

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

**BNMO**


Dipersiapkan oleh:

Kelompok 5:

Richard Haris	18221006
Harits Afiq Nugroho	18221012
Justin Yusuf Abidjoko	18221016
Ahmad Rivai Yahya	18221017
Anjani Ibrahim	18221031

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		IF2111-TB1-05		35
		Revisi		11 November 2022

# Daftar Isi

<b>1 Ringkasan</b>	<b>2</b>
<b>2 Penjelasan Tambahan Spesifikasi Tugas</b>	<b>4</b>
2.1 Spesifikasi Fitur Tambahan 1	4
<b>3 Struktur Data (ADT)</b>	<b>4</b>
3.1 ADT Array Dinamis	4
3.2 ADT Mesin Karakter	4
3.3 ADT Mesin Kata	4
3.4 ADT Queue	5
<b>4 Program Utama</b>	<b>6</b>
<b>5 Data Test</b>	<b>6</b>
5.1 Data Test 1	6
5.2 Data Test 2	7
5.3 Data Test 3	7
5.4 Data Test 4	8
5.5 Data Test 5	8
5.6 Data Test 6	9
5.7 Data Test 7	9
5.8 Data Test 8	10
5.9 Data Test 9	10
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	15
<b>6 Test Script</b>	<b>17</b>
<b>7 Pembagian Kerja dalam Kelompok</b>	<b>19</b>
<b>8 Lampiran</b>	<b>21</b>
8.1 Deskripsi Tugas Besar 1	21
8.2 Notulen Rapat	32
8.3 Log Activity Anggota Kelompok	35

# 1 Ringkasan

Pada tugas besar kali ini, kami diharuskan untuk memprogram ulang robot *video game console* milik Indra dan Doni. Indra dan Doni memiliki sebuah robot *video game console* yang bernama **BNMO** (dibaca Binomo). Robot *video game console* tersebut sempat mengalami kerusakan sekitar dua bulan yang lalu, namun sudah berhasil diperbaiki. Alat tersebut dapat kembali beroperasi, tetapi terdapat banyak bug yang berada di sistemnya. Di dalam BNMO, terdapat beberapa fitur utama, diantaranya yaitu memainkan *game*, menambahkan *game*, menghapus *game*, dan mengurutkan *game* yang akan dimainkan.

Program utama dari BNMO yang telah diprogram ulang akan melakukan suatu perintah yang dimana perintah tersebut didapatkan dari pengguna. Setelah pengguna memulai program utama, terdapat pilihan untuk melanjutkan permainan sebelumnya atau mulai dari awal. Lalu terdapat command lainnya yaitu: *save*, *create game*, *list game*, *delete game*, *queue game*, *play game*, *skipgame*, *quit*, *help*, dan *command* lain yang dimana jika pengguna memasukkan salah satu dari command tersebut, maka BNMO akan berjalan sesuai *command* yang dimasukkan.

Pada kasus BNMO kali ini, terdapat beberapa ADT yang digunakan yaitu:

1. ADT Array Dinamis
2. ADT Mesin Karakter dan Mesin Kata
3. ADT Queue

Penjelasan mengenai ADT yang digunakan di atas terdapat pada BAB 3 Struktur Data (ADT).

Laporan ini berisikan mengenai penjelasan dari program-program yang kami buat untuk memprogram ulang robot *video game console* milik Indra dan Doni. Bagian pertama dari laporan ini berisi tentang ringkasan dari laporan ini beserta gambaran umum mengenai program yang dibuat. Bagian dua menjelaskan mengenai tambahan spesifikasi yang dimana tambahan spesifikasi ini berisi penjelasan fitur yang belum rinci dari Deskripsi Tugas Besar. Bagian tiga menjelaskan tentang struktur data (ADT) yang digunakan pada program ini. Bagian empat menjelaskan program utama dimana bagian ini menjelaskan mengenai algoritma program utama. Bagian lima berisikan algoritma-algoritma menarik yang kami temukan pada pengerjaan tugas besar dua nantinya. Bagian enam menjelaskan tentang pengujian-pengujian data. Bagian tujuh berisikan test script. Bagian delapan menjelaskan tentang deskripsi kerja dalam kelompok. Bagian terakhir berisikan lampiran.

Akhirnya robot *video game console* BNMO milik Indra dan Doni dapat berjalan dengan lancar sesuai dengan spesifikasi. Seiring dengan kembali berjalannya BNMO dengan lancar, kelompok kami semakin mengerti mengenai penggunaan-penggunaan ADT yang aplikatif. Selain itu kelompok kami juga bekerja sama dengan baik.

## 2 Penjelasan Tambahan Spesifikasi Tugas

### 2.1 Spesifikasi Fitur Tambahan 1

Fitur tambahan yang ada pada tugas ini adalah fitur untuk *quit* (keluar) dari game saat *game* belum selesai (di tengah permainan). Hal ini dilakukan dengan memasukkan input *command* “0” pada saat memainkan *game* RNG dan memasukkan input *command* “QUIT” pada saat memainkan *game* DINER DASH.

## 3 Struktur Data (ADT)

### 3.1 ADT Array Dinamis

- Sketsa struktur data : struktur *array* dinamis terdiri atas *array* yang terdiri dari *Neff* yang bertipe *integer*, *Capacity* yang bertipe *integer*, dan *ElType A* yang bertipe *char*.
- Persoalan yang diselesaikan : Representasi daftar game yang terdapat dalam sistem.
- Alasan pemilihan : informasi seluruh *game* dapat disimpan dalam *array of game* ini dan setiap aksi pengurangan/penambahan game dapat terselesaikan.
- Diimplementasikan sebagai ADT *array* dengan nama *file header* “arraydin.h”

### 3.2 ADT Mesin Karakter

- Sketsa struktur data : terdapat sebuah *variable extern* bertipe *char* dengan nama *currentChar* dan *boolean* dengan nama *EOP*. Adapun Prototype yang ada dalam ADT ini yaitu *START()* dan *ADV ()*.
- Persoalan yang diselesaikan : Melakukan *parsing command program* dan pembacaan file konfigurasi ke dalam aplikasi.
- Alasan pemilihan : Mesin kata membutuhkan mesin karakter dalam fungsi-fungsi *prototype*-nya.
- Diimplementasikan sebagai ADT mesin karakter dengan nama *file header* “mesinkarakterv2.h”

### 3.3 ADT Mesin Kata

- Sketsa struktur data : struktur mesin kata terdiri atas *Word* yang terdiri dari *Length* yang bertipe *integer*, dan *TabWord[NMax]* yang bertipe *char*. Adapun *NMax* bertipe *integer* dengan *defined value* 50.
- Persoalan yang diselesaikan : Melakukan *parsing command program* dan pembacaan file konfigurasi ke dalam aplikasi.
- Alasan pemilihan : Membacakan informasi dari *file external* memerlukan akuisisi kata satu per satu dan menerjemahkannya ke dalam bentuk data *game*. Pembacaan input *command* dari *user* dan *state* dari *game* yang disimpan juga berupa string sehingga perlu digunakan mesin kata.
- Diimplementasikan sebagai ADT mesin kata dengan nama *file header* “mesinkata.h”

