

**LAPORAN TUGAS BESAR**  
**IF2110/Algoritma dan Struktur Data**


***BNMO Cooking Game Simulator***

Dipersiapkan oleh:

Kelompok H Kelas 2

13521129	Chiquita Ahsanunnisa
13521139	Nathania Calista Djunaedi
13521157	Hanif Muhammad Zhafran
13521161	Ferindya Aulia Berlianty
13521163	Zidane Firzatullah

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>IF2110-TB-H-2</i>		43
		<i>Revisi</i>	1	20 November 2022

# Daftar Isi

1 Ringkasan .....	4
1.1 Deskripsi Umum Persoalan .....	4
1.2 Ringkasan Isi Laporan .....	4
1.3 Kesimpulan Umum Tugas Besar .....	4
2 Penjelasan Tambahan Spesifikasi Tugas .....	4
2.1 Validasi Input Command .....	4
2.2 Help .....	5
2.3 Kulkas .....	5
2.4 Waktu Pengolahan Makanan .....	5
2.5 Rekomendasi Makanan .....	5
3 Struktur Data (ADT) .....	5
3.1 ADT Point .....	5
3.2 ADT Waktu .....	6
3.3 ADT Makanan .....	6
3.4 ADT Simulator .....	6
3.5 ADT List Statik .....	6
3.6 ADT Matriks .....	6
3.7 ADT Mesin Karakter dan Mesin Kata .....	7
3.8 ADT PrioQueue .....	7
3.9 ADT Stack .....	7
3.10 ADT Tree .....	7
3.11 ADT Config .....	8
3.12 ADT Commands .....	8
3.13 ADT Notif .....	8
3.14 ADT Kulkas .....	8
3.15 ADT Set .....	8
4 Program Utama .....	8
5 Algoritma-Algoritma Menarik .....	9
5.1 Two Dimensional Bin Packing Algorithm .....	9
5.2 Union-Find Path Compression .....	11
6 Data Test .....	11
6.1 Test START/EXIT GAME .....	11
6.2 Data Test Config Makanan .....	12
6.3 Data Test Config Peta .....	13
6.4 Data Test Konfigurasi Resep .....	14
6.5 Data Test Move .....	14
6.6 Data Test Buy .....	18
6.7 Data Test Chop .....	20
6.8 Data Test Mix .....	21
6.9 Data Test Fry .....	23
6.10 Data Test Boil .....	25
6.11 Data Test Delivery .....	27

6.12	Data Test Inventory .....	27
6.13	Data Test Wait .....	28
6.14	Data Test Insert Kulkas .....	29
6.15	Data Test Take Kulkas .....	30
6.16	Data Test Kulkas .....	32
6.17	Data Test Rekomendasi .....	33
6.18	Data Test Undo .....	33
6.19	Data Test Redo .....	34
6.20	Data Test Notifikasi.....	35
6.21	Data Test Help .....	35
7	Test Script.....	36
8	Pembagian Kerja dalam Kelompok .....	41
9	Lampiran.....	41
9.1	Deskripsi Tugas Besar 2 .....	41
9.2	Notulen Rapat .....	42
9.3	Log Activity Anggota Kelompok .....	42

# 1 Ringkasan

## 1.1 Deskripsi Umum Persoalan

Indra dan Doni memiliki sebuah *robot game* bernama BNMO. Indra meminta bantuan kepada BNMO dan Doni untuk membantu mempersiapkan makan malam. Namun, BNMO tidak tahu cara memasak dan Doni tidak bisa membantu persiapan makan malam karena ada hal lain. BNMO hanya dapat melakukan program simulasi. Oleh karena itu, Doni meminta bantuan penulis untuk membuat program simulasi tersebut.

Dalam persoalan kali ini, kami diberikan tugas membuat sebuah program simulasi berbasis CLI (*command-line interface*). Program ini dibuat dalam bahasa C dengan menggunakan struktur data yang sudah dipelajari dalam mata kuliah Algoritma dan Struktur Data.

## 1.2 Ringkasan Isi Laporan

Pada tugas besar IF2110 Algoritma dan Struktur Data 2022, penulis mendapat tantangan untuk membuat sebuah program simulasi berbasis *command-line interface* (CLI) mengenai pemesanan bahan makanan, pengolahan bahan makanan, dan *delivery*.

Laporan ini dibuat untuk merangkum hasil tugas besar yang penulis lakukan dan mencakup beberapa penjelasan di dalamnya. Penjelasan tersebut diantaranya, spesifikasi tambahan untuk fitur bonus, struktur data (ADT) yang digunakan, alur program utama, algoritma yang menurut penulis menarik, hasil testing program, dan beberapa lampiran.

Secara garis besar, hasil dari tugas besar penulis telah memenuhi semua spesifikasi tugas yang diberikan dan berhasil dijalankan pada mesin Windows-based dan UNIX-based. Tugas besar yang diberikan melatih penulis agar lebih memahami dan mengeksplor berbagai struktur data atau ADT yang selama ini telah dipelajari dalam kuliah IF2110 Algoritma dan Struktur Data menjadi sebuah program simulasi yang dapat melakukan berbagai *command* tertentu.

## 1.3 Kesimpulan Umum Tugas Besar

Dengan memanfaatkan *tools* bahasa pemrograman C dan materi yang telah dipelajari dalam perkuliahan Algoritma dan Struktur Data, kelompok penulis dapat membuat program yang berfungsi untuk melakukan sebuah simulasi mulai dari proses pemesanan makanan, pengolahan makanan, dan berbagai fitur yang ada di dalamnya.

# 2 Penjelasan Tambahan Spesifikasi Tugas

Pada tugas besar ini, penulis menambahkan sedikit spesifikasi tambahan dari fitur wajib dan mengerjakan berbagai fitur bonus yang diberikan. Berikut adalah detailnya.

## 2.1 Validasi Input Command

Pada program ini, terdapat sedikit perubahan dibanding spesifikasi validasi yang diberikan. Jika ada dua jenis masukan yang terdiri dari huruf yang sama namun berbeda kapital atau tidaknya, keduanya akan dianggap sama. Sebagai contoh, *command* masukan "START" diterima. *Command* masukan "start", "Start", dan kombinasi lainnya tetap diterima. Hal ini dibuat untuk menambah kenyamanan bagi pemain saat memberi masukan.

## 2.2 Help

Pada program ini, penulis menambahkan command HELP yang menampilkan list command yang valid. Hal ini ada untuk membantu pemain dalam memberikan masukan yang valid.

## 2.3 Kulkas

Pada program ini, penataan kulkas dilakukan oleh program sendiri. Algoritma yang digunakan adalah *Two Dimensional Bin Packing Algorithm*. Cara penataan ini memanfaatkan konsep *binary tree* yang diperjelas di bagian Algoritma-Algoritma Menarik.

Saat menaruh makanan di kulkas, makanan tersebut juga akan dihilangkan dari Inventory. Selama makanan berada di kulkas, sisa waktu kedaluarsa tidak akan berkurang. Setelah dikeluarkan dari kulkas, makanan akan langsung berada di Inventory lagi. *Command* untuk memasukkan makanan ke kulkas dan mengambil makanan dari kulkas akan menambah waktu game sebanyak satu menit. Untuk menaruh dan mengambil makanan dari kulkas, pengguna tidak perlu datang ke lokasi tertentu. Peletakkan makanan pada kulkas diletakkan dalam sebuah matriks dan makanan tidak bisa ditumpuk.

## 2.4 Waktu Pengolahan Makanan

Dalam tugas besar kali ini, untuk fitur bonus, makanan memiliki waktu pengolahan. Misalnya, sebelum menggunakan fitur bonus ini, proses semua makanan dianggap memiliki waktu 0 menit, sehingga makanan dapat langsung masuk ke dalam Inventory. Namun, dengan fitur waktu pengolahan makanan, tiap makanan bisa memiliki waktu pengolahan yang berbeda-beda. Misalnya, menggoreng ayam memerlukan waktu 20 menit, sedangkan memotong ayam hanya perlu waktu 5 menit.

Dalam mengerjakan spesifikasi tambahan ini, penulis mengasumsikan bahwa waktu otomatis bertambah ketika makanan diolah dan pemain tidak bergerak ke mana-mana. Sehingga, konsepnya hampir serupa dengan *command* WAIT.

## 2.5 Rekomendasi Makanan

Fitur rekomendasi makanan ini akan menampilkan rekomendasi makanan berdasarkan bahan makanan yang tersedia di Inventory. Sehingga jika makanan yang salah satu bahannya tidak tersedia di Inventory, maka makanan tersebut tidak akan direkomendasikan.

## 3 Struktur Data (ADT)

Dalam tugas besar kali ini, penulis menggunakan beberapa struktur data. Berikut adalah struktur data yang kami gunakan dalam program penulis.

### 3.1 ADT Point

ADT Point merupakan ADT sederhana yang tersusun dari komponen absis dan ordinat yang bertipe integer. Implementasi ADT ini mencakup selektor, konstruktor, kelompok baca-tulis, kelompok menentukan letak titik, dan kelompok operasi lain, yang dapat ditemukan pada file point.c. ADT ini digunakan untuk merepresentasikan lokasi simulator dan benda-benda lainnya di dalam peta karena ADT ini dapat mewakili koordinat di bidang dua dimensi.

### 3.2 ADT Waktu

ADT Waktu merupakan ADT sederhana yang tersusun dari komponen hari, jam, dan menit yang bertipe integer. Implementasi ADT ini mencakup kelompok validasi *type*, kelompok baca-tulis, kelompok konversi terhadap *type*, kelompok operasi (aritmatika dan relasional) terhadap *type* yang dapat ditemukan pada file *time.c*. ADT ini digunakan untuk merepresentasikan waktu permainan yang berjalan karena ADT ini dapat mewakili waktu dalam bentuk hari, jam, dan menit.

### 3.3 ADT Makanan

ADT Makanan merupakan ADT sederhana yang tersusun dari komponen ID makanan, nama makanan, waktu kedaluarsa, waktu aksi, dan aksi, lebar, dan tinggi makanan. Pada ADT ini juga terdapat sub-ADT lainnya, yaitu ADT List Makanan yang merupakan sebuah list statik untuk menyimpan tabel makanan. Implementasi dari ADT ini mencakup selektor, prototype, kreator, primitif add/delete, operasi tambahan, seperti pengelompokan makanan, dan operasi – operasi lain yang dapat dilihat dalam file *makanan.c*. ADT ini digunakan untuk menyimpan data tiap makanan yang ada dalam permainan. Selanjutnya, makanan akan disimpan dalam list makanan yang bersifat statik karena pada persoalan konfigurasi di permainan ini, jumlah makanan – yang berarti jumlah elemen dalam list – sudah diberi tahu terlebih dahulu di awal konfigurasi.

### 3.4 ADT Simulator

ADT Simulator merupakan ADT sederhana yang tersusun dari komponen nama pengguna, lokasi simulator saat ini. Implementasi ADT ini hanya mencakup konstruktor yang dapat ditemukan pada file *simulator.c*. ADT ini digunakan untuk menyimpan data simulator beserta kepemilikannya (dalam inventory), serta menjadi acuan state program untuk fitur undo dan redo.

