

LAPORAN TUGAS BESAR

IF2110/Algoritma dan Struktur Data

Cooking BNMO

Dipersiapkan oleh:

Kelompok I

13521042 Kevin John Wesley Hutabarat

13521061 Alex Sander


13521089 Kenneth Ezekiel

13521092 Frankie Huang

13521094 Angela Livia Arumsari

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		<i>IF2110-TB-I-01</i>		<i>45</i>
		<i>Revisi</i>	<i>0</i>	<i>20-02-2022</i>

Daftar Isi

1 Ringkasan.....	4
2 Penjelasan Tambahan Spesifikasi Tugas	4
2.1 Inisiasi.....	4
2.2 Simulator	4
2.3 Mekanisme Waktu	4
2.4 Kulkas	4
2.5 Waktu Pengolahan Makanan	5
2.6 Rekomendasi Makanan.....	5
2.7 Undo dan Redo Stack	5
3 Struktur Data (ADT)	5
3.1 ADT Sederhana	5
3.1.1 ADT Point	5
3.1.2 ADT Time	5
3.1.3 ADT Makanan.....	6
3.1.4 ADT Simulator.....	6
3.1.5 ADT List Statik	6
3.2 ADT Matriks.....	6
3.3 ADT Mesin Karakter dan Mesin Kata.....	6
3.4 ADT Queue dengan Pendekatan Array List Dinamik	6
3.5 ADT Stack	7
3.6 ADT Tree.....	7
3.7 ADT String	7
3.8 ADT Delivery List.....	7
3.9 ADT Inventory	7
3.10 ADT List Makanan	7
3.11 ADT Peta	8
3.12 ADT Resep	8
3.13 ADT State	8
3.14 ADT State Stack	8
3.15 ADT Undo Stack	8
3.16 ADT Redo Stack.....	8
3.17 ADT Kulkas.....	8
3.18 ADT Set.....	9
3.19 ADT Rekomendasi	9
3.20 ADT Notifikasi	9
3.21 ADT Process.....	9
3.22 ADT Utility.....	9
4 Program Utama	9
5 Algoritma-Algoritma Menarik.....	10
5.1 Update Makanan di dalam Inventory dan Delivery List dll	10

5.2	Mekanisme Notifikasi.....	10
6	Data Test	11
6.1	START.....	11
6.2	EXIT	11
6.3	MOVE	11
6.4	BUY	14
6.5	MIX	15
6.6	CHOP.....	16
6.7	FRY	17
6.8	BOIL	18
6.9	DELIVERY	20
6.10	CATALOG	21
6.11	COOKBOOK.....	21
6.12	REKOMENDASI	21
6.13	KULKAS	22
6.14	WAIT	29
6.15	UNDO.....	30
6.16	REDO	32
6.17	INVENTORY	35
7	Test Script	36
8	Pembagian Kerja dalam Kelompok	38
9	Lampiran	39
9.1	Deskripsi Tugas Besar	39
9.2	Notulen Rapat	39
9.3	Log Activity Anggota Kelompok	43

1 Ringkasan

Cooking BNMO adalah program yang kami, kelompok I dari K01 IF 2110 Algoritma dan Struktur Data, buat untuk menyelesaikan permasalahan yang diberikan, yaitu untuk membuat sebuah simulator memasak yang memanfaatkan beberapa ADT. ADT-ADT ini akan diimplementasikan menjadi fitur-fitur pada simulator memasak, sehingga perlu penyesuaian ADT dari ADT yang primitif menjadi ADT yang termodifikasi sesuai dengan kebutuhan fitur.

Isi dari laporan ini adalah untuk menjelaskan spesifikasi-spesifikasi, ADT, Algoritma-algoritma menarik, dan melampirkan hasil pengetesan program yang kami buat. Penjelasan spesifikasi meliputi spesifikasi wajib dan juga spesifikasi bonus yang dikerjakan. ADT berisikan Struktur-Struktur Data Primitif yang kami gunakan. Terakhir, Algoritma-Algoritma menarik adalah untuk mencatat algoritma-algoritma yang sekiranya kami anggap menarik.

Secara garis besar, Tugas Besar ini melatih kami agar dapat mengimplementasikan struktur-struktur data dan algoritma-algoritma yang kami pelajari dalam Mata Kuliah IF2110 Algoritma dan Struktur Data, pada kasus ini, adalah implementasi sebagai sebuah permainan berbasis *command*. Pada Akhirnya, kami merasa bahwa tugas besar ini sangat membantu kami dalam memahami dan mengimplementasikan ADT yang kami sudah pelajari.

2 Penjelasan Tambahan Spesifikasi Tugas

2.1 Inisiasi

Ketika user memasukkan command start, aplikasi meminta input nama user sebelum melakukan konfigurasi file. Setelah itu, program akan membaca seluruh konfigurasi yang diperlukan dari file di folder test.

2.2 Simulator

Simulator pada Cooking BNMO menyimpan dua informasi, yaitu state dan nama user. State menyimpan informasi penting dari BNMO seperti posisi, waktu, inventory, delivery list, dan list makanan yang expired, delivered, serta sedang diproses pada waktu tersebut. State dimanfaatkan untuk mengetahui keadaan BNMO pada tiap waktu.

2.3 Mekanisme Waktu

Bertambahnya waktu hanya terhitung berdasarkan keberhasilan command. Mekanisme waktu juga akan berpengaruh pada undo dan redo karena undo hanya bisa dilakukan untuk aksi yang memakan waktu, dan redo hanya bisa dipanggil setelah undo.

