

# **LAPORAN TUGAS BESAR**

## **IF2111 Algoritma dan Struktur Data STI**

### **BNMO**

Dipersiapkan oleh:

Kelompok 13

Muhammad Naufal - 18221089

Ibnu Khairy Algifari - 18221091


Christopher Febrian Nugraha - 18221115

Natasya Vercelly Harijadi - 18221119

Kevin Prayoga Abdinegara - 18221141

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-TB1-13</i>		24
		<i>Revisi</i>	<i>1</i>	<i>11-11-2022</i>

# Daftar Isi

<b>Daftar Isi</b>	<b>2</b>
<b>1 Ringkasan</b>	<b>4</b>
<b>2 Penjelasan Tambahan Spesifikasi Tugas</b>	<b>4</b>
2.1 (Bonus) Permainan Minesweeper	4
<b>3 Struktur Data (ADT)</b>	<b>4</b>
3.1 ADT Mesin Karakter	4
3.2 ADT Mesin Kata	5
3.3 ADT Mesin File	5
3.4 ADT ArrayDin	5
3.5 ADT Queue	5
3.6 ADT Queue_dinerdash	6
<b>4 Program Utama</b>	<b>6</b>
<b>5 Data Test</b>	<b>7</b>
5.1 Data Test 1 : WELCOME PAGE	7
5.2 Data Test 2 : MAIN MENU	7
5.3 Data Test 3 : LOAD	7
5.4 Data Test 4 : QUEUE DAN LIST GAME	8
5.5 Data Test 5 : DELETE GAME	8
5.6 Data Test 6 : CREATE GAME	9
5.7 Data Test 7 : SKIP GAME DAN PLAY GAME	9
5.8 Data Test 8 : HELP	10
5.9 Data Test 9 : SAVE DAN QUIT	10
5.10 Data Test 10 : RNG	10
5.11 Data Test 11 : Diner DASH	11
5.12 Data Test 12 : Spesifikasi Custom Game	11
<b>6 Test Script</b>	<b>12</b>
<b>7 Pembagian Kerja dalam Kelompok</b>	<b>16</b>
<b>8 Lampiran</b>	<b>18</b>
8.1 Deskripsi Tugas Besar 2	18
8.2 Notulen Rapat	18

8.3 Log Activity Anggota Kelompok
8.4 Form Asistensi Tugas Besar

19
20

# 1 Ringkasan

Indra dan Doni memiliki sebuah robot *video game console* bernama BNMO yang bisa memainkan sejumlah permainan. BNMO rusak dua bulan yang lalu, tetapi akhirnya berhasil diperbaiki. Namun, BNMO memiliki beberapa *bug* dalam sistemnya setelah diperbaiki. Akhirnya, Indra dan Doni mencari *programmer* yang andal untuk memprogram ulang BNMO.

BNMO adalah sebuah *game* berbasis CLI (*command-line interface*) yang mampu memainkan sejumlah permainan, menambahkan game, mengurutkan game yang ingin dimainkan, dan menghapus permainan. BNMO memiliki beberapa game *default* yang dapat dimainkan yaitu Diner Dash dan Random Number Generator.

Laporan ini berisi tentang penjelasan program-program yang telah dibuat. Bagian pertama mencakup ringkasan yang di dalamnya terdapat deskripsi umum persoalan, isi umum laporan secara singkat, dan kesimpulan tentang hasil tugas besar secara umum. Bagian kedua berisi penjelasan tambahan spesifikasi fitur-fitur yang belum rinci dari deskripsi tugas besar. Bagian ketiga menjelaskan struktur data yang digunakan untuk mengatasi persoalan pada BNMO. Bagian keempat berisi penjelasan program utama, fitur-fitur yang ada pada BNMO. Bagian keenam dan ketujuh adalah bagian testing, diisi dengan data dan script yang digunakan untuk menguji program. Bagian kedelapan berisi pembagian tugas tiap anggota. Bagian terakhir berisi lampiran-lampiran.

Permainan BNMO ini dibuat dengan menggunakan bahasa C dan ADT yang sudah dipelajari dalam mata kuliah IF2111 Algoritma dan Struktur Data STI. ADT yang digunakan pada program BNMO kali ini yaitu array dinamis, mesin file, mesin karakter, mesin kata, queue, dan queue\_dinerdash.

## 2 Penjelasan Tambahan Spesifikasi Tugas

### 2.1 (Bonus) Permainan Minesweeper

Permainan Minesweeper dimulai dengan menginisiasi jumlah baris, kolom, dan tingkat kesulitan (probabilitas sebuah *cell* berisi *mine* yang “terkubur”). Program akan mengacak letak *mine*. Kemudian, pemain akan menebak *cell* mana yang tidak berisi *mine*. Permainan akan berakhir saat pemain telah menemukan seluruh *mine* dan membuka seluruh *cell* yang kosong (menang), atau membuka *cell* yang berisi *mine* (kalah, *game over*).

