasarkan prioritas. Jika pemain ingin memainkan *game* selain yang tersedia, program akan menampilkan pesan bahwa *game* tersebut sedang dalam *maintenance* dan belum dapat dimainkan.

Laporan ini berisi penjelasan singkat mengenai persoalan dalam Tugas Besar 1. Selain itu, laporan ini juga memuat penjelasan mengenai spesifikasi fitur, ADT yang digunakan dalam pengerjaan, penjelasan mengenai program utama, *data test*, dan *test script*. Di bagian akhir laporan ini terdapat pembagian kerja kelompok dan berbagai lampiran, di antaranya deskripsi Tugas Besar 1, notulen rapat, *log activity*, dan hasil asistensi.

Di akhir pengerjaan tugas ini, mahasiswa telah berhasil membuat BNMO, sebuah permainan berbasis *command-line interface*. Mahasiswa telah menerapkan bahasa C dan struktur data yang sebelumnya telah dipelajari di kelas dan praktikum. Selain itu, dengan mengerjakan tugas ini, mahasiswa belajar cara membuat program dengan alur yang *complex*.

2 Penjelasan Tambahan Spesifikasi Tugas

2.1 BONUS (TIC TAC TOE)

Game tambahan yang kami buat adalah TIC TAC TOE. *Game* ini merupakan *game* yang dimainkan oleh dua pemain. Pemain bergantian untuk menuliskan X dan O di sebuah 3x3 *board*. Objektif dari *game* ini adalah mendapatkan 1 garis yang berisi X atau O. Jika tidak ada yang menang, maka *game* berakhir seri.

Game ini dapat dijalankan dengan membuat *game* terlebih dahulu dengan nama “TIC TAC TOE”. Kemudian masukkan *game* tersebut ke dalam *queue game*. Saat pemain meng-*input command* PLAY GAME, maka *game* TIC TAC TOE dapat dimainkan.

3 Struktur Data (ADT)

Pada program ini, terdapat 4 buah struktur data (ADT) yang kami gunakan. Berikut daftar dari ADT yang kami gunakan dalam program ini.

3.1 ADT Arraydin

ADT ArrayDin merupakan implementasi dari ADT Array yang digunakan dalam program ini. ADT ini digunakan dalam *command* STARTBNMO, CREATEGAME, LISTGAME, DELETGAME, QUEUEGAME, dan SAVE. Sketsa yang kami gunakan dari ADT ini terdiri dari:

- `int panjangstring(char *str1)`
Digunakan untuk mencari panjang string
- `boolean stringsama(char *str1, char *str2)`
Digunakan untuk memeriksa apakah string1 sama dengan string2
- `ArrayDin MakeArrayDin()`
Digunakan untuk membuat arraydinamis kosong
- `void InsertLast(ArrayDin *array, ElType el)`
Digunakan untuk memasukkan Eltype ke arraygame
- `void DeleteAt(ArrayDin *array, IdxType i)`
Digunakan untuk menghapus game dari arraygame
- `IdxType SearchArrayDin(ArrayDin array, ElType el)`
Digunakan untuk mencari apakah game tersebut sudah ada di dalam arraygame

3.2 ADT ArrayDinerDash

ADT ArrayDinerDash merupakan implementasi dari ADT Array yang digunakan dalam program ini. ADT ini berfungsi untuk membuat game Diner Dash. Sketsa yang kami gunakan dari ADT ini terdiri dari:

- `void InitialArrayDinerDash(ArrayDinerDash *arr)`
Digunakan untuk membuat sebuah *array* yang digunakan pada *game Diner Dash*. Primitif ini digunakan untuk membuat tiga *array* yaitu *array* makanan yang dipesan, makanan yang sedang dimasak, dan makanan yang dapat disajikan.

- `void insertArrayDinerDash(ArrayDinerDash *arr, Order element)`
Digunakan untuk memasukkan elemen ke dalam *array* yang telah dibuat sebelumnya.
- `void DeleteAtDinerDash(ArrayDinerDash *arr, int i)`
Digunakan untuk menghapus elemen *i* pada *array* Diner Dash.
- `void cetakpesanan(ArrayDinerDash *arr)`
Digunakan untuk menampilkan informasi makanan, durasi memasak, ketahanan, dan harga pada *game* Diner Dash yang merupakan isi dari *array* makanan yang dipesan.
- `void cetakmasakan(ArrayDinerDash *arr)`
Digunakan untuk menampilkan informasi makanan dan sisa durasi memasak pada *game* Diner Dash yang merupakan isi dari *array* makanan yang sedang dimasak.
- `void cetaksajian(ArrayDinerDash *arr)`
Digunakan untuk menampilkan informasi makanan dan sisa ketahanan makanan pada *game* Diner Dash yang merupakan isi dari *array* makanan yang dapat disajikan.
- `int randomnumberDinerDash(int min, int max)`
Digunakan untuk mengirimkan sebuah angka *random* dari rentang tertentu yang nantinya dimasukkan pada *array* yang telah dibuat sebelumnya.
- `int cekorder (ArrayDinerDash *arr, int i)`
Digunakan untuk mengecek apakah inputan terdapat pada *array* Diner Dash.