2.4 Kulkas

Kulkas ditampilkan dengan representasi 'X' sebagai tempat yang terisi dan 'O' sebagai tempat yang masih kosong. Pemilihan tempat pada kulkas oleh user dibebaskan asal ada space yang tersedia. Akan disediakan *command* untuk menaruh barang dari inventory ke dalam kulkas, mengambil barang dari kulkas ke dalam inventory, dan melihat isi kulkas.

2.5 Waktu Pengolahan Makanan

Waktu pengolahan makanan unik per makanan sehingga makanan dengan aksi yang sama belum tentu memiliki waktu pengolahan yang sama. Tiap dilakukan command FRY, BOIL, CHOP, atau MIX waktu akan bertambah satu menit. Tetapi, makanan hasil pengolahan akan masuk ke dalam inventory setelah waktu pengolahan makanan yang telah dikonfigurasi di awal program. Sebelum masuk ke inventory, makanan akan masuk ke Processed List. Processed List memiliki mekanisme yang mirip dengan Delivery List, tetapi digunakan untuk makanan hasil pengolahan.

2.6 Rekomendasi Makanan

Rekomendasi Makanan diimplementasikan menggunakan Set, tanpa implementasi rekomendasi bertingkat. Set dibuat berdasarkan List Makanan hasil konfigurasi yang telah terurut secara ID Makanan. Tiap indeks pada set merepresentasikan makanan dengan indeks yang sama pada List Makanan. Untuk pembuatan set dari inventory, angka '1' merepresentasikan makanan tersebut ada di inventory dan angka '0' artinya tidak memiliki makanan tersebut. Sedangkan untuk set dari resep, angka '1' merepresentasikan bahan tersebut dibutuhkan dan angka '0' artinya bahan tersebut tidak diperlukan. Apabila set resep suatu makanan adalah subset dari set inventory, makanan tersebut akan direkomendasikan.

2.7 Undo dan Redo Stack

Stack pada BNMO terdiri dari Undo dan Redo stack yang menyimpan state BNMO. Undo Stack akan menyimpan state setiap **sebelum** aksi yang mengubah state. Sedangkan, Redo Stack menyimpan state setiap dipanggil aksi undo. Ketika Undo Stack kosong, maka aksi Undo tidak dapat dilakukan. Hal yang sama juga berlaku untuk aksi Redo. Redo juga hanya bisa dipanggil jika aksi sebelumnya adalah undo, jika terjadi undo, lalu aksi yang memakan waktu, lalu redo, tidak akan terjadi apa-apa karena redostack sudah dikosongkan.

3 Struktur Data (ADT)

3.1 ADT Sederhana

3.1.1 ADT Point

Modul ADT Point digunakan untuk merepresetasikan lokasi simulator pada peta. ADT ini berisi fungsi untuk memindahkan lokasi simulator sesuai arah bergerak yang diinput. Struktur data yang dimiliki adalah X (Absis) dan Y (Ordinat).

3.1.2 ADT Time

Modul ADT Time digunakan untuk merepresentasikan waktu dalam simulator. ADT ini digunakan untuk mekanisme waktu, diantaranya sebagai waktu pada aplikasi dan melakukan penambahan ketika dilakukan suatu proses, merepresentasikan kapan sebuah bahan akan kadaluarsa kadaluarsa, dan berapa lama makanan akan sampai.

3.1.3 ADT Makanan

Modul ADT Makanan digunakan untuk merepresentasikan struktur data makanan yang terdiri dari id makanan, nama makanan, waktu kedaluwarsa, lokasi aksi makanan, lama pengiriman makanan, dan waktu pengolahan makanan. ADT ini mendefinisikan selektor dan digunakan untuk membentuk struktur data makanan.

3.1.4 ADT Simulator

Modul ADT Simulator digunakan untuk merepresentasikan simulator yang memiliki informasi nama pengguna, dan state simulator. ADT ini mendefinisikan selektor dan digunakan untuk membentuk simulator.

3.1.5 ADT List Statik

Modul ADT List Statik digunakan untuk menyimpan list makanan dan resep yang ada pada aplikasi. ADT ini diimplementasi dengan variasi list implisit dan rata kiri, sehingga akan diisi elemen kosong dengan sebuah value MARK. Operasi-operasi primitif yang tersedia pada List statik adalah antara lain: CreateListStatik, listLength, getFirstIdx, getLastIdx, isIdxValid, isIdxEff, isEmpty, isFull, readList, printList, plusMinusList, isListEqual, indexOf, extremeValues, insertFirst, insertAt, insertLast, deleteFirst, deleteAt, deleteLast, dan sortList.

3.2 ADT Matriks

Modul ADT Matriks digunakan untuk menyimpan informasi dan sebagai representasi peta dari program yang diupload oleh pengguna. Peta yang dibuat disimpan dalam bentuk karakter-karakter tertentu. Peta yang digunakan juga akan berpengaruh pada fitur-fitur lain dalam program, seperti pengolahan makanan, pemesanan makanan, dan lain sebagainya, karena proses tersebut memerlukan posisi simulator bertempat di posisi khusus. ADT ini juga digunakan dalam implementasi pembuatan kulkas, dimana matriks digunakan sebagai *space* kulkas yang akan diisi dengan barang-barang dari inventory..

3.3 ADT Mesin Karakter dan Mesin Kata

Modul ADT Mesin Karakter dan Mesin Kata adalah basis sebuah program dapat menerima input pengguna. ADT ini digunakan untuk membaca input melalui file maupun melalui terminal. Modul ADT ini terdiri dari 2 source code, yaitu charmachine.c dan wordmachine.c. File charmachine.c berisi implementasi mesin karakter dan file wordmachine.c berisi implementasi mesin kata model akuisisi versi III.