Pada program ini, kami menggunakan ADT mesin kata untuk menerima input. Beberapa input diubah menjadi integer (contohnya input jumlah baris dan kolom untuk kemudian dibentuk menjadi *minefield*), beberapa diubah menjadi char (contohnya input apakah pemain mau mencoba ulang game (*retry*)).

## 3 Struktur Data (ADT)

### 3.1 ADT Mesin Karakter

ADT Mesin Karakter berupa program yang membaca “pita karakter” dan mampu untuk mengeksekusi fungsi dan prosedur primitif, diantaranya merupakan START dan ADV. Persoalan yang diselesaikan adalah persoalan input dari program yang akan memanfaatkan ADT ini. Alasan ADT ini dipilih adalah karena input kebanyakan berupa kata-kata yang terdiri dari spasi,

karakter, dan newline selaku *mark* atau penanda. ADT mesin karakter dideklarasikan di *mesinkar.h*, diimplementasikan di *mesinkar.c*, dan diuji di *drivermesinkarkata.c*.

### 3.2 ADT Mesin Kata

ADT Mesin Kata berupa program yang membaca “pita kata” yang merupakan “pita karakter” dan mampu untuk mengeksekusi primitif dari ADT Mesin Kata. Primitif yang diimplementasikan oleh kami yakni *STARTWORD*, *ADVWORD*, dan *CopyWord*. Persoalan yang diselesaikan adalah persoalan input dari program yang akan memanfaatkan ADT ini. Alasan ADT ini dipilih adalah karena input kebanyakan berupa kata-kata yang terdiri dari spasi, karakter, dan titik selaku *mark* atau penanda yang dapat dijadikan sebuah kata keseluruhan karena kata adalah urutan karakter. ADT mesin kata dideklarasikan di *mesinkata.h*, diimplementasikan di *mesinkata.c*, dan diuji di *drivermesinkarkata.c*.

### 3.3 ADT Mesin File

ADT Mesin File berupa program yang membaca “pita file” yang merupakan “pita kata” yang isinya nama file. ADT Mesin File diimplementasikan dengan memodifikasi ADT Mesin Kata sehingga mampu mengeksekusi *STARTFILE*, *ADVFILE*, *IgnoreNewLine*, *ADVWORDFILE*, dan *CopyWordFile*. Persoalan yang diselesaikan adalah persoalan input berupa nama file dari program yang akan memanfaatkan ADT ini. Alasan ADT ini dipilih adalah karena ada input berupa nama file yang diperlakukan berbeda dari input yang lain (kata, angka) dan digunakan saat me-load atau save pada file tertentu. ADT ini dideklarasikan di *mesinfile.h*, diimplementasikan di *mesinfile.c*. (Pengujian didasari keberhasilan ADT dalam melakukan save atau load file, sehingga tidak diperlukan driver terpisah.)

### 3.4 ADT ArrayDin

ADT ArrayDin berupa program yang menyediakan array penyimpanan elemen. ADT ArrayDin diimplementasikan agar mampu mengeksekusi fungsi dan prosedur primitif, seperti *MakeArrayDin*, *DeallocateArrayDin*, *IsEmpty*, *Length*, *Get*, *GetCapacity*, *InsertAt*, *InsertLast*, *InsertFirst*, *DeleteAt*, *DeleteLast*, *DeleteFirst*, *PrintArrayDin*. Persoalan yang diselesaikan adalah persoalan penyimpanan hasil input dari ADT-ADT sebelumnya dan proses perbandingannya dengan data yang tersimpan di file. Alasan ADT ini dipilih adalah karena penyimpanan berupa array dinamik yang dapat disesuaikan dengan input dari ADT Mesin Kata serta file penyimpanan. ADT ini diimplementasikan sebagai ADT array dinamik beserta informasi besar kapasitas dan elemen efektif yang ada di storage. ADT arraydin dideklarasikan di *arraydin.h*, diimplementasikan di *arraydin.c*, dan diuji di *driverarraydin.c*.

### 3.5 ADT Queue

ADT Queue berupa program yang menampung elemen-elemen dalam bentuk sebuah antrian dan mampu untuk mengeksekusi fungsi dan prosedur primitif, diantaranya merupakan *dequeue* dan *enqueue*. Persoalan yang diselesaikan adalah persoalan game DinerDash dari program yang akan memanfaatkan ADT ini. Alasan ADT ini dipilih adalah karena input berupa elemen-elemen yang berada pada urutan antrian dan memerlukan proses masuk-keluar antrian serta proses lainnya. ADT ini diimplementasikan sebagai ADT array statik dengan informasi

indeks elemen *Head* (“paling depan”) dan *Tail* (“paling belakang”) yang memudahkan penyimpanan dan pemrosesan elemen di program. ADT queue dideklarasikan di `queue.h`, diimplementasikan di `queue.c`, dan diuji di `driverqueue.c`

### 3.6 ADT Queue\_dinerdash

ADT Queue\_dinerdash merupakan ADT queue yang dimodifikasi agar sesuai dengan kebutuhan game Diner Dash. ADT ini dibuat untuk mengatasi permasalahan tipe data baru yakni Food dan Cook. Struktur pada Food memuat `id`, `time`, `expired`, dan `price`. Sementara struktur pada tipe Cook memuat `id`, `cookLeft`, dan `serveLeft`. Semua primitif ADT queue dibuat untuk kedua jenis data ini (food dan cook). ADT queue\_dinerdash dideklarasikan di `queue_dinerdash.h`, diimplementasikan di `queue_dinerdash.c`, dan diuji di `driverqueue_dinerdash.c`.