### 3.4 ADT Queue

- Sketsa struktur data : struktur *queue* terdiri atas *Queue* yang terdiri dari *idxTail* yang bertipe *integer*; *idxHead* yang bertipe *integer*; dan *ElType* buffer [*CAPACITY*] yang bertipe *char*.
- Persoalan yang diselesaikan : Menentukan urutan bermain game.
- Alasan pemilihan : *Game* yang akan dimainkan oleh *user* (pemain) direpresentasikan sebagai *queue* yang dimainkan secara berurutan (*first in first out* / *FIFO*) .
- Diimplementasikan sebagai ADT *queue* dengan nama *file header* “*queue.h*”, “*queue\_of\_string.h*”

### 3.5 ADT ProcessedOrder

- Sketsa struktur data : struktur *processedorder* terdiri atas *Capacity* yang bertipe *integer*, *Neff* yang bertipe *integer*, serta *A* yang bertipe *pointer to* masakan. Masakan terdiri atas makanan yang bertipe *pointer to char* serta *sisa\_durasi\_or\_ketahanan\_masakan* yang bertipe *integer*.
- Persoalan yang diselesaikan : Menyimpan daftar masakan yang sedang dimasak dan daftar masakan yang siap untuk disajikan pada *game Diner Dash*.
- Alasan pemilihan : informasi seluruh masakan dapat disimpan dalam *processedorder* ini dan setiap aksi pengurangan/penambahan masakan dapat terselesaikan di indeks manapun.
- Diimplementasikan sebagai ADT dengan nama *file header* “*processedorder.h*”

### 3.6 ADT Circular Queue of Pesanan

- Sketsa struktur data : struktur *circular queue of pesanan* terdiri atas *Tab* yang bertipe *pointer to* pesanan, *HEAD* yang bertipe *IdxType*, *TAIL* yang bertipe *IdxType*, serta *MaxEl* yang bertipe *integer*. Pesanan terdiri atas makanan yang bertipe *pointer to char*, *durasimasak* yang bertipe *integer*, *ketahanan* yang bertipe *integer*, serta *harga* yang bertipe *integer*. *IdxType* bertipe *integer*.
- Persoalan yang diselesaikan : Menyimpan daftar pesanan pada *game Diner Dash*
- Alasan pemilihan : informasi seluruh pesanan dapat disimpan dalam *circular\_queue\_of\_pesanan* ini dan setiap aksi pengurangan/penambahan pesanan dapat terselesaikan di indeks *HEAD* atau *TAIL*.
- Diimplementasikan sebagai ADT dengan nama *file header* “*circular\_queue\_of\_pesanan.h*”

## 4 Program Utama

Program utama dengan nama file `main.c` akan meng-include file `“console.c”` yang meng-include file header `“console.h”`. File `“console.h”` meng-include file header `“console.h”` dan semua file header dari ADT yang digunakan. Program utama dimulai dengan memanggil fungsi `start` atau `load`. Kemudian terdapat command `create game` untuk menambahkan game yang dapat ditampilkan dengan fungsi `list`. Ada juga command `delete game` untuk menghapus game dari `list`, namun game yang dapat dihapus hanya game dengan nomor urutan di atas lima dan game yang tidak ada di queue. Untuk memainkan game, user perlu menambahkan antrean game dengan menggunakan command `queue game` dan dengan menggunakan command `play game`, maka game pertama yang ada di `queue game` akan dimainkan. List dalam `queue game` akan hilang dengan menggunakan command `quit` yang juga berarti keluar dari game. Untuk melewati permainan dapat menggunakan command `skip game`. Ada juga command `help` yang akan menampilkan penjelasan dari command yang ada dan command `save` untuk menyimpan state game pemain saat ini ke dalam suatu file. Jika input dari user merupakan command yang tidak dikenali, maka user akan diminta memasukkan input yang valid lagi.

Secara default, terdapat dua game yang dapat dimainkan yaitu RNG dan Diner Dash. RNG merupakan game menebak angka sampai angka yang ditebak user sama dengan angka yang ditentukan program. Diner Dash merupakan game dengan konsep restoran dengan pengantaran makanan berdasarkan prioritasnya.

## 5 Data Test

### 5.1 Data Test 1

Test ini dilakukan untuk memastikan bahwa program sudah dapat berjalan dan menampilkan tampilan judul game serta main menu untuk memulai atau keluar dari game.

```
PERINGATAN!!!
=====
PROGRAM AKAN ERROR ATAU BERHENTI KETIKA MENJALANKAN COMMAND YANG MEMILIKI KARAKTER '.'
=====

Selamat datang pada BNMO!

Jalankan command START atau LOAD <filename tanpa .txt> untuk membuka file.
Jalankan command QUIT untuk keluar dari program.

ENTER COMMAND: _
```

Gambar 5.1.1. Tampilan Judul dan Menu Awal.

```
File config berhasil dibaca. BNMO berhasil dijalankan.

ENTER COMMAND:
```

Gambar 5.1.2. Tampilan setelah command “START” dimasukkan.

## 5.2 Data Test 2

Tes ini dimaksudkan untuk menguji fitur LOAD dari *game* yang sudah pernah disimpan sebelumnya.

```
Selamat datang pada BNMO!  
  
Jalankan command START atau LOAD <filename tanpa .txt> untuk membuka file.  
Jalankan command QUIT untuk keluar dari program.  
  
ENTER COMMAND: LOAD cobasave
```

**Gambar 5.2.1.** Tampilan saat memasukkan *command* “LOAD” suatu file.

```
File cobasave berhasil dibaca. BNMO berhasil dijalankan.  
  
ENTER COMMAND:
```

**Gambar 5.2.2.** Tampilan saat *command* “LOAD” berhasil.

## 5.3 Data Test 3

Tes ini digunakan untuk menguji *command* SAVE.

```
Save file berhasil disimpan.  
  
ENTER COMMAND:
```

**Gambar 5.3.1.** Tampilan saat *command* “SAVE GAME” berhasil.

## 5.4 Data Test 4

Tes ini digunakan untuk menguji *command* CREATE GAME yang berfungsi untuk menambahkan game baru berdasarkan input dari *keyboard*. Nama *game* yang dimasukkan pada contoh adalah “Petak Umpet”.

```
Berikut adalah daftar game yang tersedia:  
1. RNG  
2. Diner Dash  
3. Dinosaur in Earth  
4. Risewoman  
5. Eifel Tower  
  
Masukkan nama game yang akan ditambahkan:
```