3.4 ADT Queue dengan Pendekatan Array List Dinamik

Modul ADT Queue dengan Pendekatan Array List Dinamik digunakan untuk menjadi *inventory* dan juga *delivery list* yang dimiliki oleh simulator. ADT ini dibuat dengan bentuk Priority Queue dengan pendekatan Dinamik. Perbedaan ADT ini dengan ADT Queue pada umumnya adalah elemennya bisa diakses dan diambil tanpa melalui head atau dequeue. Karena digunakan sebagai inventory, maka elemen yang ada didalam buffer dari Queue merupakan ADT Makanan, dan juga menggunakan variasi implementasi eksplisit, dengan indeks HEAD dan TAIL

disimpan, dan juga menyimpan ukuran. Operasi primitif yang tersedia pada ADT ini adalah sebagai berikut: IsEmpty, IsFull, NBEImt, MakeEmpty, DeAlokasi, Enqueue, Dequeue, PrintPrioQueueTime, isEImt, dan deleteEImt.

3.5 ADT Stack

Modul ADT Stack digunakan untuk mekanisme undo dan redo pada simulator. ADT ini dibuat dengan sistem LIFO (Last In First Out), sehingga elemen yang ada pada *top of stack* adalah elemen terbaru yang dimasukkan. Dengan ini dapat diketahui langkah apa yang terakhir kali dijalankan untuk melakukan proses *undo* dan *redo*. ADT ini diimplementasikan pada file stack.c.

3.6 ADT Tree

Modul ADT Tree adalah basis program dapat membuat sebuah pohon N-ary. ADT ini dipilih untuk merepresentasikan tree yang berisi semua konfigurasi resep makanan. Modul ADT ini hanya terdiri dari file tree.c. Implementasi ADT ini terdiri dari konstruktor, selektor, output, dan operasi-operasi tambahan lainnya. ADT tree pada modul ini terdiri dari 4 elemen, yaitu nilai pada node, array dinamis Address yang berisi child dari sebuah node, banyaknya elemen efektif, dan kapasitas maksimum array Address.

3.7 ADT String

Modul ADT String digunakan untuk menyimpan tipe data array of char, dengan operasi-operasi primitif seperti konstruktor, perbandingan, selektor, destruktur, dll, untuk memudahkan operasi pada String. ADT ini terdiri dari array of char dan juga jumlah elemen char yang ada di dalam array of char.

3.8 ADT Delivery List

Modul ADT Delivery List merupakan modifikasi dari Priority Queue, dengan Enqueue yang diurutkan berdasarkan delivery time. Operasi primitif yang utama dalam delivery list adalah untuk membeli makanan dan juga mengupdate delivery list dan mengirimkan makanan yang sudah “sampai” ke inventory.

3.9 ADT Inventory

Modul ADT Inventory merupakan modifikasi dari Priority Queue, dengan Enqueue yang diurutkan berdasarkan expired time. Operasi primitif yang utama dalam inventory adalah untuk mengambil makanan dan juga mengupdate inventory dan mengeluarkan makanan yang sudah expired.

3.10 ADT List Makanan

Modul ADT List Makanan merupakan modifikasi dari ADT List Statik dengan elemen yang disimpan berupa tipe data Makanan. Operasi primitif yang digunakan pada ADT ini yaitu Create List, Length, Sort List, dan Insert Last. Beberapa fungsi tambahan pada ADT ini yaitu fungsi readListMakanan untuk membaca konfigurasi makanan dari file indexOfID untuk mendapatkan indeks suatu makanan dari ID yang dimiliki, dan filterByAksi untuk membuat List Makanan yang sesuai dengan aksi yang diinginkan.

3.11 ADT Peta

Modul ADT Peta dibuat untuk menyimpan informasi peta, pelaksanaan proses *move* dan menampilkan visual peta yang digunakan. ADT Peta juga digunakan untuk menyimpan informasi yang dibutuhkan untuk melakukan aksi-aksi lain, seperti memesan bahan makanan, memotong bahan makanan, menggoreng bahan makanan dan lain sebagainya.

3.12 ADT Resep

Modul ADT Resep dibuat untuk menyimpan informasi resep makanan serta mencetak resep makanan. ADT ini juga digunakan sebagai panduan program akan pohon resep makanan yang dapat dibuat dari bahan-bahannya.

3.13 ADT State

Modul ADT ini dibuat untuk abstraksi dan juga memudahkan proses undo/redo, dimana semua state dari aplikasi bisa disimpan didalam ADT State

3.14 ADT State Stack

Modul ADT ini dibuat untuk menyimpan state-state dari permainan, dan menjadi ADT Dasar dari ADT UndoStack dan RedoStack. ADT ini didapatkan dari modifikasi ADT Stack, yaitu dengan mengubah element didalam stack menjadi ADT State.

3.15 ADT Undo Stack

Modul ADT ini dibuat untuk menyimpan state yang bisa di *undo*, yaitu state-state yang berubah karena suatu aksi yang berubah atau memakan waktu. Modul ADT ini diimplementasikan secara statik dengan menggunakan ADT State Stack.

3.16 ADT Redo Stack

Modul ADT ini dibuat untuk menyimpan state yang bisa di *redo*, yaitu state-state setelah di undo. Modul ADT ini diimplementasikan secara statik dengan menggunakan ADT State Stack.