## 4 Program Utama

Program utama dengan nama file `main.c` akan meng-include semua *file header* dari ADT dan semua file fungsi/fitur pada BNMO. Ketika program utama dijalankan, pertama akan menampilkan menu START dan LOAD. Jika pemain memilih START, pemain akan masuk ke program BNMO dengan menggunakan file `config.txt`. Jika pemain ingin menggunakan file konfigurasi lain, pemain harus melakukan LOAD. Setelah melakukan LOAD, pemain otomatis masuk ke program BNMO. Program utama BNMO akan terus berlangsung selama pemain belum memanggil fungsi QUIT.

Pada program BNMO, pemain harus menjalankan fungsi QUEUE GAME terlebih dahulu karena pemain hanya dapat bermain (PLAY GAME) dari permainan yang ada pada antriannya. Untuk saat ini, pemain hanya dapat memainkan game RNG, Diner Dash, dan Minesweeper.

Pada game RNG, pemain diminta untuk memasukan input angka untuk menebak angka yang di *generate* oleh program. Setiap tebakan angka yang salah, program akan memberi tahu angka tebakan lebih kecil atau lebih besar dari angka yang ditebak. Hasil akhir tebakan ditentukan berdasarkan seberapa banyak user menebak. Dalam game RNG ini, input diasumsikan adalah sebuah integer.

Pada game Diner DASH, pemain akan memainkan permainan mengantar makanan yang berdasarkan prioritas yang ditentukan. Hanya ada beberapa command yang diterima untuk setiap putaran game. Untuk command yang invalid, putaran pada permainan akan diabaikan. Setiap makanan perlu dimasak sampai waktu tertentu, dan kemudian siap disajikan dengan ketahanan tertentu. Setiap makanan yang ketahanannya habis, maka makanan perlu dimasak kembali. Setiap makanan yang disajikan akan menghasilkan uang yang dimasukkan ke saldo. Saldo akan menjadi skor akhir dari permainan.

Jika pemain memilih untuk memainkan game Minesweeper, pemain pertama akan diminta masukan untuk menentukan ukuran *minefield*. *Mine* akan ditanam pada suatu sel acak, kemudian diberi nama *seed*. Pemain diminta untuk menebak di mana letak *seed* tersebut dengan memberi *flag*. Pemain juga bisa memilih untuk membuka sel. Saat membuka sel, bila ada angka pada sel tersebut, artinya di dekat sel tersebut ada *seed* yang terpasang. Angka yang muncul menandakan jumlah *seed* yang ada di sekitar sel tersebut. “Di sekitar sel” didefinisikan sebagai sekeliling sel dengan radius 1 sel. Pemain harus memanfaatkan *clue* ini untuk menebak letak *seed* dan memenangkan permainan.



```

ENTER COMMAND: START
File konfigurasi sistem berhasil dibaca. BNMO berhasil dijalankan.

Press ENTER key to continue...

```

Secara default, “SAVE” akan berhasil karena program menentukan sendiri data yang digunakan.

```

Selamat datang di BNMO!
Silahkan pilih perintah yang ingin anda lakukan!

>> START

>> LOAD

ENTER COMMAND: LOAD alstrokedead.txt
File konfigurasi sistem tidak ditemukan.

Press ENTER key to back...

```

“LOAD <namafile>.txt” akan gagal apabila file yang dimasukan tidak tersimpan di folder data.

#### 5.4 Data Test 4 : QUEUE DAN LIST GAME

Test 4 ini digunakan untuk menguji fitur “QUEUE GAME” dan “LIST GAME”, list game akan menampilkan game yang tersimpan pada save file yang telah dipilih, kemudian queue game digunakan untuk menerima input nomor game dari list game untuk dimasukan dalam queue permainan.

```

===== M E N U =====

ENTER COMMAND: QUEUE GAME

Berikut adalah daftar antrian Game-mu
Antrian Kosong.

Berikut adalah daftar game yang tersedia :
1. RNG
2. Diner DASH
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER
Masukkan nomor game yang ingin dimainkan: 6
Game tidak tersedia.

Press ENTER key to back...

```

Jika input nomor game tidak tersedia di list game, makan queue game akan gagal.

#### 5.5 Data Test 5 : DELETE GAME

Test 5 digunakan untuk menguji command “DELETE GAME”. Proses penghapusan akan gagal apabila game yang dihapus tidak ada pada list.



```

===== M E N U =====
ENTER COMMAND: DELETE GAME

Berikut adalah daftar game yang tersedia :
1. RNG
2. Diner DASH
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER
Masukkan nomor game yang akan dihapus: 6

Game gagal dihapus
Press ENTER key to back...