**Gambar 5.4.1** Tampilan saat *command* “CREATE GAME” dimasukkan.

```
Berikut adalah daftar game yang tersedia:  
1. RNG  
2. Diner Dash  
3. Dinosaur in Earth  
4. Risewoman  
5. Eifel Tower  
6. Petak Umpet
```

**Gambar 5.4.2** Tampilan saat *game* baru berhasil ditambahkan.

## 5.5 Data Test 5

Tes ini dilakukan untuk menguji *command* untuk menampilkan daftar game yang ada.

```
Berikut adalah daftar game yang tersedia:  
1. RNG  
2. Diner Dash  
3. Dinosaur in Earth  
4. Risewoman  
5. Eifel Tower  
  
ENTER COMMAND:
```

**Gambar 5.5.1** Tampilan saat *command* “LIST GAME” dimasukkan.



## 5.6 Data Test 6

Tes ini dilakukan untuk menguji penghapusan game yang ada di list. *Game* yang ada pada urutan lima pertama dan yang ada pada *queuegame* tidak dapat dihapus. Jika input *game* yang akan dihapus merupakan urutan lima pertama, maka *game* akan gagal dihapus.

```
Berikut adalah daftar game yang tersedia:
1. RNG
2. Diner Dash
3. Dinosaur in Earth
4. Risewoman
5. Eifel Tower

Masukkan nomor game yang akan dihapus: 1
Game gagal dihapus.

ENTER COMMAND:
```

**Gambar 5.6.1** Tampilan jika input user mencoba menghapus lima urutan pertama pada list yang berarti *game* gagal dihapus.

```
Berikut adalah daftar game yang tersedia:
1. RNG
2. Diner Dash
3. Dinosaur in Earth
4. Risewoman
5. Eifel Tower
6. Mario Kart

Masukkan nomor game yang akan dihapus: 6
Game berhasil dihapus.

ENTER COMMAND: _
```

**Gambar 5.6.2** Tampilan jika *game* berhasil dihapus.

## 5.7 Data Test 7

Tes ini dilakukan untuk menguji penambahan daftar atau antrean *game* yang akan dimainkan.

```
Berikut adalah daftar antrean game-mu:

Berikut adalah daftar game yang tersedia:
1. RNG
2. Diner Dash
3. Dinosaur in Earth
4. Risewoman
5. Eifel Tower

Nomor game yang mau ditambahkan ke antrean: 3
Game berhasil ditambahkan ke dalam daftar antrean.

ENTER COMMAND: _
```

**Gambar 5.7.1** Tampilan jika *game* berhasil ditambahkan ke antrean.

## 5.8 Data Test 8

Tes ini dilakukan untuk mengecek program yang memulai *game* yang berada pada antrean *game* (*queue game*). Jika tidak ada antrean pada *queue game*, maka program akan meminta untuk memasukkan antrean terlebih dahulu.

```
Belum ada antrean game yang kamu miliki.  
ENTER COMMAND:
```

**Gambar 5.8.1.** Tampilan saat belum ada antrean *game* pada QUEUE GAME.

Sedangkan jika sudah terdapat antrean pada *queue game*, maka program akan memulai dari antrean pertama.

```
Loading RNG ...  
RNG Telah dimulai. Uji keberuntungan Anda dengan menebak X.  
Tebakan: _
```

**Gambar 5.8.2.** Tampilan saat *game* berhasil dimulai.

## 5.9 Data Test 9

Tes ini dilakukan untuk mengecek program yang melewati beberapa antrean pada *queue game*. Jika melewati permainan menuju ke permainan dengan nama *game* selain yang ada pada spesifikasi tugas maka akan menampilkan *game* sedang dalam proses *maintenance*.

```
Berikut adalah daftar game-mu:  
1. Diner Dash  
2. Eiffel Tower  
3. Risewoman  
  
Game Risewoman masih dalam maintenance, belum dapat dimainkan. Silahkan pilih game lain.  
ENTER COMMAND: _
```

**Gambar 5.9.1.** Tampilan saat fitur SKIPGAME melewati antrean *game* yang selain berada pada spesifikasi.

Jika melewati permainan menuju ke permainan dengan nomor urutan lebih dari lima pada *list game* maka akan menampilkan skor acak.

```

Berikut adalah daftar game-mu:
1. RNG
2. Diner Dash
3. Dinosaur in Earth
4. Rivai Bersholawat
Loading Rivai Bersholawat ...
Skor = 16

ENTER COMMAND: █

```

**Gambar 5.9.2.** Tampilan saat fitur SKIPGAME melewati antrian *game* yang dihasilkan dari fitur CREATE GAME.

Jika melewati permainan menuju ke permainan dengan nama *game* yang berada pada spesifikasi, maka akan memainkan permainan tersebut.

```

Berikut adalah daftar game-mu:
1. Dinosaur in Earth
2. Risewoman
3. RNG
Loading RNG ...
RNG Telah dimulai. Uji keberuntungan Anda dengan menebak X.
Tebakan: █

```

**Gambar 5.9.3.** Tampilan saat fitur SKIPGAME melewati antrian menuju *game* yang terdapat pada spesifikasi.

Jika melakukan fitur SKIPGAME tetapi melewati batas *n* pada antrian, akan mengeluarkan keluaran berupa informasi tidak ada permainan lagi dalam daftar *game-mu*.

```

Berikut adalah daftar game-mu:
1. RNG

Tidak ada permainan lagi dalam daftar game-mu.

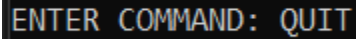
ENTER COMMAND:

```

**Gambar 5.9.4.** Tampilan saat fitur SKIPGAME melewati batas antrian.

## 5.10 Data Test 10

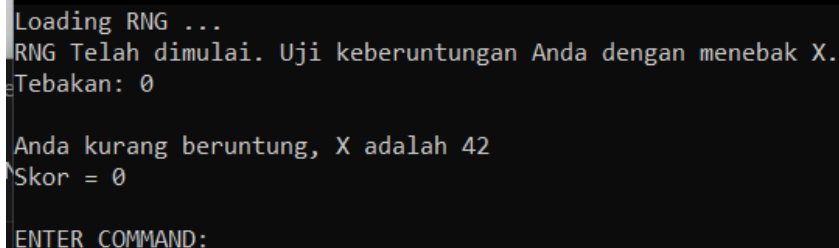
Tes ini dilakukan untuk mengecek program untuk keluar dari program.



```
ENTER COMMAND: QUIT
```

**Gambar 5.10.1.** Tampilan saat input *command* QUIT.

Ada pun untuk tampilan setelah *command* QUIT berhasil dilakukan tidak dapat ditampilkan karena saat *command* berhasil, otomatis keluar dari *game*.