3.17 ADT Kulkas

Modul ADT ini dibuat untuk penyelesaian salah satu spesifikasi bonus, yaitu keberadaannya fitur kulkas. ADT ini diimplementasi dengan 2 ADT yaitu ADT Matrix dan juga ADT List, dimana ADT Matrix akan menyimpan *mapping* dari slot kulkas, dan ADT List akan menyimpan informasi mengenai makanan seperti ID makanannya di kulkas, Indeks paling kiri atas-nya di matrix, dan juga Informasi makanannya. Semua makanan yang masuk kedalam ADT kulkas tidak akan *update* expired time nya.

3.18 ADT Set

ADT Set dibuat untuk mendukung pembuatan rekomendasi makanan. ADT ini diimplementasikan dengan list statik yang menyimpan informasi 0 atau 1. Angka 0 artinya bukan anggota set tersebut dan angka 1 artinya merupakan anggota set tersebut. ADT ini memiliki fungsi-fungsi seperti CreateSet, isEmpty, length, isIn, add, copy, isSubset, dan print.

3.19 ADT Rekomendasi

ADT Rekomendasi dibuat untuk menyelesaikan bonus rekomendasi makanan. ADT ini membuat fungsi-fungsi untuk mengecek resep makanan apa yang merupakan subset dari inventory. ADT ini memiliki fungsi seperti inventorySet, initializeSet, resepSet, cekTreeSubset, rekomendasiMakanan, dan printRekomendasiMakanan.

3.20 ADT Notifikasi

ADT Notifikasi dibuat untuk menyimpan notifikasi dalam bentuk list statik dari string dan melakukan fungsi-fungsi lain untuk mekanisme notifikasi. Beberapa fungsi mekanisme notifikasi yaitu addDelivered, addExpired, updateUndoNotif, updateRedoNotif, dan updateNotif.

3.21 ADT Process

ADT Process dibuat untuk melakukan mekanisme pembelian dan pengolahan makanan. ADT ini menggabungkan fungsi-fungsi yang telah dibuat ADT lain. Fungsi pada ADT ini yaitu buyMechanism dan olahMechanism.

3.22 ADT Utility

ADT Utility dibuat untuk mengimplementasikan fungsi-fungsi yang berkaitan dengan tampilan program. Pada ADT ini dilakukan fungsi print splash screen. Selain itu, ADT ini juga berisi fungsi untuk melakukan print berwarna ke layar.

4 Program Utama

Program utama terletak pada file main.c yang dapat dikompilasi dengan command make main. Program kemudian dapat dijalankan dan muncul sebuah splash screen yang menandakan bahwa program sudah dimulai. Pada awal program, pengguna harus melakukan input command START. Jika *command* input bukan START, maka program akan langsung keluar.

Apabila dimasukkan command START, akan dilakukan inisialisasi variabel-variabel global, seperti State pada ADT Simulator, nama user, Undo dan Redo Stack, Kulkas, Peta, dan lain-lain. Selanjutnya, program utama akan melakukan looping sampai didapatkan *command* EXIT. Selama looping berlangsung, program akan menampilkan posisi pengguna, waktu, notifikasi, dan peta. *Command* yang tidak terdaftar akan memunculkan pesan error dan langsung *ter-skip*. Beberapa command yang dapat dilakukan yaitu:

- MOVE NORTH : bergerak 1 petak ke arah Utara
- MOVE EAST : bergerak 1 petak ke arah Timur
- MOVE SOUTH : bergerak 1 petak ke arah Selatan
- MOVE WEST : bergerak 1 petak ke arah Barat

- BUY : melakukan pembelian bahan makanan
- MIX : melakukan pencampuran bahan makanan
- CHOP : melakukan pemotongan bahan makanan
- FRY : melakukan penggorengan bahan makanan
- BOIL : melakukan perebusan bahan makanan
- DELIVERY : menampilkan makanan yang sedang diantar
- CATALOG : melakukan seluruh makanan yang bisa dibeli maupun dibuat
- COOKBOOK : melakukan seluruh resep makanan
- INVENTORY : menampilkan inventory pengguna
- REKOMENDASI : menampilkan rekomendasi makanan yang bisa dibuat
- INSERTKULKAS : memasukkan bahan makanan ke kulkas
- GETKULKAS : mengambil bahan makanan dari kulkas
- DISPLAYKULKAS : menampilkan kulkas pengguna
- WAIT x y : menunggu selama x jam dan y menit
- UNDO : membatalkan aksi sebelumnya
- REDO : membatalkan undo

Program utama ini pada dasarnya digunakan untuk meng-*connect* semua ADT dan modul-modul yang ada. Program utama akan membaca input dan membandingkannya dengan *command* yang tersedia untuk dipanggil. Jika input adalah *command*, maka aksi dari *command* tersebut akan dijalankan.

5 Algoritma-Algoritma Menarik

5.1 Update Makanan di dalam Inventory dan Delivery List dll

Algoritma untuk meng-*update* makanan di dalam Inventory, Delivery List, dan lain-lain untuk mengurangi *expired time*, *delivery time*, *processing time*, dan lain-lain, menurut kami menarik karena menggabungkan fungsi untuk mengurangi hal-hal tersebut dengan aksi yang dilakukannya secara otomatis. Seperti contoh, jika dalam Inventory terdapat makanan yang *expired* nya dalam 5 menit, lalu dilakukan command WAIT selama 6 menit, makanan yang *expired* tersebut akan secara otomatis terbuang dari inventory, dan makanan yang *delivery time*-nya sudah mencapai 0 akan secara otomatis terkirim ke dalam inventory.