```

### 5.6 Data Test 6 : CREATE GAME

Test 6 digunakan untuk menguji command “CREATE GAME”. Proses pembuatan game berhasil selama game yang dibuat tidak terdapat di list game.

```

ENTER COMMAND: CREATE GAME

Masukkan nama game yang akan ditambahkan: Diner DASH
Diner DASH
Game sudah tersedia.

Press ENTER key to back...

```

### 5.7 Data Test 7 : SKIP GAME DAN PLAY GAME

Test 7 ini digunakan untuk menguji command “SKIP GAME”, dalam command “SKIP GAME”, program otomatis akan melakukan command “PLAY GAME” setelah melakukan skip.

```

ENTER COMMAND: QUEUE GAME

Berikut adalah daftar antrian Game-mu
1. RNG
2. Diner DASH
3. DINOSAUR IN EARTH

Berikut adalah daftar game yang tersedia :
1. RNG
2. Diner DASH
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER
6. Minesweeper

```

Queue sebelum melakukan skip game.

```

ENTER COMMAND: SKIPGAME 2
Game berhasil di-SKIP

Berikut adalah daftar antrian Game-mu :
    1. DINOSAUR IN EARTH

Press ENTER key to play DINOSAUR IN EARTH...

```

Setelah melakukan skip game, kemudian akan otomatis memainkan game.

## 5.8 Data Test 8 : HELP

Test 8 ini digunakan untuk menampilkan command “HELP” yang berisi penjelasan tiap command.

```

ENTER COMMAND: HELP
===== HELP =====
1. CREATE GAME - Menambahkan game baru pada daftar game
2. LIST GAME - Menampilkan daftar game yang tersedia pada sistem
3. DELETE GAME - Menghapus game yang dibuat secara custom oleh pengguna
4. QUEUE GAME - Mendaftarkan permainan ke dalam list game
5. PLAY GAME - Memainkan sebuah permainan yang tersedia pada sistem
6. SKIP GAME - Melewatkan permainan sebanyak n
7. QUIT - Keluar dari program

Press ENTER key to back...

```

## 5.9 Data Test 9 : SAVE DAN QUIT

Test 9 ini digunakan untuk menguji command “SAVE” dan “QUIT”.

```

ENTER COMMAND: QUIT
Apakah anda ingin melakukan SAVE? (Y/N) Y
Masukkan nama file: save1.txt
Save file berhasil disimpan.
Anda keluar dari game BNMO.
Bye bye...

```

Dalam command “QUIT” program memberi pilihan untuk menyimpan atau tidak file yang telah diupdate (jika ada) sebelumnya. Apabila user ingin menyimpan, maka akan masuk ke program save dan menerima masukan nama file sebagai penyimpanan. Program akan segera berakhir setelah program menyimpan. Jika user tidak ingin menyimpan, maka program akan segera berakhir.

## 5.10 Data Test 10 : RNG

Test 10 ini digunakan untuk menguji permainan RNG.

```

R.--. | N.--. | G.--. |
:(): | :(): | :/\: |
(): | (): | :V: |
'--'R| '--'N| '--'G|
-----
Selamat datang di game RNG!

RNG Telah dimulai. Uji keberuntungan Anda dengan menebak angka.
Tebakan : 10
Lebih besar.

```

Program men-*generate* random number untuk ditebak. Ketika menerima masukan yang salah, program akan memberikan hint berdasarkan masukan sebelumnya.

```

Tebakan : 30
Ya, Tebakan adalah 30
Skor akurasi anda adalah 33/100.

```

Ketika masukan benar, program akan selesai dan menunjukkan skor.

### 5.11 Data Test 11 : Diner DASH

Test 11 digunakan untuk menguji bagian Diner DASH. Dalam permainannya hanya menerima command “COOK M<nomor makanan>”, “SERVE M<nomor makanan>”, dan “SKIP”. Setiap command yang invalid, tidak dianggap sebagai satu putaran.

```

COMMAND : MASAK

COMMAND TIDAK VALID
=====

SALDO: 0

Daftar Pesanan
Makanan | Durasi memasak | Ketahanan | Harga
-----
M0      | 2              | 2         | 17000
M1      | 1              | 1         | 22000
M2      | 4              | 2         | 34000

```

### 5.12 Data Test 12 : Spesifikasi Custom Game

Test 12 digunakan untuk menguji game yang dibuat oleh user berdasarkan command “CREATE GAME”.

```

Skor akhir Anda = 33

Press ENTER key to back...

```

Game langsung berakhir dengan menampilkan skor yang nilainya di-*random* dari program.