### 3.5 ADT List Statik

ADT List Statik merupakan array dengan ukuran yang bersifat statis dengan model implisit (diberi penanda untuk nilai yang kosong). ADT ini tersusun dari komponen tabel penyimpanan elemen. List statik pada ADT ini bersifat immutable, dan hanya bisa di-copy. Implementasi ini mencakup selektor, konstruktor, selektor tambahan, selektor indeks, kelompok baca tulis, operator aritmatika, operator relasional, searching, nilai ekstrem, fungsi penambahan elemen, fungsi untuk menghapus elemen, dan sorting yang dapat ditemukan pada file *liststatik.c*. ADT ini digunakan untuk menyimpan list makanan dan list resep. List ini berfungsi untuk dijadikan acuan ketika pemesanan bahan makanan dan berisi data dari konfigurasi bukan data dari state aplikasi.

### 3.6 ADT Matriks

ADT Matriks merupakan array dua dimensi dengan ukuran statis. Implementasi pada ADT ini mencakup selektor, konstruktor, kelompok baca tulis, dan kelompok fungsi-fungsi pada peta yang dapat ditemukan pada file *matrix.c*. ADT ini digunakan untuk merepresentasikan peta pada permainan karena struktur data matriks mampu mewakili peta dua dimensi yang digunakan pada permainan.

### **3.7 ADT Mesin Karakter dan Mesin Kata**

ADT Mesin Karakter dan Mesin Kata adalah ADT yang digunakan untuk membaca input dari pengguna. ADT ini memiliki dua source code, yaitu `charmachine.c` dan `wordmachine.c`. ADT Charmachine (character machine) memproses karakter per karakter, sedangkan ADT Wordmachine memproses kata per kata dan dibuat dengan dasar charmachine. ADT ini digunakan untuk melakukan parsing command program dan pembacaan file konfigurasi ke dalam aplikasi.

### **3.8 ADT PrioQueue**

ADT PrioQueue adalah queue berbasis array list dinamik yang elemennya terurut membesar berdasarkan prioritas waktu delivery atau waktu expired suatu makanan. ADT ini memiliki komponen jenis waktu prioritas (berdasarkan waktu delivery atau waktu expired, diwakili oleh suatu nilai boolean), tabel penyimpan elemen makanan, alamat penghapusan (head), alamat penambahan (tail), dan jumlah maksimal elemen queue. Implementasi dari ADT ini mencakup selektor, prototype, kreator, destruktur, primitif add/delete, operasi tambahan, pengubahan jumlah maksimal elemen, mekanisme waktu, dan juga operasi seperti array (ini dibutuhkan karena penghapusan elemen tidak serta merta mengikuti aturan queue) yang dapat ditemukan pada file `prioqueue.c`. ADT ini dibuat dalam bentuk array list dinamik untuk mempermudah proses penyimpanan makanan, dan dibuat dalam bentuk priority queue untuk mempermudah proses penghapusan elemen karena elemen yang diprioritaskan lebih dahulu untuk dihapus dari queue adalah elemen yang waktunya lebih sedikit atau nol. ADT ini digunakan untuk membuat Delivery List dan Inventory Makanan.

### **3.9 ADT Stack**

ADT Stack merupakan array statis yang akses elemennya mengikuti aturan last in, first out (LIFO). ADT ini tersusun atas komponen tabel penyimpan data dengan tipe states dan alamat top stack. Tipe data states sendiri terdiri dari ADT Time untuk merepresentasikan waktu dalam game, ADT POINT untuk merepresentasikan posisi pemain, PrioQueue untuk merepresenatasikan Delivery dan Inventory List, serta stateNotif untuk menggambarkan notifikasi yang akan diberikan. Implementasi ADT ini mencakup selektor, konstruktor, destruktur, primitif add/delete, operasi – operasi tambahan seperti command UNDO, REDO, dan beberapa fungsi lainnya, yang dapat ditemukan pada file `stack.c`. ADT ini digunakan untuk meng-undo atau meng-redo command yang dilakukan oleh simulator pada aplikasi.

### **3.10 ADT Tree**

ADT Tree merupakan ADT berbasis node yang tersusun atas komponen value (nilai yang disimpan), n-ary node untuk children, jumlah children, dan ukuran array child. Pada ADT ini terdapat sub-ADT lainnya yaitu ADT List Tree yang merupakan sebuah list statik model eksplisit untuk menyimpan tabel berisi tree resep. Implementasi ADT ini mencakup konstruktor, primitif add, traverse, kelompok searching, yang dapat ditemukan pada file `tree.c`. ADT ini digunakan untuk menyimpan makanan, karena ADT ini menggunakan N-ary Tree agar jumlah childnya bisa tidak ada sampai berapapun, sesuai dengan struktur resep yang ada pada permainan.

### 3.11 ADT Config

ADT Config adalah ADT yang digunakan untuk membaca file konfigurasi makanan dan konfigurasi peta pada awal permainan. ADT ini hanya digunakan sekali saja, yaitu pada saat pemain BNMO mengirimkan command START.

### 3.12 ADT Commands

ADT Commands merupakan salah satu ADT tambahan yang dibuat untuk mempersingkat dan membuat tampilan *main* program menjadi lebih rapi. Secara umum, ADT Commands digunakan untuk handle beberapa input user terkait makanan dan memvalidasi input dari user.

### 3.13 ADT Notif

ADT Notif adalah ADT yang digunakan untuk mengimplementasikan fitur notifikasi. Struct yang digunakan adalah *stateNotif* yang berisi detail command berpengaruh yang dilakukan user, list makanan yang kedaluarsa, dan list makanan yang sampai dari delivery. Ketiga bagian tersebut memiliki struktur data *WordList*. Implementasi ADT ini mencakup pengecekan, konstruktor, adder dan setter, serta printing sesuai mode notif yang digunakan. ADT ini diimplementasikan pada file *notif.c*. Di sini, digunakan *WordList* karena notif hanya melibatkan penampilan dan printing saja, jadi dengan pengolahan word dan string saja sudah cukup.

### 3.14 ADT Kulkas

ADT Kulkas adalah ADT yang digunakan untuk mengimplementasikan fitur bonus yang pertama, yaitu kulkas. Kulkas diimplementasikan dengan tiga struktur: (1) list kulkas, merupakan *ordered array* yang daftar makanan yang berhasil masuk ke kulkas yang terurut berdasarkan ukuran, (2) *binary tree* yang diimplementasikan dengan array, berisi konfigurasi peletakan makanan dalam kulkas, (3) *matrix* kulkas berisi konfigurasi dan tampilan kulkas. Implementasi ADT ini mencakup primitif tiap struktur, serta mekanisme memasukkan, mendelete, dan *re-arrange* elemen kulkas yang dapat ditemukan pada file *kulkas.c*. Pada ADT ini, digunakan *ordered array* untuk mempermudah peletakan makanan secara otomatis. Digunakan pula konsep *binary tree* untuk mekanisme peletakan makanan secara otomatis.

### 3.15 ADT Set

ADT Set adalah ADT yang digunakan untuk mengimplementasikan fitur bonus yang ketiga, yaitu rekomendasi makanan. ADT ini memanfaatkan *n-ary tree* untuk penyimpanan elemen.

## 4 Program Utama

Program utama terletak pada file *main.c*. Saat pertama kali dijalankan, program akan menampilkan splash screen serta menu untuk memulai game (START) atau menutup game (EXIT).

Jika pengguna memilih command START, konfigurasi makanan, peta, dan resep untuk memulai game akan dijalankan. Pengguna akan diminta untuk memasukkan nama pengguna yang akan dimasukkan ke data simulator. *Queue Inventory* dan *Delivery* juga diinisialisasi sebagai *queue* kosong. Waktu permainan juga akan diinisialisasi dengan waktu nol (0 hari, 0 jam, 0 menit).



Setelah konfigurasi awal dan inisialisasi selesai, game akan melakukan looping: game menampilkan tampilan layar yang mencakup waktu, lokasi simulator, notifikasi, dan peta, lalu game akan meminta masukan command. Pengguna dapat memilih command sebagai berikut.

Command	Keterangan
BUY	Memesan bahan makanan
MIX	Mencampurkan dua atau lebih makanan menjadi satu makanan
CHOP	Memotong satu bahan makanan menjadi bahan yang sudah terpotong
FRY	Meng goreng satu atau lebih bahan makanan
BOIL	Merebus satu bahan makanan
MOVE NORTH, MOVE SOUTH, MOVE EAST, MOVE WEST	Menggerakkan simulator BNMO sebanyak satu satuan sesuai dengan arah mata angin
WAIT <x> <y>	Menunggu selama x jam dan y menit tanpa melakukan apa-apa
CATALOG	Menampilkan bahan dan makanan yang tersedia pada aplikasi termasuk informasi detailnya
COOKBOOK	Menampilkan resep-resep yang tersedia pada sistem
INVENTORY	Menampilkan inventory makanan milik pengguna
DELIVERY	Menampilkan delivery list
UNDO	Membatalkan command yang dilakukan sebelumnya dan mengembalikan state aplikasi ke sebelum command tersebut dieksekusi
REDO	Membatalkan command UNDO
HELP	Menampilkan guide command kepada pengguna
EXIT	Keluar dari game
KULKAS	Menampilkan makanan di kulkas
INSERT KULKAS	Menyimpan makanan dari inventory ke dalam kulkas
TAKE KULKAS	Mengambil makanan dari kulkas dan menaruhnya di inventory
RECOMM	Menampilkan rekomendasi makanan yang dapat dibuat berdasarkan makanan yang dimiliki

Sebagai catatan, setiap command akan divalidasi. Setiap command valid terkait perpindahan yang dilakukan oleh BNMO dan pengolahan makanan, waktu aplikasi akan bertambah satu menit, dan memengaruhi makanan-makanan pada delivery list dan inventory.

Game akan berakhir saat pengguna memasukkan command EXIT. Di akhir game, ditampilkan splash screen untuk menutup game.

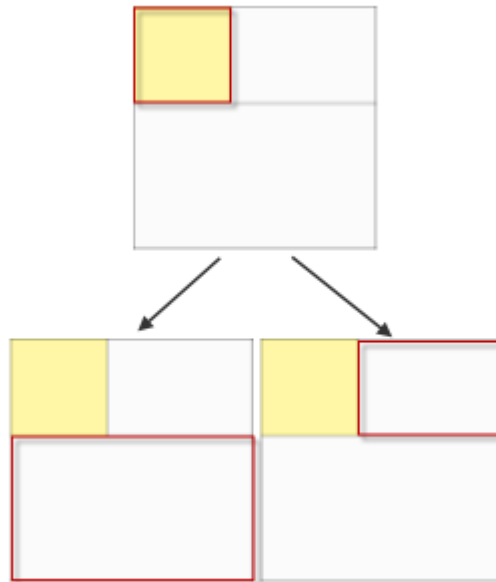
## 5 Algoritma-Algoritma Menarik

### 5.1 Two Dimensional Bin Packing Algorithm

Problem yang dibahas pada algoritma ini adalah, "diberikan suatu bin berukuran tetap (fixed size), bagaimana cara mengatur penataan balok-balok berbentuk persegi panjang sehingga space pada bin dapat digunakan secara optimal?" Jawabannya adalah dengan mengaplikasikan konsep binary tree pada penataan balok.