5.2 Mekanisme Notifikasi

Algoritma yang digunakan untuk melakukan mekanisme notifikasi sangat menarik karena memiliki beberapa langkah inisialisasi yang sangat krusial urutannya. Notifikasi direpresentasikan sebagai list of string sehingga setelah ditampilkan di awal looping, notifikasi harus diinisialisasi ulang menjadi list kosong untuk menampung notifikasi yang dihasilkan dari perubahan pada looping selanjutnya. Selain itu, mekanisme notifikasi untuk notifikasi undo dan redo juga menarik. Karena kami tidak menyimpan aksi yang dilakukan ke dalam state, kami dapat memperoleh informasi suatu aksi dibatalkan melalui perbandingan delivery list dan process list. Dengan membandingkan delivery list dan process list dari dua state, dapat dibuat notifikasi suatu aksi dibatalkan atau dilakukan.

6 Data Test

6.1 START

Command START digunakan untuk memulai program COOKING BNMO. Pada awalnya program akan meminta input nama user dan mengatur posisi dan waktu dari nol.

```
Enter Command: START
Masukkan nama user: Livia
BNMO berada di posisi: (0,0)
Waktu: 0
Notifikasi: -
* * * * *
* S . . . . . *
* . . . . T . X . *
* . M . . . . X . *
* . . . . . X . *
* . . . . X X X X . *
* . X . . . . . *
* . X . . . . . C . *
* . X X X . . F . . *
* . . . . . B . . *
* * * * *
Enter Command: 
```

Gambar 6.1.1 Tampilan awal ketika user memasukkan command START

6.2 EXIT

Command EXIT digunakan jika pemain sudah selesai menggunakan program.

```
Masukkan pilihan: 0
BNMO berada di posisi: (9,7)
Waktu: 21
Notifikasi: -
* * * * *
* . . . . T . X . *
* . M . . . . X . *
* . . . . . X . *
* . . . . X X X X . *
* . X . . . . . *
* . X . . . . . C . *
* . X X X . . F . . *
* . . . . . B S . . *
* * * * *
Enter Command: EXIT
Terima kasih sudah menggunakan program ini!
```

Gambar 6.2.1 Tampilan ketika command EXIT dimasukkan

6.3 MOVE

Command MOVE dapat dilakukan untuk memindahkan posisi simulator dalam peta agar dapat melakukan aksi-aksi yang bergantung pada posisi simulator. Command move dapat dilakukan dengan parameter tambahan untuk menentukan arah gerakannya, yaitu NORTH, SOUTH, EAST dan WEST.

```

BNMO berada di posisi: (0,0)
Waktu: 0
Notifikasi: -
* * * * *
* S . . . . . *
* . . . . T . . X . . *
* . M . . . . . X . . *
* . . . . . X . . . *
* . . . . X X X X . . *
* . X . . . . . *
* . X . . . . . C . . *
* . X X X . . F . . . *
* . . . . . *
* . . . . . B . . . *
* * * * *

Enter Command: MOVE EAST
BNMO berada di posisi: (0,1)
Waktu: 1
Notifikasi: -
* * * * *
* S . . . . . *
* . . . . T . . X . . *
* . M . . . . . X . . *
* . . . . . X . . . *
* . . . . X X X X . . *
* . X . . . . . *
* . X . . . . . C . . *
* . X X X . . F . . . *
* . . . . . *
* . . . . . B . . . *
* * * * *

Enter Command:

```

Gambar 6.3.1 Tampilan saat melakukan command MOVE EAST

```

BNMO berada di posisi: (2,3)
Waktu: 9
Notifikasi: -
* * * * *
* . . . . T . . X . . *
* . M . S . . . X . . *
* . . . . . X . . . *
* . . . . X X X X . . *
* . X . . . . . *
* . X . . . . . C . . *
* . X X X . . F . . . *
* . . . . . *
* . . . . . B . . . *
* * * * *

Enter Command: MOVE WEST
BNMO berada di posisi: (2,2)
Waktu: 10
Notifikasi: -
* * * * *
* . . . . T . . X . . *
* . M S . . . X . . *
* . . . . . X . . . *
* . . . . X X X X . . *
* . X . . . . . *
* . X . . . . . C . . *
* . X X X . . F . . . *
* . . . . . *
* . . . . . B . . . *
* * * * *

Enter Command:

```

Gambar 6.3.2 Tampilan saat melakukan command MOVE WEST

```

BNMO berada di posisi: (0,3)
Waktu: 3
Notifikasi: -
* * * * *
* . . . S . . . . *
* . . . . T . . X . *
* . M . . . . X . *
* . . . . . X . . *
* . . . . X X X X . *
* . X . . . . . *
* . X . . . . . C . *
* . X X X . . F . . *
* . . . . . . . *
* . . . . . B . . . *
* * * * *
Enter Command: MOVE SOUTH
BNMO berada di posisi: (1,3)
Waktu: 4
Notifikasi: -
* * * * *
* . . . . . *
* . . . S T . . X . *
* . M . . . . X . *
* . . . . . X . . *
* . . . . X X X X . *
* . X . . . . . *
* . X . . . . . C . *
* . X X X . . F . . *
* . . . . . . . *
* . . . . . B . . . *
* * * * *
Enter Command:

```

Gambar 6.3.3 Tampilan saat melakukan command MOVE SOUTH

```

BNMO berada di posisi: (2,2)
Waktu: 10
Notifikasi: -
* * * * *
* . . . . . *
* . . . . T . . X . *
* . M S . . . . X . *
* . . . . . X . . *
* . . . . X X X X . *
* . X . . . . . *
* . X . . . . . C . *
* . X X X . . F . . *
* . . . . . . . *
* . . . . . B . . . *
* * * * *
Enter Command: MOVE NORTH
BNMO berada di posisi: (1,2)
Waktu: 11
Notifikasi: -
* * * * *
* . . . . . *
* . . S . T . . X . *
* . M . . . . X . *
* . . . . . X . . *
* . . . . X X X X . *
* . X . . . . . *
* . X . . . . . C . *
* . X X X . . F . . *
* . . . . . . . *
* . . . . . B . . . *
* * * * *
Enter Command:

```

Gambar 6.3.4 Tampilan saat melakukan command MOVE NORTH

```

BMO berada di posisi: (2,2)
Waktu: 12
Notifikasi: -
*****
* . . . . T . . X . . *
* . M S . . . . X . . *
* . . . . . X . . . *
* . . . . X X X X . . *
* . X . . . . . . . *
* . X . . . . . C . *
* . X X X . . F . . . *
* . . . . . . . . . *
* . . . . . B . . . *
*****
Enter Command: MOVE WEST
BMO berada di posisi: (2,2)
Waktu: 12
Notifikasi:
1. Ups! Kamu tidak bisa bergerak karena terhalang :D
*****
* . . . . T . . X . . *
* . M S . . . . X . . *
* . . . . . X . . . *
* . . . . X X X X . . *
* . X . . . . . . . *
* . X . . . . . C . *
* . X X X . . F . . . *
* . . . . . . . . . *
* . . . . . B . . . *
*****
Enter Command:

```

Gambar 6.3.5 Tampilan saat melakukan command MOVE X, dengan arah tujuan gerak tidak dapat diakses

6.4 BUY

Command BUY dapat dilakukan untuk melakukan pemesanan bahan makanan. Pemesanan bahan makanan dapat gagal jika pilihan makanan tidak valid. Jika, pembelian berhasil, maka ketika makanan telah masuk ke inventory akan muncul notifikasi.

```

Enter Command: BUY
=====
=          BUY          =
=====
List Bahan Makanan:
1. Tepung (30 menit)
2. Ayam Mentah (2 menit)
3. Telur Ayam (5 menit)
4. Cabai (1 jam)
5. Minyak Goreng (1 jam)
6. Mie Kering (15 menit)

Kirim 0 untuk exit
Masukkan pilihan: 7
Makanan yang dipilih tidak valid. Silahkan input kembali.

```

Gambar 6.4.1. Tampilan saat pilihan makanan tidak valid

```

Enter Command: BUY
=====
=          BUY          =
=====
List Bahan Makanan:
1. Tepung (30 menit)
2. Ayam Mentah (2 menit)
3. Telur Ayam (5 menit)
4. Cabai (1 jam)
5. Minyak Goreng (1 jam)
6. Mie Kering (15 menit)

Kirim 0 untuk exit
Masukkan pilihan: 6
Berhasil memesan Mie Kering. Mie Kering akan diantar dalam 15 menit.
Masukkan pilihan: 2
Berhasil memesan Ayam Mentah. Ayam Mentah akan diantar dalam 2 menit.

```

Gambar 6.4.2. Tampilan saat berhasil membeli makanan

```

BNMO berada di posisi: (2,3)
Waktu: 7
Notifikasi:
1. Ayam Mentah sudah diterima oleh Cooking BNMO! :D
* * * * *
* . . . . . *
* . . . T . . X . . *
* . M . S . . . X . . *
* . . . . . X . . . *
* . . . . X X X X . . *
* . X . . . . . *
* . X . . . . . C . . *
* . X X X . . F . . . *
* . . . . . B . . . *
* * * * *

```

Gambar 6.4.3. Tampilan saat makanan telah sampai

6.5 MIX

Command MIX adalah command yang digunakan untuk memproses makanan yang dilakukan dengan cara mencampur. Pertama-tama akan ditampilkan makanan apa saja yang dapat dibuat dengan cara *mix*, lalu user memilih makanan yang mau dibuat. Jika bahan untuk membuat makanan tersebut cukup, maka makanan akan diproses. Jika kurang maka gagal diproses dan menampilkan bahan yang kurang. Command ini hanya bisa dilakukan di lokasi “M” pada peta.

```

Enter Command: MIX
=====
=          MIX          =
=====
List Bahan Makanan yang Bisa Dibuat:
1. Ayam Tepung
2. Ayam Geprek

Kirim 0 untuk exit
Masukkan pilihan: 2
Gagal membuat Ayam Geprek karena kamu tidak memiliki bahan berikut:
1. Ayam Goreng
2. Sambal

```

Gambar 6.5.1 Tampilan saat gagal memproses makanan

```

Enter Command: MIX
=====
=          MIX          =
=====
List Bahan Makanan yang Bisa Dibuat:
1. Ayam Tepung
2. Ayam Geprek

Kirim 0 untuk exit
Masukkan pilihan: 1
Ayam Tepung sedang diproses!