### 5.13 Data Test 13 : Game Minesweeper

Test 12 digunakan untuk menguji game bonus. Dalam permainan ini, user akan memasukkan input berupa 3 buah bilangan untuk menginisiasi papan permainan (koordinat x, y, dan tingkat kesulitan). Setelah papan terinisiasi, pemain harus menebak apakah sebuah *cell* berisi *mine* atau tidak. Bila berisi *mine*, maka pemain dinyatakan kalah. Bila semua *cell* yang tidak berisi *mine* berhasil dibuka, maka pemain dinyatakan menang. Skor akhir akan ditampilkan beserta pilihan untuk bermain kembali atau tidak.

```

      y      0  1
x
0      X  ?
1      ?  ?
2      ?  ?
3      ?  X
4      X  ?
5      X  ?
6      ?  ?
Anda kalah!
Skor Anda: 0
Bermain lagi? (y / n):

```

## 6 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Main	Memastikan program dapat dijalankan	Buka Command prompt  Input gcc -o <semua file yang berkaitan dengan main>  Enter	dapat dilihat pada bagian 5.1	Program menampilkan tulisan "Welcome To BNMO" dan sebuah gambar	Program menampilkan tulisan "Welcome To BNMO" dan sebuah gambar
2	Compile semua fungsi	Memastikan fungsi dapat dipanggil	Buka Command prompt	Dapat dilihat pada bagian 5.2	Program akan menampilkan tampilan main menu	Tampilan main menu dan meminta masukan

		dan digunakan	Input gcc -o <semua file yang berkaitan dengan main>  Enter  Input main  Enter			command apapun
3	LOAD	Menguji fitur load dan save untuk file default config.txt	Di main menu:  Masukkan command LOAD  Diikuti dengan nama file.txt  Enter	Dapat dilihat pada bagian 5.3	Program menampilkan tulisan "File konfigurasi sistem berhasil dibaca. BNMO berhasil dijalankan".	Program menampilkan tulisan "File konfigurasi sistem berhasil dibaca. BNMO berhasil dijalankan".
4	QUEUE GAME	Menguji fitur queue game	Di main menu:  Masukkan command QUEUE GAME  Enter  Masukkan angka pada list game yang ingin ditambahkan ke dalam queue  Enter	Dapat dilihat pada bagian 5.4	Program akan menampilkan isi Queue game dan juga menampilkan list game, serta akan meng <i>enqueue</i> game ya	Isi queue game ditampilkan disertai dengan list game dan game yang diinputkan sudah masuk ke dalam queue
5	LIST GAME	Menguji fitur list game	Di main menu:  Masukkan command LIST GAME  Enter	Dapat dilihat pada bagian 5.4	Program akan menampilkan semua list game yang sudah dibuat termasuk pada file config	Semua game yang ada di list game ditampilkan
6	DELETE GAME	Menguji fitur delete game	Di main menu:  Masukkan command DELETE GAME  Enter  Masukkan angka pada list game yang ingin dihapus  Enter	Dapat dilihat pada bagian 5.5	Program akan menghapus nama game yang berada pada urutan list game, pengecekan dapat dilakukan dengan lakukan command LIST GAME	Nama game yang ingin dihapus terhapus

7	CREATE GAME	Menguji fitur create game	Di main menu:  Masukkan command CREATE GAME  Enter  Lalu program akan meminta masukan sebuah nama game  Enter	Dapat dilihat pada bagian 5.6	Program akan memasukkan nama game tersebut ke dalam list game kemudian hasilnya dapat dicek menggunakan command LIST GAME	Game yang diinginkan masuk ke dalam list
8	SKIP GAME <n>	Menguji fitur skip game	Di main menu:  Masukkan command SKIP GAME <n>  n diisi jumlah game yang ingin di skip  Enter	Dapat dilihat pada bagian 5.7	Program akan melakukan dequeue terhadap queue game sebanyak n kali kemudian game yang valid akan langsung dimainkan	Isi queue berkurang dan game langsung dapat dimainkan
9	PLAY GAME	Menguji apakah game dapat dimainkan	Di main menu:  Masukkan command PLAY GAME  Enter	Dapat dilihat pada bagian 5.7	Program akan memainkan game yang berada pada urutan pertama dalam queue game. Program akan mengecek apakah game tersebut dapat dimainkan/tida k. Jika dapat, maka laman menu akan berubah menjadi laman game	Game yang dapat dimainkan akan langsung dapat dijalankan, apabila tidak dapat muncul pesan bahwa game tidak dapat dimainkan silahkan mencari game lain
10	HELP	Menguji fungsi help	Di main menu  Masukkan command HELP  Enter	Dapat dilihat pada bagian 5.8	Program akan menampilkan semua fitur yang ada pada main menu disertai penjelasannya	Semua fitur dan penjelasannya ditampilkan
11	SAVE	Menguji fitur save dan mengecek apakah	Di main menu:  Masukkan command SAVE	Dapat dilihat pada bagian 5.9	Program akan melakukan save terhadap state program yang dapat	Nama file sudah bisa digunakan untuk load