Pertama-tama balok-balok yang akan ditata diurutkan mengecil. Sorting yang digunakan di sini bisa beragam, bisa berdasarkan lebar balok (width), tinggi balok (height), luas balok (area), panjang sisi maksimum (maksimum antara tinggi dan luas), urutan pemasukan, dan lain-lain. Di program ini, sorting yang digunakan adalah sorting berdasarkan panjang sisi maksimum (maksimum antara tinggi dan luas). Sorting ini dipilih karena menurut [referensi](#) merupakan sorting yang menghasilkan penataan paling optimal.

Selanjutnya, letakkan balok pertama (balok paling besar) di sudut atas kiri (pada matriks, di indeks (0,0)). Akibat peletakan balok pertama tersebut, dapat terlihat bahwa bin ”terpisah” menjadi dua bagian bin kosong yang lebih kecil (bin kanan dan bin bawah).



Balok selanjutnya dapat dimasukkan ke bin kanan atau kiri. Pada algoritma ini, bagian bin kanan lebih diprioritaskan untuk diisi dibanding bin yang bawah. Saat kita memasukkan balok kedua di bin kanan, bin tersebut akan terpecah menjadi dua bagian lagi. Saat memasukkan balok selanjutnya, bin akan terus terbagi menjadi dua. Dapat dilihat bahwa struktur bin yang digunakan ini dapat diwakili dengan binary tree.



Algoritma ini adalah algoritma yang menarik karena mampu mengaplikasikan keteraturan (pola) pada penataan benda dan penggunaannya luas. Salah satunya digunakan pada fitur bonus kulkas. Program melakukan peletakan makanan dalam kulkas dengan algoritma ini agar space pada kulkas bisa dimanfaatkan secara optimal.

Union-Find algorithm merupakan algoritma yang digunakan untuk melihat di set mana sebuah elemen terletak dengan cara melihat root node dari tree elemen tersebut dan juga untuk melakukan merging antara 2 set. Union-Find Path Compression melakukan hal yang sama tetapi semua data, kecuali root node, memiliki depth 1. Konsep union-find path compression yang sudah dimodifikasi sesuai kebutuhan digunakan untuk menyelesaikan fitur bonus 3 rekomendasi makanan.

### 6.1 Test START/EXIT GAME

Saat program dijalankan, pemain hanya dapat memilih command START atau EXIT. Jika pemain memasukkan command START, maka konfigurasi akan dimulai (Gambar 6.1.1). Jika pemain memasukkan command EXIT, program akan selesai dan menampilkan splash screen penutup (Gambar 6.1.2). Jika pemain memasukkan command selain keduanya, maka program akan menampilkan pesan kesalahan dan meminta input lagi (Gambar 6.1.3).



```
Enter Command: EXIT

THANKS FOR USING

      .-. .-. .-. .-.
     /==/\ /==/\ /==/\ /==/\
    |==|  |==|  |==|  |==|
    |==|  |==|  |==|  |==|
    /==/\ /==/\ /==/\ /==/\
   /==/\ /==/\ /==/\ /==/\

----* Cooking Game Simulator *----

Created by Kelompok H Kelas K2 IF'21
```

Gambar 6.1.2 EXIT command untuk mengakhiri permainan

```
+++++
|w|e|l|c|o|m|e|t|o|
+++++

      .-. .-. .-. .-.
     /==/\ /==/\ /==/\ /==/\
    |==|  |==|  |==|  |==|
    |==|  |==|  |==|  |==|
    /==/\ /==/\ /==/\ /==/\
   /==/\ /==/\ /==/\ /==/\

----* Cooking Game Simulator *----

Created by Kelompok H Kelas K2 IF'21

Enter Command (START/EXIT): k3r3nb4ng3t
Silakan masukkan command yang valid.
Enter Command (START/EXIT): START WOI
Silakan masukkan command yang valid.
Enter Command (START/EXIT):
```

Gambar 6.1.3 Contoh input tidak valid ketika memulai permainan

## 6.2 Data Test Config Makanan

Pemain dapat melihat data makanan yang sudah dikonfigurasi dan dibaca dari file .txt dengan mengetik perintah "CATALOG". Program akan menampilkan nama makanan, durasi kadaluwarsa, aksi yang diperlukan, dan waktu untuk melaksanakan aksi dari makanan (Gambar 6.2.1).

```

Enter Command: catalog
===== List Makanan =====
Nama - Durasi Kadaluwarsa - Aksi yang Diperlukan - Action Time
1.  Ayam mentah - 1 0 0 - BUY - 0 0 15
2.  Ayam Potong - 0 2 0 - Chop - 0 0 10
3.  Tepung - 0 1 0 - Buy - 0 0 30
4.  Minyak Goreng - 0 1 30 - Buy - 0 1 0
5.  Ayam Tepung - 0 0 30 - Mix - 0 0 15
6.  Ayam Goreng - 0 0 15 - Fry - 0 30 0

```

Gambar 6.2.1 Tampilan command CATALOG

### 6.3 Data Test Config Peta

Pemain dapat langsung melihat data konfigurasi peta yang dibaca dari file .txt (Gambar 6.3.1), tanpa harus mengetikkan *command* apapun, karena peta selalu ditampilkan setiap kali ada aksi yang valid, maupun aksi yang tidak valid (Gambar 6.3.2).

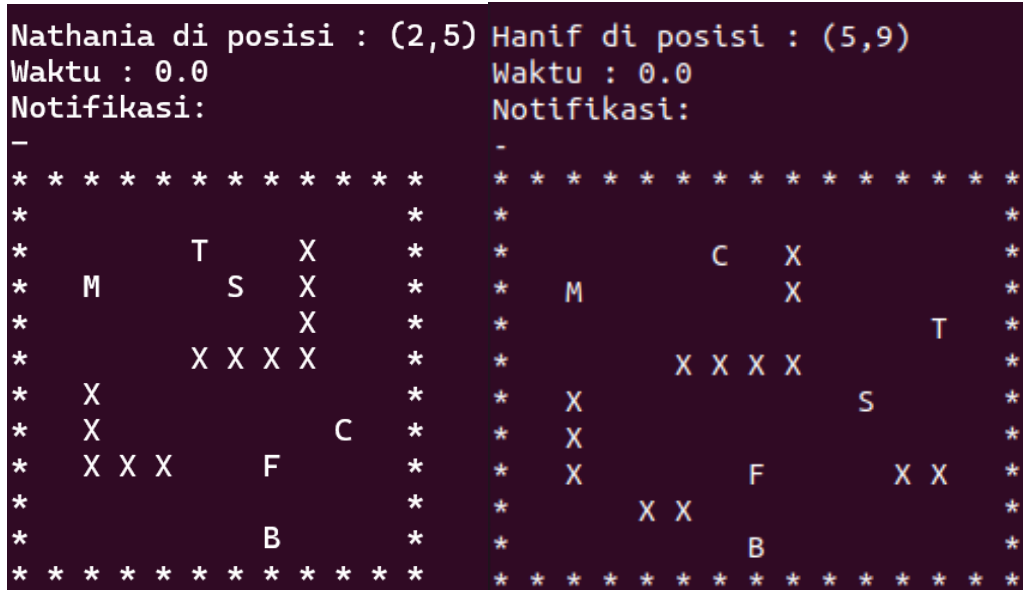
```

adt > ≡ configPeta.txt
1  10 10
2  #####
3  #####T##X##
4  #M###S#X##
5  #####X##
6  #####XXX##
7  #X#####
8  #X#####C#
9  #XXX##F###
10 #####
11 #####B###
12

adt > ≡ configPeta.txt
1  10 13
2  #####
3  #####C#X####
4  #M#####X####
5  #####T#
6  #####XXX####
7  #X#####S###
8  #X#####
9  #X####F###XX#
10 ###XX#####
11 #####B#####
12

```

Gambar 6.3.1 Isi file konfigurasi peta



Gambar 6.3.2 Tampilan peta pada program

## 6.4 Data Test Konfigurasi Resep

Pemain dapat melihat data konfigurasi resep dengan mengetik *command* "COOKBOOK" yang akan menampilkan resep – resep yang dibaca dari file .txt (Gambar 6.4.1)

```
Enter Command: cookbook
List Resep
(aksi yang diperlukan - bahan...)
1. Chop
Ayam Potong - Ayam mentah
2. Mix
Ayam Tepung - Ayam Potong - Tepung
3. Fry
Ayam Goreng - Ayam Tepung - Minyak Goreng
```

Gambar 6.4.1 Tampilan command COOKBOOK

## 6.5 Data Test Move

Pemain dapat berpindah ke kiri dengan command MOVE WEST (Gambar 6.5.2), kanan dengan command MOVE EAST (Gambar 6.5.3), ke bawah dengan command MOVE SOUTH (Gambar 6.5.4), atau ke atas dengan command MOVE NORTH (Gambar 6.5.5). Jika pemain melangkah ke ujung dari MAP atau ke salah satu simbol (X, T, M, dll.) dalam map, pemain tidak dapat jalan (Gambar 6.5.6 dan Gambar 6.5.7).

```

Nathania di posisi : (2,5)
Waktu : 0.0
Notifikasi:
-
* * * * *
*           *
*       T   X   *
*   M       S   X   *
*           X   *
*       X X X X   *
*   X           *
*   X           C   *
*   X X X       F   *
*           *
*           B   *
* * * * *
Enter Command: 

```

Gambar 6.5.1 Tampilan awal sebelum command MOVE

```

Enter Command: MOVE WEST

Nathania di posisi : (2,4)
Waktu : 0.1
Notifikasi:
-
* * * * *
*           *
*       T   X   *
*   M       S   X   *
*           X   *
*       X X X X   *
*   X           *
*   X           C   *
*   X X X       F   *
*           *
*           B   *
* * * * *
Enter Command: 

```

Gambar 6.5.2 Tampilan setelah command MOVE EAST

```

Enter Command: MOVE EAST

Nathania di posisi : (2,6)
Waktu : 0.1
Notifikasi:
-
* * * * *
*           *
*      T     X   *
*   M       S X   *
*           X     *
*        X X X X   *
*   X               *
*   X               C *
*  X X X     F     *
*               B   *
* * * * *
Enter Command: █

```

Gambar 6.5.3 Tampilan sesudah command MOVE EAST

```

Enter Command: MOVE SOUTH

Nathania di posisi : (3,5)
Waktu : 0.1
Notifikasi:
-
* * * * *
*           *
*      T     X   *
*   M       X     *
*           S X   *
*        X X X X   *
*   X               *
*   X               C *
*  X X X     F     *
*               B   *
* * * * *
Enter Command: █

```

Gambar 6.5.4 Tampilan sesudah command MOVE SOUTH



```

Enter Command: MOVE NORTH

Nathania di posisi : (1,5)
Waktu : 0.1
Notifikasi:
-
* * * * *
*           T S   X   *
*      M           X   *
*           X     X   *
*      X X X X     *
*      X           C   *
*      X X X     F     *
*           B         *
* * * * *
Enter Command: 