3.3 ADT Mesin Karakter dan Mesin Kata

ADT Mesin Karakter dan ADT mesin kata merupakan ADT yang digunakan untuk membaca masukan yang diberikan oleh user dalam menjalankan program. ADT ini digunakan untuk membaca sebuah kata dan mengubahnya menjadi *string*. Sketsa yang kami gunakan dari ADT ini terdiri dari:

- `void STARTWORD(char *directory)`
Digunakan untuk membaca file konfigurasi dan membaca *input*
- `void ADVWORD()`
Digunakan untuk memajukan satu kata. Primitif ini digunakan saat memajukan kata untuk mendapatkan nama *game*
- `void CopyWord()`
Digunakan untuk meng-*copy* kata yang dibaca dan memasukkannya ke *TabWord* dan kemudian ke *arraygame*
- `void isEndWord()`
Digunakan untuk mengecek apakah kata yang dibaca merupakan *EndWord* (*state* mesin kata).
- `int ToInt (Word word)`
Digunakan untuk mengubah kata yang terbaca dari terminal dan mengubah kata menjadi bentuk *integer*.
- `void intCommand()`
Digunakan untuk membaca *command* dari terminal yang berupa kata.
- `char* stringCommand()`
Digunakan untuk membuat sebuah *string* yang nantinya akan digunakan sebagai perintah *command* pada *game*.

- `void ADVCmd()`
Digunakan untuk memajukan satu kata pada perintah *command*.
- `boolean compareSTR(char input_command[], char command_comp[])`
Digunakan untuk membandingkan *string* masukan dengan perintah *command* yang ada pada *game*. Saat fungsi mengirimkan *true*, akan menjalankan *command* sesuai *string* masukan.

3.4 ADT Queue

ADT Queue merupakan ADT yang digunakan dalam fungsi queue game. Sketsa yang kami gunakan dari ADT ini terdiri dari:

- `void CreateQueue(Queue *q)`
Digunakan untuk membuat queue game kosong
- `boolean isEmpty(Queue q)`
Digunakan untuk memeriksa apakah queue game kosong
- `void enqueue(Queue *q, ElType val)`
Digunakan untuk memasukkan elemen di akhir queue. Primitif ini digunakan ketika *command* QUEUEGAME untuk memasukkan game ke dalam queue
- `void dequeue(Queue *q, ElType *val)`
Digunakan untuk menghapus elemen pertama queue. Primitif ini digunakan untuk *command* SKIPGAME atau ketika permainan sudah selesai
- `void displayQueue(Queue q)`
Digunakan untuk menampilkan queue. Primitif ini digunakan saat menampilkan queue game ketika permainan dimulai dan ketika pemain ingin meng-queue game.

4 Program Utama

Program utama dimulai dengan menampilkan *interface* BNMO dan pemain akan memilih antara *command* START atau *command* LOAD. Saat pemain memasukkan *command* START sistem akan membaca *file* konfigurasi dan menampilkan bahwa sistem BNMO berhasil dijalankan. Sedangkan saat pemain memasukkan *command* LOAD, sistem akan membaca suatu *file* yang di-*input* oleh pemain.

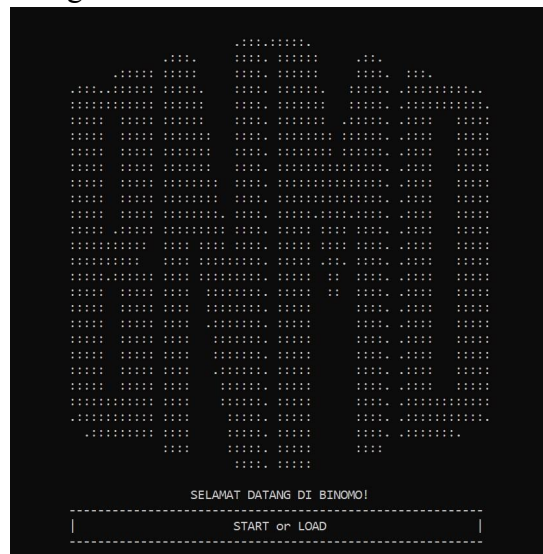
Setelah sistem berhasil membaca konfigurasi, pemain dapat memasukkan *command-command* yang tersedia pada sistem. Yang pertama yaitu, CREATE GAME yang berfungsi menambahkan *game* yang belum ada pada *list game* sesuai masukan dari pemain. Yang kedua yaitu, LIST GAME yang akan menampilkan daftar *game-game* yang tersedia dari *file* konfigurasi yang telah dibaca sebelumnya. Yang ketiga yaitu, DELETE GAME yang berfungsi menghapus *game* pada *list* sebelumnya sesuai nomor *input*-an dari pemain. Yang keempat yaitu, QUEUE GAME untuk menambahkan *game* ke antrian sebelum permainan tersebut dapat dimainkan. Yang kelima yaitu, SKIP GAME. Saat pemain belum menambahkan *game* ke antrian, saat memasukkan *command* SKIP GAME akan menampilkan bahwa antrian kosong. Saat sudah terdapat daftar *game* pada antrian, sistem akan meminta pemain untuk memberikan masukan berupa jumlah *game* yang ingin dilewati. Saat permainan berhasil dilewati, akan menampilkan sisa *game* pada list antrian. Yang keenam yaitu, PLAYGAME. Sebelum memasukkan *command* PLAY