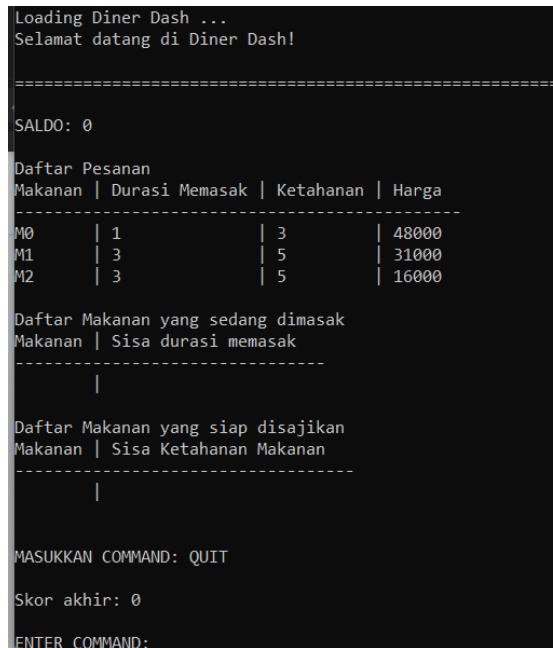
```
Loading RNG ...
RNG Telah dimulai. Uji keberuntungan Anda dengan menebak X.
Tebakan: 0

Anda kurang beruntung, X adalah 42
Skor = 0

ENTER COMMAND:
```

**Gambar 5.10.2** Tampilan saat input *command* "quit" pada permainan RNG.

Adapun cara untuk keluar dari permainan RNG yaitu dengan cara memasukkan input *command* "0".



```
Loading Diner Dash ...
Selamat datang di Diner Dash!

=====
SALDO: 0

Daftar Pesanan
Makanan | Durasi Memasak | Ketahanan | Harga
-----
M0      | 1                | 3          | 48000
M1      | 3                | 5          | 31000
M2      | 3                | 5          | 16000

Daftar Makanan yang sedang dimasak
Makanan | Sisa durasi memasak
-----
|

Daftar Makanan yang siap disajikan
Makanan | Sisa Ketahanan Makanan
-----
|

MASUKKAN COMMAND: QUIT

Skor akhir: 0

ENTER COMMAND:
```

**Gambar 5.10.3** Tampilan saat input *command* "quit" pada permainan DINER DASH.

Adapun cara untuk keluar dari permainan DINER DASH yaitu dengan cara memasukkan input *command* "QUIT".

### 5.11 Data Test 11

Tes ini dilakukan untuk mengecek program untuk menampilkan fitur bantuan.

```
DAFTAR COMMANDS:
1. START - Menjalankan file konfigurasi default
2. LOAD <filename tanpa .txt> - Menjalankan file dari input user
3. QUIT - Keluar dari program
4. HELP - Menampilkan daftar command yang dapat dijalankan

ENTER COMMAND:
```

**Gambar 5.11.1.** Tampilan untuk daftar *commands* yang tersedia sebelum menjalankan command START atau LOAD.

```
DAFTAR COMMANDS:
1. SAVE <filename tanpa .txt> - Menyimpan state sistem ke dalam file dari input user
2. CREATE GAME - Membuat game baru dari input user
3. LIST GAME - Menampilkan daftar game yang dapat tersedia dalam sistem
4. DELETE GAME - Menghapus game yang tersedia dari sistem
5. QUEUE GAME - Menambahkan game ke dalam antrean game
6. PLAY GAME - Memainkan game yang berada di depan antrean
7. SKIP GAME <n> - Melewati game yang berada di dalam antrean sebanyak n kali
8. QUIT - Keluar dari program
9. HELP - Menampilkan daftar command yang dapat dijalankan

ENTER COMMAND:
```

**Gambar 5.11.2.** Tampilan untuk daftar *commands* yang tersedia setelah menjalankan command START atau LOAD..

### 5.12 Data Test 12

Tes ini dilakukan untuk mengecek program jika terdapat *command* yang tidak sesuai dengan spesifikasi tugas. Jika *command* tidak sesuai, maka akan diminta untuk memasukkan *command* yang lain (diharapkan sesuai dengan spesifikasi).

```
Command tidak dikenali, silahkan masukkan command yang valid.

ENTER COMMAND: _
```

**Gambar 5.12.1.** Tampilan saat *command* yang dimasukkan tidak sesuai atau tidak tersedia.

### 5.13 Data Test 13

Tes ini dilakukan untuk uji coba program *game* RNG. Jika tebakan lebih besar dari angka yang sudah ditetapkan program maka program akan meminta user untuk menebak lagi dengan angka yang lebih kecil dan jika tebakan lebih kecil dari angka yang sudah ditetapkan program akan meminta user untuk menebak lagi dengan angka yang lebih kecil. Jika user berhasil menebak angkanya maka *game* akan selesai dan akan ada skor yang muncul.

```
Loading RNG ...
RNG Telah dimulai. Uji keberuntungan Anda dengan menebak X.
Masukkan 0 jika ingin menyerah.
Tebakan: 25
Lebih besar
Tebakan: 50
Lebih besar
Tebakan: 70
Lebih kecil
Tebakan: 0

Anda kurang beruntung, X adalah 62
Skor = 0

ENTER COMMAND: _
```

**Gambar 5.13.1** Tampilan RNG berhasil dijalankan.

```
Tebakan: 25
Lebih besar
Tebakan: 50
Lebih besar
```

**Gambar 5.13.2** Tampilan jika *user* menebak angka yang lebih kecil.

```
Tebakan: 90
Lebih kecil
Tebakan: 85
Lebih kecil
Tebakan: 83
Lebih besar
Tebakan: 84
```

**Gambar 5.13.3** Tampilan jika *user* menebak angka yang lebih besar.

```

Loading RNG ...
RNG Telah dimulai. Uji keberuntungan Anda dengan menebak X.
Masukkan 0 jika ingin menyerah.
Tebakan: 50
Lebih kecil
Tebakan: 10
Lebih besar
Tebakan: 25
Lebih kecil
Tebakan: 20
Lebih kecil
Tebakan: 15
Lebih besar
Tebakan: 17
Lebih kecil
Tebakan: 16

Ya, X adalah 16
Skor = 76

```

**Gambar 5.13.4** Tampilan jika *user* menebak angka dengan benar dan *game* selesai.

## 5.14 Data Test 14

Tes ini dilakukan untuk mencoba program *game* DINER DASH.

```

Loading Diner Dash ...
Selamat datang di Diner Dash!

=====

SALDO: 0

Daftar Pesanan
Makanan | Durasi Memasak | Ketahanan | Harga
-----|-----|-----|-----
M0      | 2              | 5         | 44000
M1      | 3              | 5         | 50000
M2      | 1              | 2         | 40000

Daftar Makanan yang sedang dimasak
Makanan | Sisa durasi memasak
-----|-----
|

Daftar Makanan yang siap disajikan
Makanan | Sisa Ketahanan Makanan
-----|-----
|