```

Gambar 6.5.5 Tampilan sesudah command MOVE NORTH

```

Enter Command: MOVE WEST
Simulator tidak bisa bergerak ke tempat tersebut!

Nathania di posisi : (1,5)
Waktu : 0.1
Notifikasi:
-
* * * * *
*           T S   X   *
*      M           X   *
*           X     X   *
*      X X X X     *
*      X           C   *
*      X X X     F     *
*           B         *
* * * * *
Enter Command: 

```

Gambar 6.5.6 Tampilan ketika pemain tidak bisa bergerak

```

Enter Command: MOVE NORTH
Simulator tidak bisa bergerak ke tempat tersebut!

Nathania di posisi : (0,5)
Waktu : 0.2
Notifikasi:
-
* * * * *
*           S           *
*       T       X       *
*   M           X       *
*           X       *
*       X X X X       *
*   X           *
*   X           C       *
*   X X X       F       *
*           B           *
* * * * *
Enter Command: 

```

Gambar 6.5.7 Tampilan ketika pemain tidak bisa bergerak

## 6.6 Data Test Buy

Pemain hanya dapat melakukan command "BUY" jika berada di area sekitar telepon (T). Jika pemain tidak ada di sekitar area telepon (T), program akan menampilkan pesan kesalahan (Gambar 6.6.1). Jika pemain sedang berada di sekitar area telepon, program akan menampilkan list makanan – makanan yang bisa dibeli. Setelah itu, pemain diminta memasukkan nomor makanan yang ingin dibeli. Jika nomor makanan yang ingin dibeli valid, makanan akan masuk ke delivery list dan waktu permainan bertambah 1 menit (Gambar 6.6.2). Jika nomor yang dimasukkan tidak valid, akan muncul pesan kesalahan (Gambar 6.6.3). Jika nomor yang dimasukkan adalah 0, program akan kembali ke menu utama (Gambar 6.6.4).

```

Enter Command: BUY
BNMO tidak berada di area telepon!

Nathania di posisi : (0,5)
Waktu : 0.2
Notifikasi:
-
* * * * *
*           S           *
*       T       X       *
*   M           X       *
*           X       *
*       X X X X       *
*   X           *
*   X           C       *
*   X X X       F       *
*           B           *
* * * * *
Enter Command: 

```

Gambar 6.6.1 Tampilan ketika pemain tidak berada di area telepon

```

Nathania di posisi : (0,4)
Waktu : 0.3
Notifikasi:
-
* * * * *
*       S       *
*       T       X   *
*   M       X   *
*       X   *
*   X X X X   *
*   X       *
*   X       C   *
*   X X X   F   *
*       *
*       B       *
* * * * *
Enter Command: BUY
=====
=====   Buy   =====
=====
1. Ayam mentah (15 menit)
2. Tepung (30 menit)
3. Minyak Goreng (1 jam)
Masukkan 0 untuk kembali ke Menu
Enter command:
1
Berhasil memesan Ayam mentah. Ayam mentah akan diantar dalam 15 menit

```

Gambar 6.6.2 Tampilan ketika pemain berhasil memesan makanan

```

Enter Command: BUY
=====
=====   Buy   =====
=====
1. Ayam mentah (15 menit)
2. Tepung (30 menit)
3. Minyak Goreng (1 jam)
Masukkan 0 untuk kembali ke Menu
Enter command:
4
Nomor makanan tidak terdapat, silakan masukan input lagi
=====
=====   Buy   =====
=====
1. Ayam mentah (15 menit)
2. Tepung (30 menit)
3. Minyak Goreng (1 jam)
Masukkan 0 untuk kembali ke Menu
Enter command:

```

Gambar 6.6.3 Tampilan ketika makanan yang ingin dibeli, tidak ada dalam pilihan

```

Enter Command: BUY
=====
=====   Buy   =====
=====
1. Ayam mentah (15 menit)
2. Tepung (30 menit)
3. Minyak Goreng (1 jam)
Masukkan 0 untuk kembali ke Menu
Enter command:
0

Chi di posisi : (2,4)
Waktu : 0.1
Notifikasi:
-

```

Gambar 6.6.4 Tampilan ketika pengguna memilih 0

## 6.7 Data Test Chop

Pemain dapat menggunakan command "CHOP" untuk beberapa makanan yang memiliki aksi "CHOP". Aksi "CHOP" hanya dapat dilakukan ketika pemain berada di sekitar area yang memiliki tanda "C" di peta. Jika pemain tidak ada di sekitar area CHOP (C), program akan menampilkan pesan kesalahan (Gambar 6.7.1). Jika pemain berada di posisi yang valid, program akan menampilkan list makanan – makanan yang bisa dibuat dengan command CHOP dan meminta pemain untuk memilih nomor makanan. Jika pemain memilih makanan yang tidak terdapat dalam list makanan, pemain akan diminta memasukkan input lain (Gambar 6.7.2). Jika pemain memilih 0, program akan kembali ke menu utama (Gambar 6.7.3). Jika masukan pemain valid, program akan mengecek apakah pemain memiliki bahan yang dibutuhkan. Jika bahan-bahan yang dibutuhkan ada di inventory, aksi CHOP akan berhasil (Gambar 6.7.4), dan jika tidak, aksi CHOP akan gagal (Gambar 6.7.5).

```
Enter Command: chop
BNMO tidak berada di area Chop

Nathania di posisi : (4,8)
Waktu : 0.11
Notifikasi:
-
* * * * *
*           T       X       *
*      M           X       *
*           X       X       *
*      X X X X S       *
*      X           C       *
*      X           F       *
*           B           *
* * * * *
Enter Command: █
```

Gambar 6.7.1 Tampilan ketika pemain tidak berada dalam area CHOP

```
Enter Command: CHOP
=====
=====   Chop   =====
=====
1. Ayam Potong (10 menit)
Masukkan 0 untuk kembali ke Menu
Enter command :
2
Nomor makanan tidak terdapat, silakan masukan input lagi
=====
=====   Buy   =====
=====
1. Ayam Potong (10 menit)
Masukkan 0 untuk kembali ke Menu
Enter command:
█
```

Gambar 6.7.2 Tampilan ketika aksi CHOP tidak bisa dilakukan karena input tidak valid

```

Enter Command: chop
=====
=====  Chop  =====
=====
1. Ayam Potong (10 menit)
Masukkan 0 untuk kembali ke Menu
Enter command :
0

```

Gambar 6.7.3 Tampilan ketika pengguna memilih 0

```

* * * * *
*           T       X       *
*   M           X       *
*           X       *
*       X X X X       *
*   X           S       *
*   X           C       *
*   X X X       F       *
*           B       *
* * * * *
Enter Command: CHOP
=====
=====  Chop  =====
=====
1. Ayam Potong (10 menit)
Masukkan 0 untuk kembali ke Menu
Enter command :
1
Ayam Potong selesai dibuat dan sudah dimasukkan ke inventory

```

Gambar 6.7.4 Tampilan ketika aksi CHOP berhasil dilakukan

```

* * * * *
*           T       X       *
*   M           X       *
*           X       *
*       X X X X       *
*   X           S       *
*   X           C       *
*   X X X       F       *
*           B       *
* * * * *
Enter Command: CHOP
=====
=====  Chop  =====
=====
1. Ayam Potong (10 menit)
Masukkan 0 untuk kembali ke Menu
Enter command :
1
Gagal membuat Ayam Potong karena kamu tidak memiliki bahan berikut :
1.Ayam mentah

```

Gambar 6.7.5 Tampilan ketika pemain gagal CHOP karena bahan yang dibutuhkan tidak ada

## 6.8 Data Test Mix

Pemain dapat menggunakan command MIX untuk beberapa makanan yang memiliki aksi MIX. Aksi MIX hanya dapat dilakukan ketika pemain berada di sekitar area yang memiliki tanda "M" di peta. Jika pemain tidak ada di sekitar area MIX (M), program akan menampilkan pesan kesalahan (Gambar 6.8.1). Jika pemain berada di posisi yang valid, program akan menampilkan list makanan – makanan yang bisa dibuat dengan command MIX dan meminta pemain untuk

STEI- ITB	IF2110-TB-H-2	Halaman 21 dari 43 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.		

memilih nomor makanan. Jika pemain memilih makanan yang tidak terdapat dalam list makanan, pemain akan diminta memasukkan input lain (Gambar 6.8.2). Jika pemain memilih 0, program akan kembali ke menu utama (Gambar 6.8.3). Jika masukan pemain valid, program akan mengecek apakah pemain memiliki bahan yang dibutuhkan. Jika bahan-bahan yang dibutuhkan ada di inventory, aksi MIX akan berhasil (Gambar 6.8.4), dan jika tidak, aksi MIX akan gagal (Gambar 6.8.5).

```
Nathania di posisi : (4,1)
Waktu : 1.23
Notifikasi:
-
* * * * *
*           T   X   *
*   M           X   *
*           X     *
*   S   X X X   *
*   X           *
*   X           C   *
*   X X X   F     *
*           B     *
* * * * *
Enter Command: MIX
BNMO tidak berada di area Mix
```

Gambar 6.8.1 Tampilan ketika pemain tidak berada dalam area MIX

```
Enter Command: MIX
=====
===== Mix =====
=====
1. Ayam Tepung (15 menit)
Masukkan 0 untuk kembali ke Menu
Enter command:
2
Nomor makanan tidak terdapat, silakan masukan input lagi
=====
===== Buy =====
=====
1. Ayam Tepung (15 menit)
Masukkan 0 untuk kembali ke Menu
Enter command:
```

Gambar 6.8.2 Tampilan ketika aksi MIX tidak bisa dilakukan karena input tidak valid

```
Enter Command: MIX
=====
===== Mix =====
=====
1. Ayam Tepung (15 menit)
Masukkan 0 untuk kembali ke Menu
Enter command:
0
```

Gambar 6.8.3 Tampilan ketika pengguna memilih 0

```

Enter Command: MIX
=====
=====   Mix   =====
=====
1. Ayam Tepung (15 menit)
Masukkan 0 untuk kembali ke Menu
Enter command:
1
Ayam Tepung selesai dibuat dan sudah dimasukkan ke inventory

```

Gambar 6.8.4 Tampilan ketika aksi MIX berhasil dilakukan

```

Enter Command: mix
=====
=====   Mix   =====
=====
1. Ayam Tepung (15 menit)
Masukkan 0 untuk kembali ke Menu
Enter command:
1
Gagal membuat Ayam Tepung karena kamu tidak memiliki bahan berikut :
1. Ayam Potong
2. Tepung

```

Gambar 6.8.5 Tampilan ketika pemain gagal MIX karena bahan yang dibutuhkan tidak ada

## 6.9 Data Test Fry

Pemain dapat menggunakan command FRY untuk beberapa makanan yang memiliki aksi FRY. Aksi FRY hanya dapat dilakukan ketika pemain berada di sekitar area yang memiliki tanda "F" di peta. Jika pemain tidak ada di sekitar area FRY (F), program akan menampilkan pesan kesalahan (Gambar 6.9.1). Jika pemain berada di posisi yang valid, program akan menampilkan list makanan – makanan yang bisa dibuat dengan command FRY dan meminta pemain untuk memilih nomor makanan. Jika pemain memilih makanan yang tidak terdapat dalam list makanan, pemain akan diminta memasukkan input lain (Gambar 6.9.2). Jika pemain memilih 0, program akan kembali ke menu utama (Gambar 6.9.3). Jika masukan pemain valid, program akan mengecek apakah pemain memiliki bahan yang dibutuhkan. Jika bahan-bahan yang dibutuhkan ada di inventory, aksi FRY akan berhasil (Gambar 6.9.4), dan jika tidak, aksi FRY akan gagal (Gambar 6.9.5).

```

CHI di posisi : (2,5)
Waktu : 0.0
Notifikasi:
-
* * * * *
*           T       X       *
*      M       S       X       *
*           X       X       *
*      X X X X       *
*      X           C       *
*      X X X       F       *
*           B       *
* * * * *
Enter Command: FRY
BNMO tidak berada di area Fry