```

Gambar 6.5.2 Tampilan saat berhasil memproses makanan

```

BNMO berada di posisi: (2,2)
Waktu: 57
Notifikasi:
1. Ayam Tepung sudah diterima oleh Cooking BNMO! :D
* * * * *
* . . . . .
* . . . . T . . X . .
* . M S . . . . X . .
* . . . . . . . X . .
* . . . . X X X X . .
* . X . . . . . . .
* . X . . . . . C . .
* . X X X . . F . .
* . . . . . . . .
* . . . . . B . .
* * * * *

```

Gambar 6.5.3 Tampilan saat makanan selesai di-mix dan dimasukkan ke inventory

6.6 CHOP

Command CHOP adalah command yang digunakan untuk memproses makanan yang dilakukan dengan cara memotong. Pertama-tama akan ditampilkan makanan apa saja yang dapat dibuat dengan cara *chop*, lalu user memilih makanan yang mau dibuat. Jika bahan untuk membuat makanan tersebut cukup, maka makanan akan diproses. Jika kurang maka gagal diproses dan menampilkan bahan yang kurang. Command ini hanya bisa dilakukan di lokasi “C” pada peta.

```

Enter Command: CHOP
=====
=          CHOP          =
=====
List Bahan Makanan yang Bisa Dibuat:
1. Ayam Potong
2. Sambal

Kirim 0 untuk exit
Masukkan pilihan: 2
Gagal membuat Sambal karena kamu tidak memiliki bahan berikut:
1. Cabai

```


Gambar 6.6.1 Tampilan saat gagal memproses makanan.

```

Enter Command: CHOP
=====
=          CHOP          =
=====
List Bahan Makanan yang Bisa Dibuat:
1. Ayam Potong
2. Sambal

Kirim 0 untuk exit
Masukkan pilihan: 1
Ayam Potong sedang diproses!

```

Gambar 6.6.2 Tampilan saat berhasil memproses makanan.

```

BNMO berada di posisi: (6,6)
Waktu: 35
Notifikasi:
1. Ayam Potong sudah diterima oleh Cooking BNMO! :D
* * * * *
* . . . . . *
* . . . T . X . *
* . M . . . . X . *
* . . . . . X . *
* . . . X X X X . *
* . X . . . . . *
* . X . . . . S . C . *
* . X X X . . F . . *
* . . . . . . . *
* . . . . . B . . . *
* * * * *

```

Gambar 6.6.3 Tampilan saat makanan selesai di-*chop* dan dimasukkan ke inventory

6.7 FRY

Command FRY adalah command yang digunakan untuk memproses makanan yang dilakukan dengan cara menggoreng. Pertama-tama akan ditampilkan makanan apa saja yang dapat dibuat dengan cara *fry*, lalu user memilih makanan yang mau dibuat. Jika bahan untuk membuat makanan tersebut cukup, maka makanan akan diproses. Jika kurang maka gagal diproses dan menampilkan bahan yang kurang. Command ini hanya bisa dilakukan di lokasi “F” pada peta.

```

Enter Command: FRY
=====
=          FRY          =
=====
List Bahan Makanan yang Bisa Dibuat:
1. Telur Mata Sapi
2. Mie Goreng
3. Ayam Goreng

Kirim 0 untuk exit
Masukkan pilihan: 3
Gagal membuat Ayam Goreng karena kamu tidak memiliki bahan berikut:
1. Ayam Tepung
2. Minyak Goreng
Masukkan pilihan: 

```

Gambar 6.7.1 Tampilan saat gagal memproses makanan.

```

Enter Command: FRY
=====
=          FRY          =
=====
List Bahan Makanan yang Bisa Dibuat:
1. Telur Mata Sapi
2. Mie Goreng
3. Ayam Goreng

Kirim 0 untuk exit
Masukkan pilihan: 1
Telur Mata Sapi sedang diproses!
Masukkan pilihan: █

```

Gambar 6.7.2 Tampilan saat berhasil memproses makanan

```

BNMO berada di posisi: (6,5)
Waktu: 1.23
Notifikasi:
1. Telur Mata Sapi sudah diterima oleh Cooking BNMO! :D
* * * * *
*   .   .   .   .   .   .   *
*   .   .   .   T   .   X   .   *
*   .   M   .   .   .   .   X   .   *
*   .   .   .   .   .   .   X   .   *
*   .   .   .   X   X   X   X   .   *
*   .   X   .   .   .   .   .   .   *
*   .   X   .   .   .   S   .   C   .   *
*   .   X   X   X   .   .   F   .   .   *
*   .   .   .   .   .   .   .   .   *
*   .   .   .   .   .   .   B   .   .   *
* * * * *

```

Gambar 6.7.3 Tampilan saat makanan selesai di-fry dan dimasukkan ke inventory

6.8 BOIL

Command BOIL adalah command yang digunakan untuk memproses makanan yang dilakukan dengan cara merebus. Pertama-tama akan ditampilkan makanan apa saja yang dapat dibuat dengan cara *boil* , lalu user memilih makanan yang mau dibuat. Jika bahan untuk membuat makanan tersebut cukup, maka makanan akan diproses. Jika kurang maka gagal diproses dan menampilkan bahan yang kurang. Command ini hanya bisa dilakukan di lokasi “B” pada peta.

```

Enter Command: BOIL
=====
=          BOIL          =
=====
List Bahan Makanan yang Bisa Dibuat:
1. Telur Rebus
2. Mie Rebus

Kirim 0 untuk exit
Masukkan pilihan: 2
Gagal membuat Mie Rebus karena kamu tidak memiliki bahan berikut:
1. Mie Kering

```

Gambar 6.8.1 Tampilan saat gagal memproses makanan.

```

Enter Command: BOIL
=====
=          BOIL          =
=====
List Bahan Makanan yang Bisa Dibuat:
1. Telur Rebus
2. Mie Rebus

Kirim 0 untuk exit
Masukkan pilihan: 1
Telur Rebus sedang diproses!