```

**Gambar 5.14.1.** Tampilan menu awal *game* DINER DASH

```

MASUKKAN COMMAND: COOK M0

Berhasil memasak M0
=====

SALDO: 0

Daftar Pesanan
Makanan | Durasi Memasak | Ketahanan | Harga
-----
M0      | 2              | 5         | 44000
M1      | 3              | 5         | 50000
M2      | 1              | 2         | 40000
M3      | 4              | 3         | 18000

Daftar Makanan yang sedang dimasak
Makanan | Sisa durasi memasak
-----
M0      | 2

Daftar Makanan yang siap disajikan
Makanan | Sisa Ketahanan Makanan
-----
|

```

**Gambar 5.14.2.** Tampilan pada saat input *command* "COOK"

Disaat masakan sudah disajikan, maka otomatis saldo akan bertambah dengan harga dari masakan yang disajikan. Dalam contoh ini M0 sudah disajikan dan saldo bertambah 44000 yang merupakan harga dari M0.

```

MASUKKAN COMMAND: SERVE M0

Berhasil mengantar M0
=====

SALDO: 44000

Daftar Pesanan
Makanan | Durasi Memasak | Ketahanan | Harga
-----
M1      | 3              | 5         | 50000
M2      | 1              | 2         | 40000
M3      | 4              | 3         | 18000
M4      | 2              | 3         | 15000
M5      | 1              | 4         | 30000
M6      | 2              | 2         | 45000

Daftar Makanan yang sedang dimasak
Makanan | Sisa durasi memasak
-----
M3      | 3

Daftar Makanan yang siap disajikan
Makanan | Sisa Ketahanan Makanan
-----
M2      | 1

```

**Gambar 5.14.3.** Tampilan pada saat input *command* "SERVE"



## 6 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1.	Fitur START GAME	Memeriksa apakah permainan dapat dijalankan	Memasukkan <i>command</i> START pada menu awal	Data Test 1	Permainan berhasil dimulai lalu menampilkan <i>opening message</i> bahwa berhasil masuk	Sesuai yang diharapkan
2.	Fitur LOAD	Memeriksa apakah fitur load berhasil dijalankan	Memasukkan <i>command</i> LOAD (nama file) pada menu utama untuk file yang sudah di <i>save</i>	Data Test 2	<i>Game</i> berhasil dilanjutkan sesuai dengan kondisi saat <i>save game</i>	Sesuai yang diharapkan
3.	Fitur SAVE	Memeriksa apakah fitur save berhasil dijalankan	Memasukkan <i>command</i> SAVE (nama file) untuk menyimpan kondisi <i>game</i>	Data Test 3	<i>Game</i> berhasil disimpan dan bisa di <i>load</i> dengan <i>command</i> LOAD	Sesuai yang diharapkan
4.	Fitur CREATE GAME	Memeriksa apakah fitur CREATE GAME berhasil dijalankan	Memasukkan input <i>command</i> CREATE GAME. Kemudian input nama <i>game</i> yang ingin ditambah	Data Test 4	<i>Game</i> baru berhasil terdaftar pada <i>list game</i>	Sesuai yang diharapkan
5.	Fitur LIST GAME	Memeriksa apakah <i>command</i> dapat menampilkan list game yang ada	Memasukkan <i>command</i> LIST GAME	Data Test 5	Mengeluarkan list game yang ada	Sesuai yang diharapkan
6.	Fitur DELETE GAME	Memeriksa apakah program <i>command</i> dapat menghapus game yang dipilih user	Memasukkan input <i>command</i> DELETE GAME. Kemudian input urutan <i>game</i> yang ingin dihapus	Data Test 6	Penghapusan game dari list dapat dilakukan	Sesuai yang diharapkan
7.	Fitur QUEUE GAME	Memeriksa apakah program <i>command</i> dapat menambahkan antrian <i>game</i> yang akan dimainkan.	Memasukkan input <i>command</i> QUEUE GAME. Kemudian input urutan game yang ingin dimasukkan antrian	Data Test 7	Penambahan game ke antrian yang akan dimainkan dapat dilakukan	Sesuai yang diharapkan
8.	Fitur PLAY GAME	Memeriksa apakah program <i>command</i> dapat memainkan <i>game</i>	Memasukkan input <i>command</i> "PLAY GAME" untuk memulai <i>game</i> pada antrian pertama	Data Test 8	<i>Game</i> pada antrian pertama dapat dimainkan	Sesuai yang diharapkan

9.	Fitur SKIPGAME	Memeriksa apakah program <i>command</i> dapat melewati antrean sejumlah n	Memasukkan input <i>command</i> "SKIPGAME" untuk melewati antrean game sebanyak n	Data Test 9	Antrean permainan dapat dilewati sejumlah n dan kemudian memainkan permainan yang berada di spesifikasi. Jika permainan tidak terdapat pada spesifikasi akan memunculkan pesan <i>maintenance</i> . Jika permainan yang merupakan hasil dari fitur "CREATE GAME" akan memunculkan skor acak	Sesuai yang diharapkan
10.	Fitur QUIT	Memeriksa apakah program <i>command</i> dapat keluar dari program	Memasukkan input <i>command</i> "QUIT" untuk keluar dari program	Data Test 10	Jika memasukkan input <i>command</i> "quit" pada program utama, maka akan keluar dari program. Jika memasukkannya pada permainan RNG maka akan keluar dari permainan tersebut. Begitu pula dengan permainan DINER DASH.	Sesuai yang diharapkan
11.	Fitur HELP	Memeriksa apakah program <i>command</i> dapat menampilkan fitur <i>help</i>	Memasukkan input <i>command</i> "HELP" jika mendapat suatu kesusahan dalam menjalankan program	Data Test 11	Mengeluarkan daftar-daftar mengenai fitur <i>help</i>	Sesuai yang diharapkan
12.	Fitur COMMAND LAIN	Memeriksa apakah program <i>command</i> dapat membedakan antara <i>command</i> yang sesuai dengan spesifikasi dan tidak	Memasukkan input <i>command</i> dan memeriksanya	Data Test 12	Jika input <i>command</i> tidak sesuai dengan spesifikasi, program akan meminta input <i>command</i> ulang	Sesuai yang diharapkan
13.	Fitur Game RNG	Memeriksa apakah <i>game</i>	Memasukkan input <i>command</i> PLAY GAME setelah menambahkan	Data Test 13	Game berhasil dijalankan sesuai	Sesuai yang diharapkan

		RNG berjalan dengan baik	game RNG pada antrean (QUEUE GAME)		dengan ketentuan game	
14.	Fitur <i>Game</i> DINER DASH	Memeriksa apakah <i>game</i> DINER DASH berjalan dengan baik	Memasukkan input <i>command</i> sesuai dengan apa yang tertera pada spesifikasi	Data <i>Test</i> 14	Game berhasil dijalankan sesuai dengan spesifikasi	Sesuai yang diharapkan