```

Gambar 6.9.1 Tampilan ketika pemain tidak berada dalam area FRY

```

Enter Command: FRY
=====
===== Fry =====
=====
1. Ayam Goreng (30 jam)
Masukkan 0 untuk kembali ke Menu
Enter command:
3
Nomor makanan tidak terdapat, silakan masukan input lagi
=====
===== Buy =====
=====
1. Ayam Goreng (30 jam)
Masukkan 0 untuk kembali ke Menu
Enter command:

```

Gambar 6.9.2 Tampilan ketika aksi FRY tidak bisa dilakukan karena input tidak valid

```

Enter Command: FRY
=====
===== Fry =====
=====
1. Ayam Goreng (30 jam)
Masukkan 0 untuk kembali ke Menu
Enter command:
0

CHI di posisi : (6,6)
Waktu : 0.44
Notifikasi:
-

```

Gambar 6.9.3 Tampilan ketika pengguna memilih 0



```

Enter Command: FRY
=====
===== Fry =====
=====
1. Ayam Goreng (30 jam)
Masukkan 0 untuk kembali ke Menu
Enter command:
1
Ayam Goreng selesai dibuat dan sudah dimasukkan ke inventory

```

Gambar 6.9.4 Tampilan ketika aksi FRY berhasil dilakukan

```

Enter Command: FRY
=====
===== Fry =====
=====
1. Ayam Goreng (30 jam)
Masukkan 0 untuk kembali ke Menu
Enter command:
1

Gagal membuat Ayam Goreng karena kamu tidak memiliki bahan berikut :
1. Ayam Tepung
2. Minyak Goreng

```

Gambar 6.9.5 Tampilan ketika pemain gagal FRY karena bahan yang dibutuhkan tidak ada

## 6.10 Data Test Boil

Pemain dapat menggunakan command BOIL untuk beberapa makanan yang memiliki aksi BOIL. Aksi BOIL hanya dapat dilakukan ketika pemain berada di sekitar area yang memiliki tanda "B" di peta. Jika pemain tidak ada di sekitar area BOIL (B), program akan menampilkan pesan kesalahan (Gambar 6.10.1). Jika pemain berada di posisi yang valid, program akan menampilkan list makanan – makanan yang bisa dibuat dengan command BOIL dan meminta pemain untuk memilih nomor makanan. Jika pemain memilih makanan yang tidak terdapat dalam list makanan, pemain akan diminta memasukkan input lain (Gambar 6.10.2). Jika pemain memilih 0, program akan kembali ke menu utama (Gambar 6.10.3). Jika masukan pemain valid, program akan mengecek apakah pemain memiliki bahan yang dibutuhkan. Jika bahan-bahan yang dibutuhkan ada di inventory, aksi BOIL akan berhasil (Gambar 6.10.4), dan jika tidak, aksi BOIL akan gagal (Gambar 6.10.5).

```

CHI di posisi : (2,5)
Waktu : 0.0
Notifikasi:
-
* * * * *
*           T       X       *
*      M       S     X       *
*           X X X X       *
*      X                   *
*      X                   C  *
*    X X X       F         *
*                   B         *
* * * * *
Enter Command: BOIL
BNMO tidak berada di area BOIL

```

Gambar 6.10.1 Tampilan ketika pemain tidak berada dalam area BOIL

```

Enter Command: BOIL
=====
===== Boil =====
=====
1. Telur Rebus (1 jam)
Masukkan 0 untuk kembali ke Menu
Enter command:
2
Nomor makanan tidak terdapat, silakan masukan input lagi

```

Gambar 6.10.2 Tampilan ketika aksi BOIL tidak bisa dilakukan karena input tidak valid

```

Enter Command: BOIL
=====
===== Boil =====
=====
1. Telur Rebus (1 jam)
Masukkan 0 untuk kembali ke Menu
Enter command:
0

```

Gambar 6.10.3 Tampilan ketika pengguna memilih 0

```

Enter Command: BOIL
=====
===== Boil =====
=====
1. Telur Rebus (1 jam)
Masukkan 0 untuk kembali ke Menu
Enter command:
1
Telur Rebus selesai dibuat dan sudah dimasukkan ke inventory

```

Gambar 6.10.4 Tampilan ketika aksi BOIL berhasil dilakukan

```

Enter Command: BOIL
=====
===== Boil =====
=====
1. Telur Rebus (1 jam)
Masukkan 0 untuk kembali ke Menu
Enter command:
1

Gagal membuat Telur Rebus karena kamu tidak memiliki bahan berikut :
1.Telur

```

Gambar 6.10.5 Tampilan ketika pemain gagal BOIL karena bahan yang dibutuhkan tidak ada

## 6.11 Data Test Delivery

Pemain dapat melihat makanan yang sedang berada dalam perjalanan dengan mengetikkan command "DELIVERY". Command ini tidak akan menambah waktu permainan. Delivery queue bisa kosong (Gambar 6.11.1) ataupun berisi (Gambar 6.11.2).

```

*****
Enter Command: DELIVERY
Tidak ada makanan di perjalanan

```

Gambar 6.11.1 Tampilan ketika tidak ada makanan yang berada dalam perjalanan

```

*****
Enter Command: delivery
List Makanan di Perjalanan
(nama - waktu sisa delivery)
1. Ayam mentah - 13 menit
2. Tepung - 29 menit

Nathania di posisi : (2,4)
Waktu : 0.3
Notifikasi:
-

```

Gambar 6.11.2 Tampilan ketika delivery queue tidak kosong

## 6.12 Data Test Inventory

Pemain dapat memasukkan command "INVENTORY" untuk menampilkan isi inventory pemain. Command ini tidak akan menambah waktu permainan. Inventory bisa kosong (Gambar 6.12.1) ataupun berisi (Gambar 6.12.2).

```

Enter Command: INVENTORY
Tidak ada makanan di inventory

```

Gambar 6.12.1 Tampilan inventory kosong

```

Enter Command: INVENTORY
List Makanan di Inventory
(nama - waktu sisa kedaluarsa)
1. Tepung - 28 menit
2. Minyak Goreng - 1 jam 29 menit
3. Ayam mentah - 23 jam 12 menit

```

Gambar 6.12.2 Tampilan inventory tidak kosong

### 6.13 Data Test Wait

Pemain dapat menjalankan command “WAIT x y” dengan x adalah lama jam dan y adalah lama menit. Waktu pada game akan otomatis bertambah selama x jam dan y menit ketika pemain menjalankan command ini (Gambar 6.13.1, Gambar 6.13.2).

```
Enter Command: DELIVERY
List Makanan di Perjalanan
(nama - waktu sisa delivery)
1. Ayam mentah - 14 menit

JUP di posisi : (2,4)
Waktu : 0.2
Notifikasi:
-
* * * * *
*           T       X       *
*   M       S       X       *
*           X       X       *
*         X X X X       *
*   X                               *
*   X                               C *
*   X X X       F       *
*                               *
*           B       *
* * * * *
Enter Command: WAIT 0 13

JUP di posisi : (2,4)
Waktu : 0.15
Notifikasi:
-
* * * * *
*           T       X       *
*   M       S       X       *
*           X       X       *
*         X X X X       *
*   X                               *
*   X                               C *
*   X X X       F       *
*                               *
*           B       *
* * * * *
Enter Command: DELIVERY
List Makanan di Perjalanan
(nama - waktu sisa delivery)
1. Ayam mentah - 1 menit
```

Gambar 6.13.1 Tampilan ketika pemain menjalankan command WAIT 0 13

```

Hanif di posisi : (3,5)
Waktu : 0.1
Notifikasi:
-
* * * * *
*           T       X       *
*      M           X       *
*           S       X       *
*         X X X X       *
*      X                   *
*      X                   C  *
*    X X X       F         *
*                   B         *
* * * * *
Enter Command: WAIT 1 59

Hanif di posisi : (3,5)
Waktu : 2.0
Notifikasi:
-

```

Gambar 6.13.2 Tampilan ketika pemain menjalankan command WAIT 1 59

## 6.14 Data Test Insert Kulkas

Pemain dapat memasukkan command “INSERT KULKAS” untuk memasukkan makanan dari inventory ke dalam kulkas. Command ini akan menambah waktu permainan sebanyak 1 menit. Jika tidak ada makanan di inventory, program akan menampilkan pesan bahwa inventory kosong (Gambar 6.14.1). Jika inventory tidak kosong, program akan menampilkan isi inventory dan meminta pilihan nomor makanan yang akan diambil (Gambar 6.14.2). Jika masukan nomor pengguna tidak valid, program akan menampilkan pesan kesalahan dan meminta masukan lagi (Gambar 6.14.3). Jika pengguna memberi masukan yang valid, makanan tersebut berhasil dikeluarkan dari kulkas (Gambar 6.14.4). Jika pengguna memilih 0, program akan kembali ke menu utama (Gambar 6.14.5).

```

Enter Command: INSERT KULKAS
Inventory kosong, tidak ada makanan yang bisa
dimasukkan ke kulkas.

```

Gambar 6.14.1 Tampilan saat inventory kosong

```

Enter Command: INSERT KULKAS
List Makanan di Inventory
(nama - waktu sisa kedaluarsa)
1. Tepung - 29 menit
2. Minyak Goreng - 1 jam 30 menit
3. Ayam mentah - 23 jam 13 menit

Masukkan 0 untuk kembali ke menu.
Masukkan nomor makanan di inventory: 4

Nomor makanan tidak terdapat, silakan masukan input lagi

Masukkan 0 untuk kembali ke menu.
Masukkan nomor makanan di inventory: 1
Makanan berhasil dimasukkan ke dalam kulkas.

```

Gambar 6.14.2 Tampilan ketika inventory tidak kosong dan pemain memilih nomor yang tidak valid

```

Enter Command: INSERT KULKAS
List Makanan di Inventory
(nama - waktu sisa kedaluarsa)
1. Ayam mentah - 23 jam 59 menit

Masukkan 0 untuk kembali ke menu.
Masukkan nomor makanan di inventory: 1
Makanan berhasil dimasukkan ke dalam kulkas.

```

Gambar 6.14.3 Tampilan ketika inventory tidak kosong dan pemain memilih nomor yang valid

```

Enter Command: INSERT KULKAS
List Makanan di Inventory
(nama - waktu sisa kedaluarsa)
1. Ayam mentah - 23 jam 57 menit

Masukkan 0 untuk kembali ke menu.
Masukkan nomor makanan di inventory: 0

Chi di posisi : (2,4)
Waktu : 1.11
Notifikasi:
-

```

Gambar 6.14.4 Tampilan ketika inventory tidak kosong dan pemain memilih nomor 0

```

Enter Command: INSERT KULKAS
List Makanan di Inventory
(nama - waktu sisa kedaluarsa)
1. Ayam mentah - 23 jam 57 menit

Masukkan 0 untuk kembali ke menu.
Masukkan nomor makanan di inventory: 1
Makanan gagal dimasukkan ke dalam kulkas karena kulkas penuh.