GAME, pemain sebelumnya harus memasukkan *game-game* ke dalam antrian terlebih dahulu melalui *command QUEUE GAME*. Selanjutnya sistem akan menampilkan *list* permainan pada antrian dan memulai permainan sesuai dengan urutan pada antrian *game*. Yang ketujuh yaitu, HELP. *Command HELP* akan menampilkan *command-command* yang tersedia pada sistem. Yang kedelapan yaitu, SAVE yang digunakan untuk menyimpan data permainan pemain ke dalam suatu *file*. Yang kesembilan yaitu, QUIT di mana saat pemain memasukkan *command* ini maka akan keluar dari program permainan. Saat pemain memasukkan *command* yang tidak sesuai, akan menampilkan bahwa *command* tidak dikenali dan pemain diperintah untuk memasukkan *command* baru.

5 Data Test

5.1 Data Test 1

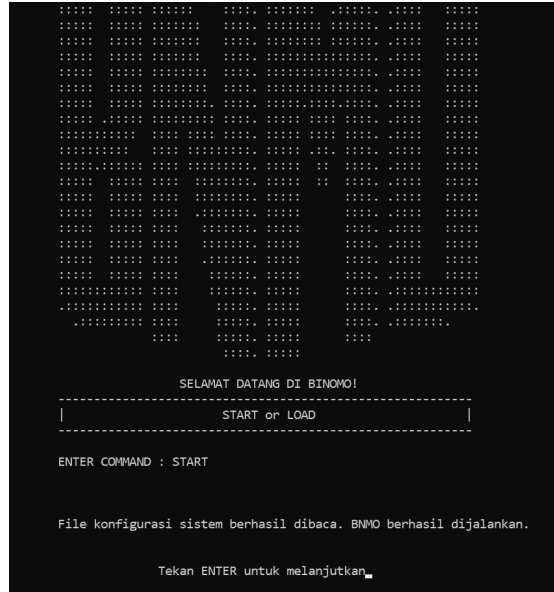
Test ini dilakukan untuk memastikan bahwa program dapat berjalan, menampilkan tampilan video game console BNMO, serta main menu, yaitu START dan LOAD. Cara melakukan kompilasi program adalah dengan run main.c dengan menggunakan debugger ataupun menggunakan perintah gcc main.c -o main.



Gambar 5.1.1 Tampilan Judul dan Main Menu

5.2 Data Test 2

Test ini dilakukan untuk memastikan input START pada main menu bisa dijalankan.



Gambar 5.2.1 Tampilan setelah input “START” dimasukkan

5.3 Data Test 3

Test ini dilakukan untuk memastikan input LOAD pada main menu bisa dijalankan.



Gambar 5.3.1 Tampilan setelah input “LOAD” dimasukkan dan load file berhasil

5.4 Data Test 4

Test ini dilakukan untuk memastikan input HELP pada main menu bisa dijalankan.


```

| KETIK HELP UNTUK MELIHAT LIST COMMAND |
-----

ENTER COMMAND : HELP

===== HELP =====
1. START      - Memulai program Binomo.
2. LOAD       - Melanjutkan progress sebelumnya.
3. SAVE       - Menyimpan progress.
4. CREATE GAME - Menambah game baru pada daftar game.
5. LIST GAME  - Menampilkan daftar game yang tersedia.
6. DELETE GAME - Menghapus game yang tersedia pada list.
7. QUEUE GAME - Menambah suatu game ke dalam antrean yang ingin dimainkan.
8. PLAY GAME  - Memulai permainan.
9. SKIPGAME <n> - Melewatkan permainan sebanyak <n>.
10. QUIT      - Keluar dari Binomo.
11. HELP     - Melihat command yang aktif.

Tekan ENTER untuk melanjutkan