## 7 Pembagian Kerja dalam Kelompok

Nama Anggota Kelompok	Pembagian Kerja
Richard Haris	<ul style="list-style-type: none"> <li>Membuat fungsi: <ol style="list-style-type: none"> <li>QUEUEGAME</li> <li>DINERDASH</li> <li>Beberapa fungsi tambahan pendukung: random_number, food_id_generator, stringlen, getCurrentWord</li> </ol> </li> <li>Memodifikasi ADT serta memeriksa dengan membuat driver ADT</li> <li>Membantu pengerjaan penjelasan ADT khusus diner dash di laporan</li> </ul>
Harits Afiq Nugroho	<ul style="list-style-type: none"> <li>Membuat fungsi: <ol style="list-style-type: none"> <li>PLAYGAME</li> <li>SKIPGAME</li> <li>GAMEBUATAN</li> </ol> </li> <li>Mengerjakan laporan bagian Ringkasan, beberapa Data Test, dan Test Script.</li> </ul>
Justin Yusuf Abidjoko	<ul style="list-style-type: none"> <li>Membuat fungsi: <ol style="list-style-type: none"> <li>START</li> <li>LOAD</li> <li>SAVE</li> <li>Beberapa fungsi tambahan yang tidak dicantumkan dalam spesifikasi tugas (compareString, concat, wordToString, wordToInteger, inputString, inputInteger)</li> </ol> </li> <li>Memberi arahan pembagian tugas serta cara pengerjaan tugas.</li> </ul>

	<ul style="list-style-type: none"> <li>• Membuat <i>repository GitHub</i></li> <li>• Memodifikasi ADT serta memeriksa dengan membuat driver ADT</li> <li>• Menggabungkan <i>code</i> fungsi dari anggota kelompok.</li> <li>• Membantu mengisi laporan bagian Data Test</li> </ul>
Ahmad Rivai Yahya	<ul style="list-style-type: none"> <li>• Membuat fungsi: <ol style="list-style-type: none"> <li>1. CREATEGAME</li> <li>2. LISTGAME</li> <li>3. HELP</li> </ol> </li> <li>• Mengerjakan laporan sedikit bagian program utama, Spesifikasi fitur tambahan, beberapa Data Test, dan Test Script.</li> </ul>
Anjani Ibrahim	<ul style="list-style-type: none"> <li>• Membuat fungsi: <ol style="list-style-type: none"> <li>1. DELETEDGAME</li> <li>2. QUIT</li> <li>3. RNG</li> </ol> </li> <li>• Mengerjakan laporan bagian program utama, beberapa Data Test, dan Test Script.</li> </ul>

## 8 Lampiran

### 8.1 Deskripsi Tugas Besar 1

#### 1. 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.

ENTER COMMAND: **START**

File konfigurasi sistem berhasil dibaca. BNMO berhasil dijalankan.

##### 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](#).

ENTER COMMAND: **LOAD savefile1.txt**

Save file berhasil dibaca. BNMO berhasil dijalankan.

##### 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.

ENTER COMMAND: **SAVE savefile1.txt**

Save file berhasil disimpan.

**d. CREATEGAME**

CREATEGAME merupakan command yang digunakan untuk menambahkan game baru pada daftar game. Spesifikasi game yang dibuat dapat dilihat pada section Spesifikasi Game

ENTER COMMAND: **CREATE GAME**

Masukkan nama game yang akan ditambahkan: **EXTRA1**

Game berhasil ditambahkan

**e. LISTGAME**

LISTGAME merupakan command yang digunakan untuk menampilkan daftar game yang disediakan oleh sistem.

ENTER COMMAND: **LIST GAME**

Berikut adalah daftar game yang tersedia

1. RNG
2. LUNCH SLOW
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER

**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.

ENTER COMMAND: **DELETE GAME**

Berikut adalah daftar game yang tersedia

1. RNG
2. LUNCH SLOW
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER
6. CUSTOM GAME 1

Masukkan nomor game yang akan dihapus: **6**

Game berhasil dihapus
<p>ENTER COMMAND: <b>DELETE GAME</b></p> <p>Berikut adalah daftar game yang tersedia</p> <ol style="list-style-type: none"> <li>1. RNG</li> <li>2. LUNCH SLOW</li> <li>3. DINOSAUR IN EARTH</li> <li>4. RISEWOMAN</li> <li>5. EIFFEL TOWER</li> </ol> <p>Masukkan nomor game yang akan dihapus: 1</p> <p>Game gagal dihapus</p>

**g. QUEUEGAME**

QUEUEGAME merupakan command yang digunakan untuk mendaftarkan permainan kedalam list. List dalam queue akan hilang ketika pemain menjalankan command **QUIT**.

<p>ENTER COMMAND: <b>QUEUE GAME</b></p> <p>Berikut adalah daftar antrean game-mu</p> <ol style="list-style-type: none"> <li>1. EIFFEL TOWER</li> <li>2. RISEWOMAN</li> <li>3. LUNCH SLOW</li> </ol> <p>Berikut adalah daftar game yang tersedia</p> <ol style="list-style-type: none"> <li>1. RNG</li> <li>2. LUNCH SLOW</li> <li>3. DINOSAUR IN EARTH</li> <li>4. RISEWOMAN</li> <li>5. EIFFEL TOWER</li> </ol> <p>Nomor Game yang mau ditambahkan ke antrean: <b>2</b></p> <p>Game berhasil ditambahkan ke dalam daftar antrean.</p>
<p>ENTER COMMAND: <b>QUEUE GAME</b></p> <p>Berikut adalah daftar antrean game-mu</p> <ol style="list-style-type: none"> <li>1. EIFFEL TOWER</li> <li>2. RISEWOMAN</li> <li>3. LUNCH SLOW</li> </ol>

Berikut adalah daftar game yang tersedia

1. RNG
2. LUNCH SLOW
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER

Nomor Game yang mau ditambahkan ke antrean: **9**

Nomor permainan tidak valid, silahkan masukkan nomor game pada list.

#### **h. PLAYGAME**

PLAY GAME merupakan command yang digunakan untuk memainkan sebuah permainan. Game yang dimainkan adalah game dengan urutan pertama di antrean 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.

ENTER COMMAND: **PLAY GAME**

Berikut adalah daftar Game-mu

1. EIFFEL TOWER
2. RISEWOMAN
3. LUNCH SLOW

~~ENTER GAME: 1~~

Loading EIFFEL TOWER ...

ENTER COMMAND: **PLAY GAME**

Berikut adalah daftar Game-mu

1. RISEWOMAN
2. RISEWOMAN
3. LUNCH SLOW

~~ENTER GAME: 2~~

Game RISEWOMAN masih dalam maintenance, belum dapat dimainkan. Silahkan pilih game lain.



i. **SKIPGAME <n>**

SKIPGAME merupakan command yang digunakan untuk melewati permainan sebanyak n.