		semua progress dapat disimpan	Enter  Masukkan nama file disertai dengan .txt  Enter		dicek dengan cara melihat langsung pada folder data, atau juga dapat dicek setelah quit, lalu dilakukan load sesuai nama file yang sudah dibuat tadi. Jika berhasil maka fitur save sudah berjalan	
12	QUIT	Menguji fitur QUIT dan integrasi dengan fitur SAVE	Di main menu:  Masukkan command QUIT  Enter  Masukkan huruf Y/N karena program akan meminta masukan apakah state program di save atau tidak  Enter  # Jika Y Masukkan nama file dengan .txt  Enter  #Jika N Enter	Dapat dilihat pada bagian 5.9	Jika di save, maka program akan melakukan perintah save terlebih dahulu, kemudian program akan melakukan dequeue semua isi queue game dan program akan menjalankan fungsi exit. Program otomatis selesai. Hal serupa terjadi apabila tidak di save, yang berbeda hanya program tidak akan melakukan save dan akan mengembalikan state awal program.	Program keluar dengan adanya file txt baru apabila di save
13	Game RNG	Menguji apakah game RNG dapat dijalankan	Di main menu lakukan prosedur play game dengan RNG berada di urutan queue pertama  Atau gunakan fitur skip game	Dapat dilihat pada bagian 5.10	Program akan menampilkan laman utama dari game RNG dan user dapat memasukkan	Laman program menjadi laman game dan user dapat memainkan game

			<p>Pada laman game, program akan meminta masukan sebuah angka</p> <p>Apabila angka meleset dari angka yang sudah ditetapkan program, program akan memberikan petunjuk</p> <p>Masukkan angka lagi</p> <p>Ulangi terus hingga angka tertebak</p> <p>Selesai</p>		input serta bermain hingga game selesai	
14	Game DINER DASH	Menguji apakah game diner dash dapat dimainkan	<p>Di main menu ada 2 pilihan:</p> <p>&gt; Menggunakan PLAY GAME</p> <p>Masukkan command QUEUE GAME lalu enter</p> <p>Input nomor yang menunjukkan urutan game DINER DASH lalu enter</p> <p>Enter</p> <p>Masukkan command PLAY GAME lalu enter</p> <p>&gt; Menggunakan SKIP GAME dengan asumsi game diner dash berada di lokasi n pada queue</p> <p>Masukkan command SKIP GAME</p> <p>Diikuti dengan angka senilai n-1</p> <p>Enter</p>	Dapat dilihat pada bagian 5.11	Program akan otomatis memulai game diner dash. dengan ketentuan yang sudah dijelaskan.	Game diner dash siap dimainkan
15	Game Custom	Menguji apakah game custom dapat dijalankan,	<p>Di main menu lakukan prosedur play game dengan game custom berada di urutan queue pertama</p>	Dapat dilihat pada bagian 5.12	Program akan menampilkan skor akhir pemain dan meminta pemain untuk	Skor akhir ditampilkan dan permainan selesai



		serta mengecek apakah game hasil buatan user dapat dijalankan	Atau gunakan fitur skip game  Enter  Selesai		menekan enter untuk keluar dari permainan	
16	Game Minesweeper	Menguji apakah game minesweeper dapat dijalankan pengguna				

## 7 Pembagian Kerja dalam Kelompok

No.	Fitur	NIM Coder	NIM Tester
1	ADT Mesin karakter (primitif dan driver)	18221089, 18221141	18221089, 18221141
2	ADT Mesin kata (primitif dan driver)	18221089, 18221141	18221089, 18221141
3	ADT Mesin File (primitif dan driver)	18221141	18221089, 18221141
4	ADT ArrayDin (primitif dan driver)	18221089, 18221141	18221089, 18221141
5	ADT Queue (primitif dan driver)	18221089, 18221141	18221089, 18221141
6	ADT Queue_dinerdash (primitif dan driver)	18221089	18221089, 18221141
7	Prosedur Start	18221141	18221089, 18221091, 18221115, 18221119, 18221141
8	Prosedur Load <filename>	18221141	18221089, 18221091, 18221115, 18221119, 18221141
9	Prosedur Save <filename>	18221141	18221089, 18221091, 18221115, 18221119, 18221141
10	Prosedur CreateGame	18221089	18221089, 18221091, 18221115, 18221119, 18221141
11	Prosedur ListGame	18221119	18221089, 18221091, 18221115, 18221119, 18221141

12	Prosedur DeleteGame	18221141	18221089, 18221091, 18221115, 18221119, 18221141
13	Prosedur QueueGame	18221091	18221089, 18221091, 18221115, 18221119, 18221141
14	Prosedur PlayGame	18221141	18221089, 18221091, 18221115, 18221119, 18221141
15	Prosedur Skip Game <n>	18221091	18221089, 18221091, 18221115, 18221119, 18221141
16	Prosedur Quit	18221091	18221089, 18221091, 18221115, 18221119, 18221141
17	Prosedur CommandLain	18221119	18221089, 18221091, 18221115, 18221119, 18221141
18	Game RNG	18221089	18221089, 18221091, 18221115, 18221119, 18221141
19	Game DinerDash	18221089	18221089, 18221091, 18221115, 18221119, 18221141
20	Game Tambahan	18221089	18221089, 18221091, 18221115, 18221119, 18221141
21	Game Minesweeper	18221115	18221089, 18221091, 18221115, 18221119, 18221141
22	Main Program	18221089, 18221141	18221089, 18221091, 18221115, 18221119, 18221141