```

Gambar 5.4.1 Tampilan setelah input “HELP” dimasukkan

5.5 Data Test 5

Test ini dilakukan untuk memastikan input CREATE GAME pada main menu bisa dijalankan.

```

-----
| KETIK HELP UNTUK MELIHAT LIST COMMAND |
-----

ENTER COMMAND : CREATE GAME
Masukkan nama game yang akan ditambahkan : VALORANT

Game berhasil ditambahkan

Tekan ENTER untuk melanjutkan

```

Gambar 5.5.1 Tampilan setelah input “CREATE GAME” dimasukkan

5.6 Data Test 6

Test ini dilakukan untuk memastikan input LIST GAME pada main menu bisa dijalankan.

```

-----
| KETIK HELP UNTUK MELIHAT LIST COMMAND |
-----

ENTER COMMAND : LIST GAME

Berikut adalah daftar game yang tersedia
1. RNG
2. DINNER DASH
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER
6. FIFA
7. STRAY CAT
8. VALORANT

Tekan ENTER untuk melanjutkan

```

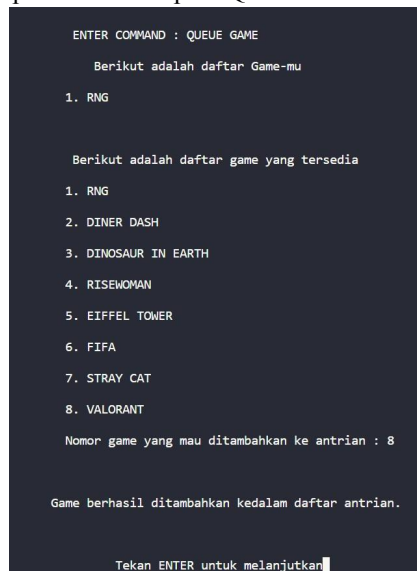
Gambar 5.6.1 Tampilan setelah input “LIST GAME” dimasukkan

5.7 Data Test 7

Test ini dilakukan untuk memastikan input QUEUE GAME pada main menu bisa dijalankan.



Gambar 5.7.1 Tampilan setelah input “QUEUE GAME” dimasukkan



Gambar 5.7.2 Berhasil menambahkan game ke antrean

```
| KETIK HELP UNTUK MELIHAT LIST COMMAND |  
-----  
  
ENTER COMMAND : QUEUE GAME  
  
Berikut adalah daftar Game-mu  
  
1. RNG  
2. VALORANT  
3. RNG  
  
Berikut adalah daftar game yang tersedia  
  
1. RNG  
2. DINER DASH  
3. DINOSAUR IN EARTH  
4. RISEWOMAN  
5. EIFFEL TOWER  
6. FIFA  
7. STRAY CAT  
8. VALORANT  
  
Nomor game yang mau ditambahkan ke antrian : |
```

Gambar 5.7.3 Berhasil menambahkan game yang sudah ditambahkan sebelumnya ke antrian

```
| KETIK HELP UNTUK MELIHAT LIST COMMAND |  
-----  
  
ENTER COMMAND : QUEUE GAME  
  
Berikut adalah daftar Game-mu  
  
1. RNG  
2. DINER DASH  
  
Berikut adalah daftar game yang tersedia  
  
1. RNG  
2. DINER DASH  
3. DINOSAUR IN EARTH  
4. RISEWOMAN  
5. EIFFEL TOMER  
6. FIFA  
7. STRAY CAT  
  
Nomor game yang mau ditambahkan ke antrian :  
  
Nomor permainan tidak valid, silahkan masuk nomor game pada list.  
  
Tekan ENTER untuk melanjutkan|
```

Gambar 5.7.4 Gagal menambahkan game ke antrian

5.8 Data Test 8

Test ini dilakukan untuk memastikan input PLAY GAME pada main menu bisa dijalankan.



Gambar 5.8.1 Tampilan setelah input “PLAY GAME” dimasukkan



Gambar 5.8.2 Memainkan game buatan pemain akan langsung menghasilkan skor.

5.9 Data Test 9

Test ini dilakukan untuk memastikan input SKIP GAME pada main menu bisa dijalankan.