```

Gambar 6.14.5 Tampilan ketika kulkas sudah penuh

## 6.15 Data Test Take Kulkas

Pemain dapat memasukkan command “TAKE KULKAS” untuk mengambil makanan dari kulkas dan menaruhnya kembali di inventory. Command ini akan menambah waktu permainan sebanyak 1 menit. Jika tidak ada makanan di kulkas, program akan menampilkan pesan bahwa

STEI- ITB	IF2110-TB-H-2	Halaman 30 dari 43 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.		

```
Enter Command: TAKE KULKAS
Kulkas kosong.
```

```
Enter Command: TAKE KULKAS
1 1 1 1 1 * * * * * * * * * * * * * * *
1 1 1 1 1 * * * * * * * * * * * * * * *
1 1 1 1 1 * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * *
```

Isi Kulkas (Sorted dari ukuran terbesar):  
(nomor. nama - id - waktu kedaluarsa)  
1. Ayam mentah - 11 - 1 hari

Masukkan 0 untuk kembali ke menu.  
Masukkan nomor makanan di kulkas: 3

Nomor makanan tidak terdapat, silakan masukan input lagi

Masukkan 0 untuk kembali ke menu.  
Masukkan nomor makanan di kulkas:

```
Enter Command: TAKE KULKAS
1 1 1 1 1 * * * * * * * * * * * * * * * *
1 1 1 1 1 * * * * * * * * * * * * * * * *
1 1 1 1 1 * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * *
```

Isi Kulkas (Sorted dari ukuran terbesar):  
(nomor. nama - id - waktu kedaluarsa)  
1. Ayam mentah - 11 - 1 hari

Masukkan 0 untuk kembali ke menu.  
Masukkan nomor makanan di kulkas: 1  
Makanan berhasil dihapus dari kulkas.

STEI- ITB	IF2110-TB-H-2	Halaman 31 dari 43 halaman
<p>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</p>		



```
Enter Command: TAKE KULKAS  
1 1 1 1 1 * * * * *  
1 1 1 1 1 * * * * *  
1 1 1 1 1 * * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *
```

Isi Kulkas (Sorted dari ukuran terbesar):  
(nomor. nama - id - waktu kedaluarsa)  
1. Ayam mentah - 11 - 1 hari

Masukkan 0 untuk kembali ke menu.  
Masukkan nomor makanan di kulkas: 0

CHI di posisi : (2,4)  
Waktu : 0.18  
Notifikasi:

Gambar 6.15.4 Tampilan ketika kulkas tidak kosong dan pemain memilih nomor 0

### 6.16 Data Test Kulkas

Pemain dapat memasukkan command “KULKAS” untuk menampilkan isi kulkas pemain. Kulkas bisa kosong (Gambar 6.16.1) ataupun berisi (Gambar 6.16.2).

[illegible]

Gambar 6.16.1 Tampilan ketika kulkas kosong



```

Enter Command: KULKAS

1 1 1 1 1 2 2 * * * * *
1 1 1 1 1 3 3 * * * * *
1 1 1 1 1 3 3 * * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

Isi Kulkas (Sorted dari ukuran terbesar):
(nomor. nama - id - waktu kedaluarsa)
1. Ayam mentah - 11 - 23 jam 11 menit
2. Tepung - 10 - 29 menit
3. Minyak Goreng - 15 - 1 jam 29 menit

```

Gambar 6.16.2 Tampilan ketika kulkas tidak kosong

## 6.17 Data Test Rekomendasi

Pemain dapat memasukkan command “RECOMM” untuk menampilkan rekomendasi resep berdasarkan inventory pemain. Rekomendasi yang ditampilkan bisa kosong (Gambar 6.17.1) ataupun berisi (Gambar 6.17.2).

```

Enter Command: RECOMM
Tidak ada rekomendasi karena inventory kurang/tidak lengkap.

```

Gambar 6.17.1 Tampilan ketika tidak ada rekomendasi

```

Enter Command: RECOMM
Rekomendasi:
1. Ayam Potong
2. Ayam Tepung
3. Ayam Goreng

```

Gambar 6.17.2 Tampilan ketika ada rekomendasi

## 6.18 Data Test Undo

Pemain dapat memasukkan command “UNDO” untuk kembali ke state permainan sebelumnya (sebelum command yang berpengaruh terhadap waktu dijalankan). Command bisa tidak valid karena program sudah di state awal (Gambar 6.18.1) atau berhasil (Gambar 6.18.2).

```

Enter Command: UNDO
BNMO sudah berada pada keadaan semula, tidak bisa UNDO!

```

Gambar 6.18.1 Command UNDO yang gagal

```

Enter Command: UNDO

CHI di posisi : (2,4)
Waktu : 0.17
Notifikasi:
Aksi WAIT 24 0 dibatalkan
Ayam mentah dikembalikan ke inventory
* * * * *
*           T       X       *
*   M       S       X       *
*           X       *
*           X X X X       *
*   X                               *
*   X                               C   *
*   X X X       F                               *
*                                           *
*           B                               *
* * * * *

```

Gambar 6.18.2 Command UNDO yang berhasil

## 6.19 Data Test Redo

Pemain dapat memasukkan command “REDO” untuk kembali ke state program sebelum di-UNDO. Command bisa tidak valid karena tidak ada aksi yang bisa di-REDO (Gambar 6.19.1) atau berhasil (Gambar 6.19.2).

```

Enter Command: REDO
Tidak ada command yang bisa di-redo!

```

Gambar 6.19.1 Command REDO yang gagal

```

Enter Command: REDO

CHI di posisi : (2,4)
Waktu : 0.17
Notifikasi:
Aksi WAIT 24 0 dilakukan ulang
Ayam mentah kedaluarsa
* * * * *
*           T       X       *
*   M       S       X       *
*           X       *
*           X X X X       *
*   X                               *
*   X                               C   *
*   X X X       F                               *
*                                           *
*           B                               *
* * * * *

```

Gambar 6.19.2 Command REDO yang berhasil

## 6.20 Data Test Notifikasi

Notifikasi selalu dimunculkan di menu utama. Pemain tidak perlu mengetikkan command apapun untuk melihat notifikasi. Notifikasi dapat kosong (Gambar 6.20.1), menginformasikan makanan yang masuk ke inventory (Gambar 6.20.2), menginformasikan makanan yang kedaluarsa (Gambar 6.20.3), menginformasikan keberhasilan command UNDO (Gambar 6.20.4), dan menginformasikan keberhasilan command REDO (Gambar 6.20.5).

```
CHI di posisi : (2,4)
Waktu : 0.2
Notifikasi:
-
```

Gambar 6.20.1 Notifikasi kosong

```
CHI di posisi : (2,4)
Waktu : 0.17
Notifikasi:
Ayam mentah sudah sampai dari perjalanan
```

Gambar 6.20.2 Notifikasi makanan masuk ke inventory

```
CHI di posisi : (2,4)
Waktu : 0.17
Notifikasi:
Ayam mentah kedaluarsa
```

Gambar 6.20.3 Notifikasi makanan kedaluarsa

```
CHI di posisi : (2,4)
Waktu : 0.17
Notifikasi:
Aksi WAIT 24 0 dibatalkan
Ayam mentah dikembalikan ke inventory
```

Gambar 6.20.4 Notifikasi keberhasilan command UNDO

```
CHI di posisi : (2,4)
Waktu : 0.17
Notifikasi:
Aksi WAIT 24 0 dilakukan ulang
Ayam mentah kedaluarsa
```

Gambar 6.20.5 Notifikasi keberhasilan command REDO

## 6.21 Data Test Help

Pemain dapat memasukkan command “HELP” untuk melihat list command yang valid (Gambar 6.21.1).



No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
5	Move	Memeriksa apakah simulator bergerak sesuai dengan arah yang ditentukan oleh pengguna. Memeriksa kasus ketika input pengguna mengarahkan simulator ke tempat yang tidak bisa dilewati.	Saat permainan sudah dimulai, masukkan command MOVE <arah>. Arah yang memungkinkan ada 4 jenis, yaitu NORTH, SOUTH, EAST, dan WEST.	MOVE NORTH  MOVE SOUTH  MOVE EAST  MOVE WEST	Jika tempat yang dituju dapat dilewati simulator, simulator akan pindah ke tempat tersebut, kemudian waktu game akan bertambah 1 menit.  Jika tempat yang dituju oleh simulator tidak bisa dilewati, maka akan muncul tampilan bahwa simulator tidak dapat berpindah ke tempat tujuan, waktu tidak bertambah.	Gambar 6.5.1  Gambar 6.5.2  Gambar 6.5.3  Gambar 6.5.4  Gambar 6.5.5  Gambar 6.5.6  Gambar 6.5.7
6	Buy	Memeriksa apakah program dapat melakukan pemesanan bahan makanan	Ketik BUY ketika permainan sudah dimulai. Program akan menampilkan bahan makanan yang dapat dipesan beserta waktu delivery nya. Kemudian pilih bahan yang akan dipesan dengan cara menginputkan nomor sesuai dengan tampilan.	BUY	Jika simulator berada dekat dengan 'T' pada peta, aksi BUY dapat dilakukan, kemudian akan muncul list makanan yang dapat dibeli.  Jika simulator tidak berada dekat dengan 'T' pada peta, program akan menampilkan keterangan bahwa simulator tidak berada di area telepon.  Aksi BUY tidak akan dilakukan dan kembali ke menu utama ketika pengguna memasukkan 0 ketika pemilihan bahan.	Gambar 6.6.1  Gambar 6.6.2  Gambar 6.6.3  Gambar 6.6.4
7	Chop	Memeriksa apakah program dapat melakukan aksi CHOP yaitu memotong satu bahan makanan menjadi bahan yang sudah terpotong	Ketik CHOP ketika permainan sudah dimulai. Program akan menampilkan bahan makanan yang dapat dipotong beserta waktu delivery nya. Kemudian pilih bahan yang akan dipotong dengan cara menginputkan nomor sesuai dengan tampilan.	CHOP	Jika simulator berada dekat dengan 'C' pada peta, aksi CHOP dapat dilakukan kemudian bahan yang dibuat akan ditambahkan ke inventory dan bahan yang digunakan akan dihapus dari inventory. Aksi CHOP akan gagal ketika bahan makanan yang diperlukan tidak lengkap/tidak ada.  Jika simulator tidak berada dekat dengan 'C' pada peta, program akan menampilkan keterangan bahwa aksi CHOP tidak dapat dilakukan.  Aksi CHOP tidak akan dilakukan dan kembali ke menu utama ketika pengguna memasukkan 0 ketika pemilihan bahan.	Gambar 6.7.1  Gambar 6.7.2  Gambar 6.7.3  Gambar 6.7.4  Gambar 6.7.5