ENTER COMMAND: **SKIPGAME 2**

Berikut adalah daftar Game-mu

1. RISEWOMAN
2. LUNCH SLOW
3. RISEWOMAN

Loading RISEWOMAN ...

ENTER COMMAND: **SKIPGAME 5**

Berikut adalah daftar Game-mu

1. RISEWOMAN
2. LUNCH SLOW
3. RISEWOMAN

Tidak ada permainan lagi dalam daftar game-mu.

j. **QUIT**

Keluar dari program.

ENTER COMMAND: **QUIT**

Anda keluar dari game BNMO.

Bye bye ...

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.

ENTER COMMAND: **COMMAND\_ANEH**

Command tidak dikenali, silahkan masukkan command yang valid.

ENTER COMMAND:

## Spesifikasi Game

### 1. RNG

BNMO tidak selalu menikmati *game* yang sudah pasti *outcome*-nya. Karena itu, ia suka dengan *game* yang melibatkan RNG (*Random number generator*). Berikut adalah spesifikasi *game* ini:

- Setiap permainan dimulai dengan program sudah menentukan sebuah angka acak X.
- Di setiap giliran, pemain diberi kesempatan menebak angka X. *Game* akan memberi tahu apakah tebakan pemain dibandingkan terhadap X lebih besar atau lebih kecil.
- Permainan selesai jika pemain menebak angka X dengan benar.
- Skor untuk *game* ini tergantung dengan seberapa cepat pemain menebak X. Formula skor dibebaskan.
- Batasan X dan maksimal giliran dibebaskan.

Catatan:

- Gunakan fungsi pembangkit angka acak bawaan bahasa c.
- Secara *default*, pembangkit angka acak bawaan bahasa c menghasilkan *pseudo random number* yang artinya angka yang dihasilkan memiliki pola dan tidak benar-benar acak. Cari tahu bagaimana membangkitkan angka acak dengan benar-benar acak.

RNG Telah dimulai. Uji keberuntungan Anda dengan menebak X.

Tebakan: **80**

Lebih kecil

Tebakan: **40**

Lebih besar

Tebakan: **41**

Ya, X adalah 41.

### 2. Diner Dash

Indra dan Doni juga suka permainan yang menegangkan. Oleh karena itu, ia ingin ada sebuah game Diner Dash dalam BNMO. Secara singkat, Diner Dash merupakan permainan mengantar makanan namun terurut berdasarkan prioritasnya. Berikut adalah spesifikasi game ini:

- Terdapat 3 command yang dapat dilakukan pada game, yaitu **COOK** dan **SERVE**
  - **COOK** merupakan command yang bertujuan untuk memasak makanan

- **SERVE** merupakan command yang bertujuan untuk menyajikan makanan kepada pelanggan.
- **SKIP** merupakan command yang bertujuan untuk menyelesaikan 1 putaran tanpa melakukan apa apa (seluruh durasi memasak dan ketahanan berkurang 1, pelanggan bertambah 1).
- Command **selain COOK dan SERVE** dianggap tidak valid dan **tidak terhitung sebagai satu putaran**.
- Command **COOK dan SERVE yang tidak valid juga tidak terhitung sebagai satu putaran**
- Permainan akan dimulai dengan 3 pelanggan. Setiap pelanggan hanya dapat memesan satu makanan. Untuk setiap makanan, terdapat informasi tentang ID makanan yang dihasilkan secara *increment* (M01, M02, M03, dst), durasi memasak, harga makanan, serta ketahanan makanan. Semua informasi tersebut akan didapatkan secara random dengan menggunakan **random number generator**. Durasi dan ketahanan makanan akan berkisar diantara 1-5. Sedangkan, harga makanan akan berkisar diantara 10000 - 50000.
- Kapasitas dari pemain adalah memasak 5 makanan dalam waktu yang sama. Pelanggan yang dilayani adalah pelanggan yang duluan memasuki antrean.
- Permainan selesai apabila antrean melebihi 7 pelanggan atau jumlah pelanggan yang sudah dilayani mencapai 15 pelanggan.
- Pada setiap putaran, akan terdapat 1 pelanggan baru.
- Pada setiap putaran, seluruh durasi dari makanan yang sedang dimasak akan berkurang 1. Ketika durasi makanan mencapai 0, maka makanan sudah dapat di SERVE.
- Pada setiap putaran, seluruh ketahanan dari makanan yang sudah dapat disajikan akan berkurang 1. Ketika ketahanan makanan mencapai 0, maka makanan sudah akan hangus dan harus ulang dimasak.
- Ketika makanan sudah di SERVE, maka makanan dapat diantar kepada pelanggan dan pelanggan dapat meninggalkan antrean. Setelah pelanggan meninggalkan antrean, maka pemain akan menerima uang
- SERVE hanya dapat digunakan untuk pesanan yang berada di paling depan.
- Skor akhir dari pemain adalah total uang yang diterima oleh pemain.

Selamat Datang di Diner Dash!

SALDO: 0

Daftar Pesanan

Makanan | Durasi memasak | Ketahanan | Harga

M0	2	3	15000
M1	3	1	15000
M2	1	4	15000

Daftar Makanan yang sedang dimasak

Makanan | Sisa durasi memasak

-----  
|

Daftar Makanan yang dapat disajikan

Makanan | Sisa ketahanan makanan

-----  
|

MASUKKAN COMMAND: **COOK M0**

Berhasil memasak M0

=====

SALDO: 0

Daftar Pesanan

Makanan | Durasi memasak | Ketahanan | Harga

M0	2	3	15000
M1	3	1	15000
M2	1	4	15000
M3	1	4	15000

Daftar Makanan yang sedang dimasak

Makanan | Sisa durasi memasak

-----  
M0 | 2

Daftar Makanan yang dapat disajikan  
Makanan | Sisa ketahanan makanan

-----

|

MASUKKAN COMMAND: **COOK M1**

Berhasil memasak M1

=====

SALDO: 0

Daftar Pesanan

Makanan | Durasi memasak | Ketahanan | Harga

-----

M0	2	3	15000
M1	3	1	15000
M2	1	4	15000
M3	1	4	15000
M4	1	4	15000

Daftar Makanan yang sedang dimasak  
Makanan | Sisa durasi memasak

-----

M0	1
M1	3

Daftar Makanan yang dapat disajikan  
Makanan | Sisa ketahanan makanan

-----

|

MASUKKAN COMMAND: **COOK M2**

Berhasil memasak M2

Makanan M0 telah selesai dimasak

=====

SALDO: 0

Daftar Pesanan

Makanan | Durasi memasak | Ketahanan | Harga

M0 | 2 | 3 | 15000  
M1 | 3 | 1 | 15000  
M2 | 1 | 4 | 15000  
M3 | 3 | 1 | 15000  
M4 | 1 | 4 | 15000  
M5 | 1 | 4 | 15000