Gambar 5.9.1 Tampilan setelah input “SKIPGAME <nomor antrean game yang ingin di skip>” dimasukkan

5.10 Data Test 10

Test ini dilakukan untuk memastikan input DELETE GAME pada main menu bisa dijalankan.

```
| KETIK HELP UNTUK MELIHAT LIST COMMAND |
-----

ENTER COMMAND : DELETE GAME

Berikut adalah daftar game yang tersedia
1. RNG
2. DINER DASH
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER
6. FIFA
7. STRAY CAT
8. VALORANT
Masukkan nomor game yang akan dihapus: 8

Game berhasil dihapus

Tekan ENTER untuk melanjutkan
```

Gambar 5.10.1 Tampilan setelah input “DELETE GAME” dimasukkan dan memberikan input game yang ingin dihapus

5.11 Data Test 11

Test ini dilakukan untuk memastikan game RNG bisa dimainkan.

```
Game RNG dimulai!

Tebaklah angka dari 0-100!
-----
Anda mempunyai kesempatan menebak sebanyak 10 kali
Masukkan tebakan Anda: _
```

Gambar 6.11.1 Tampilan saat game RNG dimulai

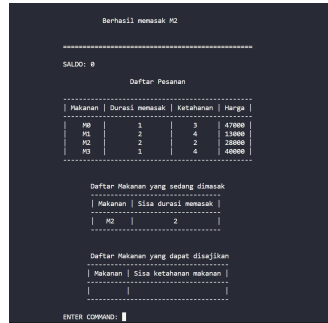
```
Tebaklah angka dari 0-100!
-----
Anda mempunyai kesempatan menebak sebanyak 7 kali
Masukkan tebakan Anda: 30

-----
Tebakan Anda benar!
-----
Score Anda = 70
-----
Tekan ENTER untuk melanjutkan _
```

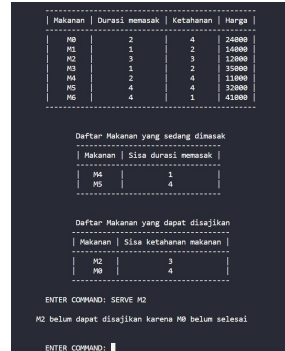
Gambar 6.11.2 Tampilan saat berhasil menebak di game RNG

5.12 Data Test 12

Test ini dilakukan untuk memastikan game Diner Dash bisa dimainkan.



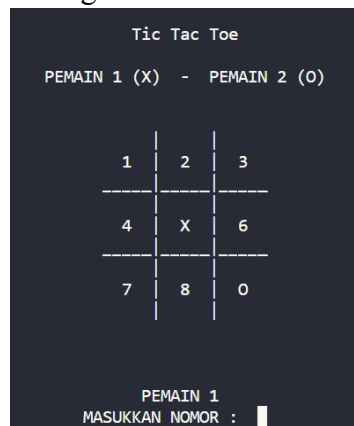
Gambar 5.12.1 Tampilan setelah memasukkan COOK M2



Gambar 5.13.1 Tampilan saat pemain memasukkan perintah SERVE, tetapi bukan pada antrian paling depan

5.13 Data Test 13

Test ini dilakukan untuk memastikan game Tic Tac Toe bisa dimainkan.



Gambar 5.13.1 Tampilan saat bermain TIC TAC TOE

5.14 Data Test 14

Test ini dilakukan untuk memastikan input QUIT pada main menu bisa dijalankan.



Gambar 5.14.1 Tampilan setelah input “QUIT” dimasukkan

6 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Fitur START	Memeriksa apakah program dapat menjalankan command START	Memasukkan command START ketika program meminta input di main menu	Data Test 2	File konfigurasi sistem berhasil dibaca dan bisa melanjutkan untuk mengisi command lainnya	Sesuai dengan yang diharapkan
2	Fitur LOAD	Memeriksa apakah program dapat menjalankan command LOAD	Memasukkan command LOAD <nama file> ketika program meminta input di main menu	Data Test 3	Berhasil membaca file yang ada di command, berhasil menjalankan program, dan bisa melanjutkan untuk mengisi command lainnya	Sesuai dengan yang diharapkan
3	Fitur CREATE GAME	Memeriksa apakah program dapat menjalankan command CREATE GAME	Memasukkan command CREATE GAME ketika program meminta input di main menu, memasukkan input nama game yang ingin ditambahkan ketika diminta program	Data Test 5	Menambahkan nama game ke file save.txt	Sesuai dengan yang diharapkan
4	Fitur LIST GAME	Memeriksa apakah program dapat menjalankan command LIST GAME	Memasukkan command LIST GAME ketika program meminta input di main menu	Data Test 6	Menampilkan list game yang tersedia	Sesuai dengan yang diharapkan