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
8	Mix	Memeriksa apakah program dapat melakukan aksi MIX dari bahan makanan yang ada di dalam inventory simulator menjadi bahan makanan yang baru kemudian tersimpan di inventory	Ketik MIX ketika permainan sudah dimulai. Program akan menampilkan bahan makanan yang dapat dibuat berdasarkan resep. Kemudian pilih bahan makanan yang akan dibuat dengan memasukkan angka sesuai tampilan.	MIX	<p>Jika simulator berada dekat dengan 'M' pada peta, aksi MIX dapat dilakukan, kemudian bahan yang dibuat akan ditambahkan ke inventory dan bahan yang digunakan akan dihapus dari inventory. Aksi MIX akan gagal ketika bahan makanan yang diperlukan tidak lengkap/tidak ada.</p> <p>Jika simulator tidak berada dekat dengan 'M' pada peta, program akan menampilkan keterangan bahwa aksi MIX tidak dapat dilakukan.</p> <p>Aksi MIX tidak akan dilakukan dan kembali ke menu utama ketika pengguna memasukkan 0 ketika pemilihan bahan.</p>	<p>Gambar 6.8.1</p> <p>Gambar 6.8.2</p> <p>Gambar 6.8.3</p> <p>Gambar 6.8.4</p> <p>Gambar 6.8.5</p>
9	Fry	Memeriksa apakah program dapat melakukan aksi FRY yaitu menggoreng satu atau lebih bahan makanan.	Ketik FRY ketika permainan sudah dimulai. Program akan menampilkan bahan makanan yang dapat digoreng beserta waktu delivery nya. Kemudian pilih bahan yang akan digoreng dengan cara menginputkan nomor sesuai dengan tampilan.	FRY	<p>Jika simulator berada dekat dengan 'F' pada peta, aksi FRY dapat dilakukan, kemudian bahan yang dibuat akan ditambahkan ke inventory dan bahan yang digunakan akan dihapus dari inventory. Aksi FRY akan gagal ketika bahan makanan yang diperlukan tidak lengkap/tidak ada.</p> <p>Jika simulator tidak berada dekat dengan 'F' pada peta, program akan menampilkan keterangan bahwa simulator tidak berada di area Fry.</p> <p>Aksi FRY tidak akan dilakukan dan kembali ke menu utama ketika pengguna memasukkan 0 ketika pemilihan bahan.</p>	<p>Gambar 6.9.1</p> <p>Gambar 6.9.2</p> <p>Gambar 6.9.3</p> <p>Gambar 6.9.4</p> <p>Gambar 6.9.5</p>
10	Boil	Memeriksa apakah program dapat melakukan aksi BOIL, yaitu merebus bahan makanan	Ketik BOIL ketika permainan sudah dimulai. Program akan menampilkan bahan makanan yang dapat direbus beserta waktu delivery nya. Kemudian pilih bahan yang akan direbus dengan cara menginputkan nomor sesuai dengan tampilan.	BOIL	<p>Jika simulator berada dekat dengan 'B' pada peta, aksi BOIL dapat dilakukan, kemudian bahan yang dibuat akan ditambahkan ke inventory dan bahan yang digunakan akan dihapus dari inventory. Aksi BOIL akan gagal ketika bahan makanan yang diperlukan tidak lengkap/tidak ada.</p> <p>Jika simulator tidak berada dekat dengan 'B' pada peta, program akan menampilkan keterangan bahwa simulator tidak berada di area Boil.</p> <p>Aksi BOIL tidak akan dilakukan dan kembali ke menu utama ketika pengguna memasukkan 0 ketika pemilihan bahan.</p>	<p>Gambar 6.10.1</p> <p>Gambar 6.10.2</p> <p>Gambar 6.10.3</p> <p>Gambar 6.10.4</p> <p>Gambar 6.10.5</p>
11	Delivery	Memeriksa apakah program dapat melihat pemesanan makanan dengan command DELIVERY	Ketik DELIVERY ketika permainan sudah dimulai	DELIVERY	<p>Program menampilkan bahan makan yang sedang dalam proses delivery dengan urutan penampilan dari waktu tercepat sampai waktu terlama.</p> <p>Jika tidak ada bahan makanan yang sedang dalam proses delivery,</p>	<p>Gambar 6.11.1</p> <p>Gambar 6.11.2</p>

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
					program menampilkan keterangan bahwa tidak ada bahan yang sedang di-delivery.	
12	Inventory	Memeriksa apakah program dapat menampilkan isi inventory pemainan dengan memasukkan command INVENTORY	Ketik INVENTORY ketika permainan sudah dimulai. Program akan menampilkan bahan makanan yang dimiliki pemain.	INVENTORY	Program menampilkan isi inventory pemain beserta dengan waktu kedaluarsa setiap bahan.  Program menampilkan keterangan inventory kosong jika pemain tidak memiliki bahan apa-apa di dalam inventory	Gambar 6.11.1  Gambar 6.12.2
13	Wait	Memeriksa apakah waktu game bertambah sesuai dengan input pengguna.	Ketik WAIT <x> <y> ketika permainan sudah dimulai, dengan x adalah waktu dalam jam, dan y adalah waktu dalam menit.	WAIT 0 13  WAIT 1 59	Waktu game bertambah sebanyak 13 menit.  Waktu game bertambah sebanyak 1 jam 59 menit	Gambar 6.13.1  Gambar 6.13.2
14	Insert Kulkas	Memeriksa apakah program dapat menampilkan list makanan dari inventory yang dimasukkan ke kulkas.	Ketik INSERT KULKAS ketika permainan sudah dimulai. Program akan menampilkan bahan makanan yang dapat dimasukkan ke kulkas beserta waktu sisa kedaluarsanya. Kemudian pilih bahan yang akan dimasukkan ke kulkas dengan cara menginputkan nomor sesuai dengan tampilan.	INSERT KULKAS	Program dapat menampilkan bahan makanan yang dapat dimasukkan ke kulkas beserta waktu sisa kedaluarsanya ketika inventory tidak kosong dan pemain memilih nomor yang valid.  Aksi INSERT KULKAS akan gagal dan menampilkan keterangan tidak ada makanan yang bisa dimasukkan ke kulkas ketika bahan makanan yang diperlukan tidak tersedia di inventory atau inventory kosong.  Aksi INSERT KULKAS tidak akan dilakukan dan kembali ke menu utama ketika pengguna memasukkan 0 ketika pemilihan bahan.	Gambar 6.14.1  Gambar 6.14.2  Gambar 6.14.3  Gambar 6.14.4  Gambar 6.14.5
15	Take Kulkas	Memeriksa apakah program dapat menampilkan list makanan dari inventory yang dapat dikeluarkan dari kulkas.	Ketik TAKE KULKAS ketika permainan sudah dimulai. Program akan menampilkan bahan makanan yang dapat dikeluarkan dari kulkas beserta waktu kedaluarsanya. Kemudian pilih bahan yang akan dikeluarkan dari kulkas dengan cara menginputkan nomor sesuai dengan tampilan.	TAKE KULKAS	Program dapat menampilkan bahan makanan yang dapat dikeluarkan dari kulkas beserta waktu kedaluarsanya ketika kulkas tidak kosong dan pemain memilih nomor yang valid.  Aksi TAKE KULKAS akan gagal dan menampilkan keterangan bahwa kulkas kosong ketika bahan makanan yang akan dikeluarkan tidak ada dalam kulkas atau kulkas kosong.  Aksi TAKE KULKAS tidak akan dilakukan dan kembali ke menu utama ketika pengguna memasukkan 0 ketika pemilihan bahan.	Gambar 6.15.1  Gambar 6.15.2  Gambar 6.15.3  Gambar 6.15.4
16	Kulkas	Memeriksa apakah program dapat menampilkan isi kulkas pemain	Ketik KULKAS jika permainan sudah dimulai, maka program akan menampilkan isi kulkas dari pemain beserta ID dan waktu kedaluarsa.	KULKAS	Program dapat menampilkan isi kulkas pemain beserta ID dan waktu kedaluarsa.  Program menampilkan keterangan kulkas kosong jika pemain tidak memiliki bahan apa-apa di dalam kulkas.	Gambar 6.16.1  Gambar 6.16.2

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
17	Rekomendasi	Memeriksa makanan yang direkomendasikan sesuai dengan barang yang dimiliki di inventory	Ketik RECOMM Ketika permainan sudah dimulai, lalu program akan menampilkan rekomendasi makanan berdasarkan inventory yang dimiliki	RECOMM	Program menampilkan rekomendasi makanan yang bisa dibuat berdasarkan inventory yang dimiliki	Gambar 6.17.1  Gambar 6.17.2
18	Undo	Memeriksa apakah program dapat melakukan aksi untuk kembali ke state permainan sebelumnya.	Ketik UNDO ketika permainan sudah dimulai, jika program sudah berada di state awal maka program akan menampilkan keterangan simulator sudah berada pada keadaan semula.  Jika simulator tidak berada di state awal maka program dapat menampilkan state permainan sebelumnya.	UNDO	Program menampilkan state permainan sebelumnya, jika simulator tidak berada di state awal.  Jika simulator sudah berada di state awal permainan, maka program akan menampilkan keterangan bahwa simulator sudah berada pada keadaan semula.	Gambar 6.18.1  Gambar 6.18.2
19	Redo	Memeriksa apakah program dapat melakukan aksi untuk kembali ke state program sebelum di-UNDO.	Ketik REDO ketika permainan sudah dimulai, jika tidak ada aksi yang dapat di-REDO maka program akan menampilkan keterangan tidak ada command yang bisa di-redo.  Jika ada aksi yang dapat di-REDO maka program akan menampilkan state program sebelum di-UNDO.	REDO	Program menampilkan state permainan selanjutnya, jika terdapat aksi yang dapat di-redo.  Jika tidak ada aksi yang dapat di-redo, maka program akan menampilkan keterangan tidak ada command yang bisa di-redo.	Gambar 6.19.1  Gambar 6.19.2
20	Notifikasi	Memeriksa apakah program dapat menampilkan notifikasi sesuai dengan aksi yang dilakukan.	Ketika permainan sudah dimulai, pemain tidak perlu mengetikkan command apapun untuk melihat notifikasi. Notifikasi dapat berupa menginformasikan makanan yang masuk ke inventory, makanan yang kedaluarsa, keberhasilan command UNDO, dan keberhasilan command REDO. Notifikasi dapat kosong.	Pemain tidak perlu mengetikkan command apapun untuk melihat notifikasi.	Program dapat menampilkan notifikasi sesuai aksi yang sedang dijalankan. Jika tidak ada aksi yang dijalankan, maka notifikasi kosong. Jika ada makanan masuk ke inventory, maka muncul notifikasi makanan masuk ke inventory. Jika ada makanan yang kedaluarsa, maka muncul notifikasi makanan kedaluarsa. Jika berhasil melakukan undo, maka muncul notifikasi keberhasilan command UNDO. Jika berhasil melakukan redo, maka muncul notifikasi keberhasilan command REDO.	Gambar 6.20.1  Gambar 6.20.2  Gambar 6.20.3  Gambar 6.20.4  Gambar 6.20.5
21	Help	Memeriksa apakah program dapat menampilkan penjelasan dari	Ketik HELP, maka program akan menampilkan berbagai guide commands.	HELP	Program akan menampilkan berbagai penjelasan dari list command yang valid.	Gambar 6.21.1



No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
		list command yang valid.				