## 8 Lampiran

### 8.1 Deskripsi Tugas Besar 2

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`

## 8.2 Notulen Rapat

### Rapat 30 Oktober 2022

Rapat ini merupakan rapat perdana yang membahas bagaimana kami melakukan pembagian tugas. Awalnya tugas dibagi berdasarkan fitur utama BNMO dengan rincian sebagai berikut:

- a. START - Kevin
- b. LOAD <filename> - Toper
- c. CREATE GAME - Naufal
- d. LIST GAME - Natasya
- e. DELETE GAME - Kevin
- f. QUEUE GAME - Ibnu
- g. PLAY GAME - Kevin
- h. SKIP GAME <n> - Ibnu
- i. QUIT - Ibnu
- j. HELP - Natasya
- k. COMMAND LAIN - Natasya
- l. RNG - Naufal
- m. Diner Dash - Naufal
- n. Game tambahan/buatan pemain - Naufal
- o. Main program - Toper
- p. Bonus - Toper

Selanjutnya, ADT yang digunakan: queue, arraydin, mesinkar, mesinkata menggunakan ADT yang telah ada di praktikum. Driver pada setiap ADT akan dibuat kemudian. Struktur program yang disepakati seperti ini:

```
> data
    | config.txt
> src
    | > ADT
        | arraydin
        | queue
        | mesinkar
        | mesinkata
    | implementasi dari seluruh command
```

### Rapat 5 November 2022

Rapat ini dilaksanakan atas dasar revisi-revisi yang ada pada asistensi 1 tanggal 4 November 2022, revisi tersebut meliputi:

- Mengubah pembacaan mesin kata dengan berhenti pada newline (“\n”)

- Membuat ADT baru yang masih menggunakan dasar sesuai yang dipelajari di kelas yang disesuaikan dengan kebutuhan prosedur/fungsi
- Membuat main.c

Berdasarkan hasil asistensi tersebut, kami merasa perlu diadakan meet agar dapat membahas tindak lanjut dari hasil revisi tersebut, kami menghendaki supaya minimal kerangka main.c sudah selesai dilanjutkan dengan finishing fungsi dan prosedur lain. Rapat ini dilaksanakan secara hybrid dengan ke ITB lalu menggunakan zoom, mengingat waktu yang cukup fleksibel dan beberapa teman kami belum berada di Bandung. Rapat kali ini menghasilkan:

- Kerangka main.c
- Adanya ADT baru yang disesuaikan dengan kebutuhan fungsi/prosedur

### **Rapat 9 November 2022**

Pada rapat ini, kami membahas:

- kemajuan progress masing-masing dalam menyelesaikan finishing program
- melanjutkan penggabungan fungsi-fungsi yang sudah dibuat masing-masing dan menyamakan ADT yang sudah dibuat supaya kami dapat melakukan debugging menggunakan main.c.
- Kemudian pada rapat ini kami juga membahas kemajuan pengerjaan laporan tugas besar yang pembagiannya sudah dibahas pada rapat sebelumnya.

Rapat kali ini menghasilkan: revisi minor dan mayor terhadap beberapa fungsi dan prosedur maupun header file yang telah dibuat, atas hasil dari diskusi bersama dan debugging menggunakan main.c.

### **Rapat 10 November 2022**

Pada rapat ini, kami membahas hasil asistensi ke-2 yang dilaksanakan hari ini juga, kemudian melanjutkan finishing program dan laporan. rapat kali ini dilaksanakan secara offline, karena rapat dilaksanakan langsung sehabis kelas.

## **8.3 Log Activity Anggota Kelompok**

<b>Waktu</b>	<b>NIM</b>	<b>Keterangan</b>
30-10-2022	18221089, 18221091, 18221115, 18221119, 18221141	Rapat perdana membahas rencana tugas besar dan pembagian tugas
01-11-2022	18221141	Membuat ADT dan fungsi tambahan
02-11-2022	18221141	Merevisi ADT dan fungsi tambahan
03-11-2022	18221089, 18221119, 18221141	Push command yang dibuat
04-11-2022	18221089, 18221091, 18221119, 18221141	Update command, push command yang dibuat, menambahkan ADT, dan membuat main

05-11-2022	18221089, 18221091, 18221141	Update fungsi, revisi ADT, dan menambahkan ADT baru.
07-11-2022	18221119	Update command buatan
09-11-2022	18221089, 18221091, 18221115, 18221119, 18221141	Rapat evaluasi progress program dan pengerjaan laporan
10-11-2022	18221089, 18221091, 18221115, 18221119, 18221141	Mengerjakan laporan, merevisi ADT, merevisi command, dan membuat program utama
11-11-2022	18221089, 18221091, 18221119, 18221141	Mengerjakan laporan, merevisi ADT, merevisi command, dan merevisi program utama