5	Fitur DELETE GAME	Memeriksa apakah program dapat menjalankan command DELETE GAME	Memasukkan command DELETE GAME ketika program meminta input di main menu, memasukkan input nomor game yang akan dihapus ketika diminta program	Data Test 10	Menghapus game sesuai dengan nomor game yang di input	Sesuai dengan yang diharapkan
6	Fitur QUEUE GAME	Memeriksa apakah program dapat menjalankan command QUEUE GAME	Memasukkan input nomor game sesuai daftar game yang tersedia	Data Test 7	Menampilkan antrian game, Menambahkan game ke antrian sesuai dengan nomor game yang di input. Urutan antrian disesuaikan dengan urutan mendaftarkan. Game yang diantrekan bisa game yang sama	Sesuai dengan yang diharapkan
	Fitur QUEUE GAME	Memeriksa apakah program dapat menjalankan command QUEUE GAME	Memasukkan input nomor game tidak sesuai daftar game yang tersedia (tidak valid)	Data Test 7	Menampilkan antrian game, meminta user untuk memasukkan input yang sesuai dengan nomor game yang ada pada list	Sesuai dengan yang diharapkan
	Fitur PLAY GAME	Memeriksa apakah program dapat menjalankan command PLAY GAME	Memainkan game yang ada di save.txt	Data Test 8	Menampilkan daftar game yang telah diantrekan dan melanjutkan untuk memainkan game yang berada di nomor 1	Sesuai dengan yang diharapkan
	Fitur PLAY GAME	Memeriksa apakah program dapat menjalankan command PLAY GAME	Memainkan game yang tidak ada di save.txt	Data Test 8	Menampilkan daftar game yang telah diantrekan dan menampilkan skor yang di generate secara random	Sesuai dengan yang diharapkan

	Fitur SKIP GAME	Memeriksa apakah program dapat menjalankan command SKIPGAME	Memasukkan command SKIPGAME <nomor antrean game yang ingin di skip> ketika program meminta input di main menu	Data Test 9	Menampilkan sisa game yang ada setelah memasukkan command SKIPGAME	Sesuai dengan yang diharapkan
	Fitur QUIT	Memeriksa apakah program dapat menjalankan command QUIT	Memasukkan command QUIT ketika program meminta input command di main menu	Data Test 14	Program akan keluar dari game BNMO	Sesuai dengan yang diharapkan
	Fitur HELP	Memeriksa apakah program dapat menjalankan command QUIT	Memasukkan command HELP ketika program meminta input command di main menu	Data Test 4	Menampilkan command yang bisa digunakan dan kegunaannya	Sesuai dengan yang diharapkan
	Fitur RNG	Memeriksa apakah program dapat memainkan game RNG	Memasukkan input angka tebakan maksimal sebanyak 10 kali	Data Test 11	Menampilkan score sesuai jumlah tebakan yang di input user	Sesuai dengan yang diharapkan
	Fitur Diner Dash	Memeriksa apakah program dapat memainkan game Diner Dash	Memasukkan command COOK <kode makanan> ketika program meminta input command, Memasukkan command SERVE <kode makanan> ketika program meminta input command	Data Test 12	Memasukkan kode makanan ke dalam tabel makanan yang sedang dimasak, tidak dapat SERVE karena bukan antrean paling depan	Sesuai dengan yang diharapkan
	Fitur Tic Tac Toe	Memeriksa apakah program dapat memainkan game Tic Tac Toe	Pemain bergantian untuk memasukkan input X dan O	Data Test 13	Mendapatkan 1 garis yang berisi X atau O. Jika tidak ada yang menang, maka game berakhir seri.	Sesuai dengan yang diharapkan

7 Pembagian Kerja dalam Kelompok

No.	Fitur/ADT	NIM Coder	NIM Tester
1.	Main Program	18221011	18221011

2.	START	18221015	18221011
3.	LOAD	18221015 18221011	18221011
4.	CREATEGAME	18221015	18221011
5.	LISTGAME	18221020	18221020 18221005
6.	DELETGAME	18221015 18221011	18221005 18221009
7.	QUEUEGAME	18221015	18221011 18221014 18221009
8.	PLAYGAME	18221015 18221011	18221005
9.	SKIPGAME	18221015 18221011	18221009 18221014
10.	QUIT	18221005	18221015 18221020
11.	HELP	18221009	18221014
12.	RNG	18221020	18221011 18221015
13.	DINER DASH	18221015 18221011 18221005	18221015 18221020
14.	TIC TAC TOE (Game Bonus)	18221011 18221015	18221015 18221020
15.	Laporan	18221009 18221014 18221020 18221005 18221011 18221015	

8 Lampiran

8.1 Deskripsi Tugas Besar 1

Buatlah sebuah permainan 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 Mechanics

1. About the System

BNMO merupakan suatu robot game console yang dapat menjalankan permainan. BNMO memiliki beberapa fitur utama, yaitu:

1. Memainkan game
2. Menambahkan game
3. Menghapus game
4. Mengurutkan game yang akan dimainkan

2. Main Menu

Ketika program pertama kali dijalankan, BNMO akan memperlihatkan main menu yang berisi welcome page dan beberapa menu pilihan yaitu START dan LOAD. Setelah itu, main menu akan menerima input commands yang akan dijelaskan pada bagian berikutnya.