Daftar Makanan yang sedang dimasak

Makanan | Sisa durasi memasak

M1 | 2  
M2 | 1

Daftar Makanan yang dapat disajikan

Makanan | Sisa ketahanan makanan

M0 | 3

MASUKKAN COMMAND: **SERVE M0**

Berhasil mengantar M0

Berhasil memasak M2

SALDO: 15000

Daftar Pesanan

Makanan | Durasi memasak | Ketahanan | Harga

M1 | 3 | 1 | 15000  
M2 | 1 | 4 | 15000  
M3 | 3 | 1 | 15000  
M4 | 1 | 4 | 15000  
M5 | 1 | 4 | 15000  
M6 | 1 | 4 | 15000

Daftar Makanan yang sedang dimasak

Makanan | Sisa durasi memasak

M1 | 1

Daftar Makanan yang dapat disajikan  
Makanan | Sisa ketahanan makanan

-----  
M2 | 4

MASUKKAN COMMAND: **SERVE M2**

M2 belum dapat disajikan karena M1 belum selesai

### 3. Game tambahan/ Buatan pemain

Game buatan pemain yang dibuat menggunakan command CREATE GAME akan langsung selesai dan masuk ke tahap game over dengan skor akhir berupa integer random.

### Daftar ADT yang Digunakan

Anda diwajibkan menggunakan ADT di bawah ini. Selain itu, Anda dapat pula menggunakan ADT lain, namun cantumkan analisis alasan kenapa menggunakan ADT tersebut pada laporan.

#### 1. ADT Array

ADT ini digunakan untuk merepresentasikan daftar game yang terdapat dalam sistem.

#### 2. ADT Mesin Karakter dan Mesin Kata

ADT ini digunakan untuk melakukan parsing command program dan pembacaan file konfigurasi ke dalam aplikasi.

#### 3. ADT Queue

ADT ini digunakan untuk menentukan urutan bermain game serta salah satu game, yaitu Diner Dash.

## 8.2 Notulen Rapat

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







No. Kelompok/Kelas : 05/03  
Nama Kelompok : Kelompok LYMA  
Anggota Kelompok (Nama/NIM) :  
1. Richard Haris/18221006  
2. Harits Afiq Nugroho/18221012  
3. Justin Yusuf Abidjoko/18221016  
4. Ahmad RIvai Yahya/18221017  
5. Anjani Ibrahim/18221031  
Asisten Pembimbing : Aditya Bimawan

---







### Asistensi I

<b>Tanggal : 03 November 2022</b>	<b>Catatan Asistensi:</b>
<b>Tempat : Google Meet</b>	<ul style="list-style-type: none"><li>• File konfigurasi boleh diubah dikit-dikit seperti ditambah mark. Jawabannya boleh, tapi lebih baik dipastikan bertanya di FAQ.</li><li>• Boleh menggunakan fungsi atoi (mengubah ke integer).</li><li>• Menanyakan urutan main untuk diner dash.</li><li>• Apakah di game RNG ada angka maksimal? Jawabannya bebas karena di speknya juga tidak ada batasan.</li><li>• Fungsi dari game buatan user dengan fungsi creategame apakah sekarang tinggal print saja? Iya.</li><li>• antrean pada diner dash tidak akan habis. Cook dan serve mungkin habis.</li><li>• Penggunaan ADT terserah kita, jika ADT Queue tidak digunakan, apakah perlu digunakan? Jawabannya harus menggunakan ADT Queue. File save apakah boleh menambahkan “.csv” sendiri? Boleh tapi dijelaskan saja.</li><li>• Sistem penyelenggaraan demo seperti apa? Belum tahu (masih diomongkan dengan dosen. Bisa jadi offline).</li><li>• Deadline tanggal 11 meliputi semuanya (source file dan laporan). Tidak menggunakan scanf tapi mesinkata → diawali dengan start kata lalu infinite loop.</li><li>• Apakah di dalam console.c dimasukkan semua fungsinya? Boleh dipecah boleh tidak.</li></ul>



<p><b>Kehadiran Anggota Kelompok:</b></p> <p>1 18221006</p>  <p>2 18221012</p>  <p>3 18221016</p>  <p>4 18221017</p>  <p>5 18221031</p> 	<ul style="list-style-type: none"> <li>• Laporan bagian 6 dan 7 bedanya apa? Data test lebih ke kondisi program. Misal ada data test 1 yang queue-nya kosong.</li> <li>• Beda pembagian kelompok dan log activity? log activity terdapat waktu (perminggu atau perhari).</li> <li>• Kalau mau bikin ADT Key tapi ElTypenya beda gimana kak? Terserah kalian.</li> <li>• Jika mau nanya apakah bisa langsung ke asisten? langsung ke FAQ</li> </ul>
	<p><b>Tanda Tangan Asisten:</b></p> 

## Asistensi II

<b>Tanggal : 10 November 2022</b>	<b>Catatan Asistensi:</b> <ul style="list-style-type: none"> <li>• Apakah perlu dibuat driver masing-masing untuk tiap ADT? Iya, dibuat semuanya ya. Boleh menggunakan test case yang terdapat pada praktikum</li> <li>• Apakah boleh kita minta input “0” untuk pemain keluar dari game ditengah permainan? boleh, silahkan</li> <li>• Dalam algoritma menarik, definisi menarik itu apa? Kalau menurut kalian menarik masukan aja</li> <li>• Di lampiran laporan, terdapat deskripsi tugas besar dua, diisi apa? diisi tugas besar satu</li> <li>• Melakukan demo program dengan meminta saran jika masih terdapat kekurangan</li> <li>• Jadi, apakah boleh “cook” makanan yang sama lebih dari 1 kali untuk game diner dash? boleh ya</li> <li>• Pembuatan instruksi kompilasi program apakah boleh menggunakan file .exe langsung? tidak boleh harus instruksi meng-compilanya</li> <li>• Untuk tanda tangan kakak bagaimana kak (untuk keperluan form asistensi) ? silahkan chat saja ya, nanti di copy kan gambar tanda tangannya</li> </ul>
<b>Tempat : Google Meet</b>	
<b>Kehadiran Anggota Kelompok:</b> <div style="text-align: center;"> <p>1 18221006</p>  <p>2 18221012</p>  <p>3 18221016</p>  <p>4 18221017</p>  <p>5 18221031</p>  </div>	
	<b>Tanda Tangan Asisten:</b> 

### 8.3 Log Activity Anggota Kelompok

No.	Waktu	Keterangan
1.	29 Oktober 2022	Pembagian tugas
2.	30 Oktober 2022 - 9 November 2022	Pengerjaan dan menyatukan program berdasarkan tugas masing-masing
3.	03 November 2022	Asistensi pertama
4.	10 November 2022	Asistensi kedua, membuat laporan, dan menyempurnakan program
5.	11 November 2022	Finalisasi akhir dan pengumpulan tugas