## 8 Pembagian Kerja dalam Kelompok

NIM	Nama	Pembagian Kerja
13521129	Chiquita Ahsanunnisa	<ul style="list-style-type: none"> <li>- ADT Queue dengan Pendekatan Array List Dinamik</li> <li>- Command Parser</li> <li>- Inisiasi (Splash Screen, Command START, Command EXIT)</li> <li>- Fitur notifikasi</li> <li>- Bonus: Fitur kulkas</li> <li>- Laporan</li> </ul>
13521139	Nathania Calista Djunaedi	<ul style="list-style-type: none"> <li>- ADT Sederhana (Waktu, Point, Makanan, Simulator)</li> <li>- Fitur Makanan (Membaca makanan dari file, Command CATALOG)</li> <li>- Fitur Mekanisme Waktu (Waktu bertambah seiring command yang valid)</li> <li>- ADT Stack</li> <li>- Fitur Undo Redo</li> <li>- Bonus: Fitur Waktu Pengolahan Makanan</li> <li>- Laporan</li> </ul>
13521157	Hanif Muhammad Zhafran	<ul style="list-style-type: none"> <li>- ADT Matriks</li> <li>- Fitur Peta (Membaca peta dari file, Command MOVE NORTH/EAST/SOUTH/WEST, pengecekan adjacency simulator dengan tempat perintah tertentu)</li> <li>- Laporan</li> </ul>
13521161	Ferindya Aulia Berlianty	<ul style="list-style-type: none"> <li>- ADT List Statik</li> <li>- ADT Mesin Karakter dan Mesin Kata</li> <li>- Laporan</li> </ul>
13521163	Zidane Firzatullah	<ul style="list-style-type: none"> <li>- ADT Tree</li> <li>- Fitur Resep</li> <li>- Bonus: Fitur Rekomendasi Makanan</li> <li>- Laporan</li> </ul>

## 9 Lampiran

### 9.1 Deskripsi Tugas Besar 2

BNMO tidak tahu cara memasak dan Doni tidak bisa membantu persiapan karena ada hal lain. BNMO tidak bisa belajar dari video *youcub* karena BNMO adalah sebuah komputer sehingga hal yang paling mudah untuk dilakukan adalah membuat program simulasi untuk ditiru BNMO. Oleh karena itu, Doni meminta bantuan kalian untuk membuat program simulasi tersebut.

Buatlah sebuah program simulasi berbasis CLI (*command-line interface*). Program ini dibuat dalam bahasa C dengan menggunakan struktur data yang sudah kalian pelajari di mata kuliah ini. Kalian boleh menggunakan struktur data yang sudah kalian buat untuk praktikum pada tugas besar ini.

## 9.2 Notulen Rapat

Hari, Tanggal	NIM	Notulensi
Minggu, 23 Oktober 2022	13521129 13521139 13521157 13521161 13521163	<ul style="list-style-type: none"> <li>- Gambaran besar program yang akan dikerjakan</li> <li>- Rencana asistensi 1</li> <li>- Timeline pengerjaan</li> <li>- Pembagian tugas minggu pertama</li> <li>- Struktur folder</li> <li>- Code convention</li> </ul>
Senin, 31 Oktober 2022	13521129 13521139 13521157 13521161 13521163	<ul style="list-style-type: none"> <li>- Progress report pengerjaan minggu 1</li> <li>- Kendala pengerjaan tugas besar</li> <li>- Rencana minggu 2</li> </ul>
Jumat, 18 November 2022	13521129 13521139 13521157 13521161 13521163	<ul style="list-style-type: none"> <li>- Last finishing</li> <li>- Laporan</li> <li>- Evaluasi dan testing</li> </ul>

## 9.3 Log Activity Anggota Kelompok

Nama	NIM	Tanggal	Activity
Chiquita Ahsanunnisa	13521129	21 Oktober 2022	GitHub: Initial commit Laporan: Initial preparation (template, commenting)
		28 Oktober 2022	GitHub: Create ADT PriorityQueue, Fix typos & add time functions, Create parser & input-output handling, Update main
		29 Oktober 2022	GitHub: Add word handling functions, Fix ADT PriorityQueue errors, Add ADT PriorityQueue driver
		31 Oktober 2022	Laporan: Struktur Data (ADT)
		1 November 2022	GitHub: Create splash screen, Update PriorityQueue, Fix EXIT and Add HELP command Laporan: Struktur Data (ADT), Pembagian Kerja Kelompok, Program Utama, Lampiran
		4 November 2022	GitHub: Fix conflicting commit, Add notif feature, Config notif to main
		13 November 2022	GitHub: Rapiin output, Hapus driver yang salah, Fix parser error
		17 November 2022	GitHub: Create draft for bonus kulkas
		18 November 2022	GitHub: Config kulkas, Fix notif & stack bugs
		19 November 2022	GitHub: Fix take kulkas error, Delete useless files, Fix small bugs in kulkas, Fix conflict in main Laporan: Algoritma Menarik, Data test, Struktur Data (ADT), Lampiran
Nathania Calista Djunaedi	13521139	24 Oktober 2022	GitHub: Create ADT Time, ADT Point
		26 Oktober 2022	GitHub: Adding makanan.h & konfigurasi makanan, Commenting, Fix typos
		27 Oktober 2022	GitHub: Configure main, Update main
		28 Oktober 2022	GitHub: Change POINT, Add buy command in main
		29 Oktober 2022	GitHub: Fix buy, Fix buy in main, Fix conflicting commit, Add wait in buy
		31 Oktober 2022	GitHub: Add tree to fry and chop command, Merapikan kode





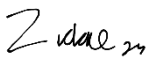

		1 November 2022	GitHub: Add validasi posisi untuk semua command makanan, Create ADT Stack & Undo-Redo Feature, Fix bugs on Undo-Redo Feature Laporan: Struktur Data (ADT), Lampiran
		15 November 2022	GitHub : Add bonus 2
		19 November 2022	GitHub : Fixing typos and adding comments, fixing bugs in config, adding comments, fixing file's structure, update README Laporan : Data test, Struktur Data (ADT), Lampiran
Hanif Muhammad Zhafran	13521157	31 Oktober 2022	GitHub: Create ADT Matrix, Add Config Map, MOVE Command, menyatukan sistem peta dengan main, pengecekan adjacency dari Simulator dengan tempat perintah tertentu.
		11 November 2022	GitHub: Create ADT Matrix driver, Map Config driver, mengubah TulisPOINT menjadi (Ordinat, Absis).
		19 November 2022	Laporan: Data test, Test script, lampiran
Ferindya Aulia Berlianty	13521161	25 Oktober 2022	GitHub: Create ADT List Statik, ADT Mesin Kata
		27 Oktober 2022	GitHub: Update main
		30 Oktober 2022	Laporan : Struktur Data (ADT), Ringkasan, Lampiran
		19 November 2022	GitHub: Update README, Create ADT Charmachine driver, Creare Wordmachine driver Laporan : Ringkasan, Struktur Data (ADT), Test script
Zidane Firzatullah	13521163	23 Oktober 2022	GitHub: Create dump.txt (untuk membuat struktur folder)
		29 Oktober 2022	GitHub: Create ADT Tree, Create Object Union
		30 Oktober 2022	GitHub: Update tree code, Add config resep for testing, Update parameters for tree functions, Update tree driver, Update tree config loader, Fix segmentation fault & parsing error in tree and config
		19 November 2022	Github: update all tree files Laporan

**Form Asistensi Tugas Besar  
IF2110/Algoritma dan Struktur Data  
Sem. 1 2021/2022**







No. Kelompok/Kelas : H/2  
Nama Kelompok : IBM  
Anggota Kelompok (Nama/NIM) :  
1. Chiquita Ahsanunnisa/13521129  
2. Nathania Calista Djunaedi/13521139  
3. Hanif Muhammad Zhafran/13521157  
4. Ferindya Aulia Berlianty/13521161  
5. Zidane Firzatullah/13521163  
  
Asisten Pembimbing : Widya Anugrah Putra/13519105

---

# Asistensi I

<b>Tanggal : 28 Oktober 2022</b>	<b>Catatan Asistensi:</b> <ul style="list-style-type: none"> <li>- Progress pengerjaan tubes sebagian besar ADT sudah dipush di GitHub</li> <li>- ADT digunakan hanya untuk memenuhi spesifikasi.</li> <li>- Untuk parser command, validasinya cek currentWord, terus advWord, terus cek currentWord lagi dan seterusnya sampai sesuai command yang diinginkan, marknya adalah '\n'.</li> <li>- ADT list statik bersidat immutable, elemennya hanya boleh dicopy. Saat inisialisasi (config untuk list makanan dan list resep), setiap elemen input boleh diubah sebanyak 1 kali. Kapasitas list statik boleh mengikuti kapasitas saat praktikum.</li> </ul>
<b>Tempat : Zoom</b>	
<b>Kehadiran Anggota Kelompok:</b> <p>No NIM Tanda tangan</p> <p>1 13521129 </p> <p>2 13521139 </p> <p>3 13521157 </p> <p>4 13521161 </p> <p>5 13521163 </p>	
	<b>Tanda Tangan Asisten:</b> 

## Asistensi II

<b>Tanggal : 1 November 2022</b>	<b>Catatan Asistensi:</b> <ul style="list-style-type: none"><li>- Tambahin command EXIT, sama tambahin PrioQueue nya di Simulator (atau langsung sambungin ke fitur notifikasi aja)</li><li>- Selesain spek yang belum (UNDO-REDO, Notifikasi)</li></ul>
<b>Tempat : Laboratorium Programming</b>	
<b>Kehadiran Anggota Kelompok:</b> <div><div>No</div><div>NIM</div><div>Tanda tangan</div><div>1</div><div>13521129</div><div></div><div>2</div><div>13521139</div><div></div><div>3</div><div>13521157</div><div></div><div>4</div><div>13521161</div><div></div><div>5</div><div>13521163</div><div></div></div>	
	<b>Tanda Tangan Asisten:</b> 

## Progress Pengerjaan Tugas Besar

### Kelompok H/K2

Anggota Kelompok (Nama/NIM) :

1. Chiquita Ahsanunnisa/13521129
2. Nathania Calista Djunaedi/13521139
3. Hanif Muhammad Zhafran/13521157
4. Ferindya Aulia Berlianty/13521161
5. Zidane Firzatullah/13521163

Tanggal		01-11-2022
No	Fitur	Progress (0-100%)
1.	Command Parser	100%
2.	Inisiasi	
	a. Splash Screen	100%
	b. Command START	100%
	c. Command EXIT	100%
3.	Simulator	
	a. ADT Simulator	100%
4.	Makanan	
	a. Membaca makanan dari file	100%
	b. ADT Makanan	100%
	c. Command CATALOG	100%
5.	Peta	
	a. Membaca peta dari file	100%
	b. Command MOVE NORTH/EAST/SOUTH/WEST	100%
6.	Mekanisme Waktu	
	a. ADT Waktu	100%

	b. Waktu bertambah seiring command yang valid	100%
7.	Laporan (50%)	100%