3. Command

Pada setiap giliran, pemain dapat memasukkan command-command berikut:

a. START

START merupakan salah satu command yang dimasukkan pertama kali oleh pemain ke BNMO. Setelah menekan Enter, dibaca file konfigurasi default yang berisi list game yang dapat dimainkan.

b. LOAD <filename>

LOAD merupakan salah satu command yang dimasukkan pertama kali oleh pemain ke BNMO. Memiliki satu argumen yaitu filename yang merepresentasikan suatu *save file* yang ingin dibuka. Setelah menekan Enter, akan dibaca save file <filename> yang berisi list game yang dapat dimainkan, histori dan scoreboard game, lebih detailnya bisa dilihat pada Konfigurasi Sistem.

- c. **SAVE <filename>**
SAVE merupakan command yang digunakan untuk menyimpan state game pemain saat ini ke dalam suatu file. Command SAVE memiliki satu argumen yang merepresentasikan nama file yang akan disimpan pada disk.
- d. **CREATEGAME**
CREATEGAME merupakan command yang digunakan untuk menambahkan game baru pada daftar game. Spesifikasi game yang dibuat dapat dilihat pada section [Spesifikasi Game](#)
- e. **LISTGAME**
LISTGAME merupakan command yang digunakan untuk menampilkan daftar game yang disediakan oleh sistem.
- f. **DELETGAME**
DELETGAME merupakan command yang digunakan untuk menghapus sebuah game dari daftar game. Adapun aturan penghapusan game adalah:
- Game yang dapat dihapus hanya game yang dibuat secara custom oleh pengguna.
 - 5 game pertama pada file konfigurasi tidak dapat dihapus.
 - Game yang saat itu terdapat di dalam queue game tidak dapat dihapus.
- g. **QUEUEGAME**
QUEUEGAME merupakan command yang digunakan untuk mendaftarkan permainan kedalam list. List dalam queue akan hilang ketika pemain menjalankan command QUIT.
- h. **PLAYGAME**
PLAY GAME merupakan command yang digunakan untuk memainkan sebuah permainan. Game yang dimainkan adalah game dengan urutan pertama di antrian game. Ketika salah satu permainan dimulai, sistem akan menjalankan game sesuai pada section [Spesifikasi Game](#). Permainan selain yang dispesifikasikan pada [Spesifikasi Game](#) akan menampilkan pesan bahwa game tidak dapat dimainkan.
- i. **SKIPGAME <n>**
SKIPGAME merupakan command yang digunakan untuk melewati permainan sebanyak n.
- j. **QUIT**
Keluar dari program.
- k. **HELP**
Bantuan command-command yang disebutkan di atas. Tampilan dan kata-kata dibebaskan.
- l. **COMMAND LAIN**
Command-command lain selain yang disebutkan diatas tidak valid. Keluar dari program.

8.2 Notulen Rapat

Rapat Pertama (30 Oktober 2022)

- Pembagian Tugas
- Mulai mengerjakan tugas secara mandiri

Rapat Kedua (9 November 2022)

- Mulai pembuatan laporan

Rapat Ketiga (10 November 2022)

- Menyempurnakan diner dash
- Menyelesaikan laporan
- Merapikan source code
- Memperbaiki bug

8.3 Log Activity Anggota Kelompok

No.	Tanggal	NIM	Nama	Aktivitas
1.	29/10/22	18221011	Farchan Martha	Membuat grup line
2.	30/10/22	18221005 18221009 18221011 18221014 18221015 18221020	Nicholas Francis Sri Laksmi Farchan Martha Sjora Okalani Fariz Putra Aniqa Fayyaza	Pembagian tugas
3.	3/11/22	18221014	Sjora Okalani	Membuat repository dan menginvite anggota di github
4.	3/11/22	18221005 18221009 18221011 18221014 18221015 18221020	Nicholas Francis Sri Laksmi Farchan Martha Sjora Okalani Fariz Putra Aniqa Fayyaza	Asistensi 1
5.	30/11/22 - 9/11/12	18221005 18221009 18221011 18221014 18221015 18221020	Nicholas Francis Sri Laksmi Farchan Martha Sjora Okalani Fariz Putra Aniqa Fayyaza	Pengerjaan tugas dengan mandiri sesuai dengan pembagian yang telah dilakukan.
6.	9/11/22	18221005 18221009 18221011 18221014 18221015 18221020	Nicholas Francis Sri Laksmi Farchan Martha Sjora Okalani Fariz Putra Aniqa Fayyaza	Asistensi 2
7.	10/11/22	18221005 18221009 18221011	Nicholas Francis Sri Laksmi Farchan Martha	Menyelesaikan laporan