## 8.4 Form Asistensi Tugas Besar

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

No. Kelompok/Kelas : 13/K01  
 Nama Kelompok : Alstrokedead  
 Anggota Kelompok (Nama/NIM) : 1. Muhammad Naufal/18221089  
 2. Christopher Febrian Nugraha/18221115





STEI- ITB	IF2111-TB1-13	Halaman 21 dari 25 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		



3. Kevin Prayoga Abdinegara/18221141
4. Natasya Vercelly Harijadi/18221119
5. Ibnu Khairy Algifari/18221091

Asisten Pembimbing






: Aditya Bimawan/13519064

### Asistensi I


<b>Tanggal : 04 November 2022</b>	<b>Catatan Asistensi:</b> <ul style="list-style-type: none"> <li>- Pembacaan file config kayak diner dash biar bisa baca kalimat, jadi baca file config itu per baris, bukan per kata sehingga mengabaikan blank space agar dimasukkan ke elemen array satu baris</li> <li>- Gimana cara baca command karena command ada spasinya → dibaca sampai ketemu karakter newline \n, supaya karakter spasi/blank terbaca</li> <li>- Boleh membuat implementasi ADT Point sendiri misalnya jika ingin digunakan di spek bonus (mau membuat game catur)</li> <li>- Kan console harus start dulu baru bisa load file, setelah itu bisa create game, play game, maupun liat list game. Nah selanjutnya misal user meng-input (ngomong apa tadi tops) → bikinantisipasi kalau ada input yang gak valid</li> <li>- Kalau ada command lain diabaikan aja, tidak keluar dari program</li> <li>- Naufal: pakai STARTWORD untuk nerima input dari user, tetapi pas mau nerima input lagi di kode yang sama gak bisa; apakah pakai fungsi start word berkali?; jadi kemarin pakai driver dari create game, tapi manggil create gamenya 2x. nah waktu pemanggilan pertama itu berhasil untuk mau input, tapi di pemanggilan kedua gak bisa ambil input langsung error dan keluar → mungkin masalahnya karena pakai start word berkali2, harusnya STARTWORD dipakai 1 kali saja sisanya ADV</li> </ul>
<b>Tempat : Daring</b>	
<b>Kehadiran Anggota Kelompok:</b>	
1. Muhammad Naufal/18221089 Tanda tangan 	
2. Christopher Febrian Nugraha/18221115 Tanda tangan 	
3. Kevin Prayoga Abdinegara/18221141 Tanda tangan 	
4. Natasya Vercelly Harijadi/18221119 Tanda Tangan 	

<p>5. Ibnu Khairy Algifari/18221091</p> <p>Tanda tangan</p> 	
	<p><b>Tanda Tangan Asisten:</b></p> 

## Asistensi II

<b>Tanggal : 10 November 2022</b>	<b>Catatan Asistensi:</b>
<b>Tempat : Daring</b>	
<p><b>Kehadiran Anggota Kelompok:</b></p> <p>1. Muhammad Naufal/18221089 Tanda tangan</p>  <p>2. Christopher Febrian Nugraha/18221115 Tanda tangan</p>  <p>3. Kevin Prayoga Abidinegara/18221141 Tanda tangan</p>  <p>4. Natasya Vercelly Harijadi/18221119 Tanda Tangan</p>  <p>5. Ibnu Khairy Algifari/18221091 Tanda tangan</p> 	<ul style="list-style-type: none"> <li>- Log activity berdasarkan apa ya? <ul style="list-style-type: none"> <li>- Bukan berdasarkan commit, tapi berdasarkan apa yang kamu kerjakan di hari itu</li> </ul> </li> <li>- Di menu awal apa aja ya? Apakah hanya ada start, load, atau bisa ada quit dan help? <ul style="list-style-type: none"> <li>- Kalau dilihat dari spek ya start sama load aja, tapi kalau mau masukin quit gapapa</li> </ul> </li> <li>- Console.c itu semua fungsi yang dibuat atau gimana? <ul style="list-style-type: none"> <li>- Ga harus persis kayak gitu, yang penting strukturnya rapi gak di 1 folder semua</li> </ul> </li> <li>- Makefile itu apa ya? <ul style="list-style-type: none"> <li>- Makefile itu supaya nanti gak usah repot copy paste command untuk compile, tinggal make aja</li> </ul> </li> <li>- Makefile wajib atau kalau bikin akan dapat nilai tambahan gitu ya? <ul style="list-style-type: none"> <li>- Gak wajib</li> </ul> </li> <li>- Untuk test case nanti bakal dari kakak atau bagaimana ya? <ul style="list-style-type: none"> <li>- Nanti ada test case standard, jadi semua asisten sama gitu</li> </ul> </li> <li>- Untuk algoritma menarik itu wajib gak ya? <ul style="list-style-type: none"> <li>- Gapapa kalau gak nemu yang menarik, buat bonus aja</li> </ul> </li> </ul>



	<p><b>Tanda Tangan Asisten:</b></p> 
--	------------------------------------------------------------------------------------------------------------------------