		18221014 18221015 18221020	Sjora Okalani Fariz Putra Aniqa Fayyaza	
--	--	----------------------------------	---	--



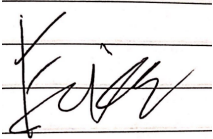

8.4 Hasil Asistensi

Form Asistensi Tugas Besar IF2110/Algoritma dan Struktur Data Sem. 1 2022/2023




No. Kelompok/Kelas : 06/K03
 Nama Kelompok : Three Point Six
 Anggota Kelompok (Nama/NIM) :
 1. Nicholas Francis Aditjandra / 18221005
 2. Sri Laksmi Purwoningtyas / 18221009
 3. Farchan Martha Adji Chandra / 18221011
 4. Raden Sjora Okalani / 18221014
 5. Fariz Putra Hanggara / 18221015
 6. Aniqa Fayyaza Akbar / 18221020

Asisten Pembimbing : Kadek Surya Mahardika



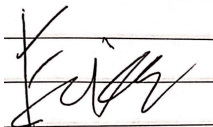

Asistensi I

Tanggal : 3 November 2022	Catatan Asistensi:
Tempat : https://itb-ac-id.zoom.us/j/2113314820?pwd=VEIyM0k4bkSyUmxFOXc3dz09	
Kehadiran Anggota Kelompok:	
No NIM Tanda tangan	
<div data-bbox="418 510 522 567">1 18221020</div> <div data-bbox="324 577 592 808"></div> <div data-bbox="418 814 522 871">2 18221014</div> <div data-bbox="381 913 511 1018"></div> <div data-bbox="418 1123 522 1180">3 18221015</div> <div data-bbox="365 1186 576 1323"></div> <div data-bbox="418 1365 522 1421">4 18221005</div> <div data-bbox="381 1438 544 1606"></div> <div data-bbox="418 1654 522 1711">5 18221009</div>	




1. Q: Score di RNG masih perlu ditampilkan?
2. A: Iya, ditampilkan aja tapi nggak disave.
3. Q: History dan scoreboard disimpan di mana?
4. A: Iya, scoreboard sudah dihapus, jadi hanya menyimpan list permainan.
5. Q: Kan list nya boleh dimodifikasi kak, kalo buat MARK nya gitu boleh?
6. A: Iya, boleh.
7. Q: Kalau dia mau mulai game baru, dia langsung nge STARTGAME atau lanjut ngequeue selanjutnya?
8. A: Harusnya semuanya mulai dari STARTGAME
9. Q: Kalo mau nambahin game yang udah ada di antrian, ditulis lagi atau inputannya malah ga diterima?
10. A: Tetep ditulis lagi kalo udah ada, di bawah

 <p>6 18221011</p> 	
	<p>Tanda Tangan Asisten:</p>  <p>Kadek Surya Mahardika 13519165</p>

Asistensi II

Tanggal : 9 November 2022	Catatan Asistensi:
Tempat : bit.ly/skuuyyy	
Kehadiran Anggota Kelompok:	
No	
NIM	
Tanda tangan	
1	1. Sudah bisa START dan sudah berhasil dibaca oleh sistem. Hampir semua command sudah masuk, kecuali diner dash yang sudah dibuat tapi belum dimasukkan.
18221020	2. Q: Penghitungan score RNG bagaimana?
	3. A: Jika menebak pada first try skornya 100. Tiap tries dikurangi 10.
2	4. Q: Di bagian mesin karakter, aku mau nge-import mesinkarakter.h itu harus pake directory path yang panjang banget, menurut kakak itu kenapa ya? Soalnya nanti kalo dikumpulin ke olympia nya bakal ngikutin directorynya.
18221014	5. A: Bisa gak kalo pake titik dua gitu?
	6. Q: Pas di-run tulisannya no such file or directory
3	7. A: ..ADT nya dihapus coba
18221015	
	
4	
18221005	
	
5	
18221009	

1. Sudah bisa START dan sudah berhasil dibaca oleh sistem. Hampir semua command sudah masuk, kecuali diner dash yang sudah dibuat tapi belum dimasukkan.
2. Q: Penghitungan score RNG bagaimana?
3. A: Jika menebak pada *first try* skornya 100. Tiap *tries* dikurangi 10.
4. Q: Di bagian mesin karakter, aku mau nge-import mesinkarakter.h itu harus pake directory path yang panjang banget, menurut kakak itu kenapa ya? Soalnya nanti kalo dikumpulin ke olympia nya bakal ngikutin directorynya.
5. A: Bisa gak kalo pake titik dua gitu?
6. Q: Pas di-run tulisannya no such file or directory
7. A: ..ADT nya dihapus coba

 <p>6 18221011</p> 	
	<p>Tanda Tangan Asisten:</p>  <p>Kadek Surya Mahardika 13519165</p>