```

Gambar 6.8.2 Tampilan saat berhasil memproses makanan.

```

BNMO berada di posisi: (9,7)
Waktu: 32
Notifikasi:
1. Telur Rebus sudah diterima oleh Cooking BNMO! :D
* * * * *
* . . . . .
* . . . T . . X . .
* . M . . . . X . .
* . . . . . X . .
* . . . . X X X X .
* . X . . . . .
* . X . . . . . C .
* . X X X . . F . .
* . . . . .
* . . . . . B S . .
* * * * *

```

Gambar 6.8.3 Tampilan saat makanan selesai di-boil dan dimasukkan ke inventory

6.9 DELIVERY

Command ini digunakan untuk melihat daftar bahan yang sedang berada di dalam perjalanan. Barang yang dibeli akan dimasukkan ke dalam delivery list. Jika barang sudah sampai, bahan akan dikeluarkan dari delivery list

```
Enter Command: DELIVERY

=====DELIVERY LIST=====
Delivery List Kosong
```

Gambar 6.9.1 Tampilan saat sedang tidak ada makanan yang diantar

```
Enter Command: DELIVERY

=====DELIVERY LIST=====
List Makanan di Perjalanan
(nama - waktu sisa delivery)
1. (12) Telur Ayam - 3 menit
2. (14) Cabai - 59 menit
```

Gambar 6.9.2 Tampilan saat sudah membeli makanan dan makanan sedang dalam perjalanan

```
Notifikasi:
1. Telur Ayam sudah diterima oleh Cooking BNMO! :D
* * * * *
*   . . . . . S . . *
*   . . . . . T . . X . . *
*   . M . . . . . X . . *
*   . . . . . X . . *
*   . . . . . X X X X . . *
*   . X . . . . . . . *
*   . X . . . . . C . . *
*   . X X X . . F . . *
*   . . . . . . . . *
*   . . . . . B . . *
* * * * *
Enter Command: DELIVERY

=====DELIVERY LIST=====
List Makanan di Perjalanan
(nama - waktu sisa delivery)
1. (14) Cabai - 56 menit
```

Gambar 6.9.3 Tampilan saat makanan sudah sampai dan dikeluarkan dari delivery list

6.10 CATALOG

Pemain dapat melihat seluruh makanan hasil konfigurasi dengan command CATALOG. Tampilan yang akan muncul setelah command ini dipanggil adalah sebagai berikut.

```
Enter Command: CATALOG
=====List Makanan=====
Nama Makanan - Durasi Kedaluwarsa - Aksi yang Diperlukan - Delivery Time - Waktu Pengolahan Makanan - Ukuran Makanan
1. Tepung - 1 jam - BUY - 30 menit - 0 - 1 x 2
2. Ayam Mentah - 1 hari - BUY - 2 menit - 0 - 3 x 3
3. Telur Ayam - 1 hari - BUY - 5 menit - 0 - 1 x 1
4. Cabai - 1 hari - BUY - 1 jam - 0 - 2 x 2
5. Minyak Goreng - 5 hari - BUY - 1 jam - 0 - 2 x 1
6. Mie Kering - 3 hari - BUY - 15 menit - 0 - 2 x 1
7. Ayam Potong - 2 jam - CHOP - 0 - 10 menit - 2 x 2
8. Sambal - 12 jam - CHOP - 0 - 10 menit - 2 x 2
9. Telur Rebus - 30 menit - BOIL - 0 - 5 menit - 1 x 1
10. Mie Rebus - 3 jam - BOIL - 0 - 3 menit - 1 x 2
11. Telur Mata Sapi - 2 jam - FRY - 0 - 2 menit - 2 x 1
12. Ayam Tepung - 3 jam 30 menit - MIX - 0 - 5 menit - 2 x 2
13. Mie Goreng - 10 jam - FRY - 0 - 8 menit - 2 x 2
14. Ayam Goreng - 5 jam - FRY - 0 - 10 menit - 2 x 2
15. Ayam Geprek - 12 jam - MIX - 0 - 5 menit - 2 x 3
```

Gambar 6.10.1 Tampilan saat dijalankan command CATALOG

6.11 COOKBOOK

Pemain dapat melihat seluruh resep hasil konfigurasi dengan command COOKBOOK. Tampilan yang akan muncul setelah command ini dipanggil adalah sebagai berikut.

```
Enter Command: COOKBOOK
=====RESEP=====
Telur Rebus
BOIL - Telur Ayam
Ayam Geprek
MIX - Ayam Goreng - Sambal
Ayam Goreng
FRY - Ayam Tepung - Minyak Goreng
Ayam Tepung
MIX - Ayam Potong - Tepung
Ayam Potong
CHOP - Ayam Mentah
Sambal
CHOP - Cabai
Mie Goreng
FRY - Telur Mata Sapi - Mie Rebus - Minyak Goreng
Telur Mata Sapi
FRY - Minyak Goreng - Telur Ayam
Mie Rebus
BOIL - Mie Kering
```

Gambar 6.11.1 Tampilan saat dijalankan command COOKBOOK

6.12 REKOMENDASI

Pemain dapat memperoleh rekomendasi makanan dari bahan makanan yang dimiliki. Jika belum memiliki bahan yang mencukupi, maka bisa belum ada rekomendasi.

STEI- ITB	IF2110-TB-I-01	Halaman 21 dari 45 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: REKOMENDASI

=====REKOMENDASI=====
Rekomendasi makanan buat kamu ;)
Belum ada rekomendasi yang pas, nih! Beli bahan dulu, yuk :D

```

Gambar 6.12.1 Tampilan saat belum ada rekomendasi

```

Enter Command: REKOMENDASI

=====REKOMENDASI=====
Rekomendasi makanan buat kamu ;)
1. Telur Rebus
2. Ayam Potong
3. Telur Mata Sapi
4. Mie Rebus

```

Gambar 6.12.2. Tampilan rekomendasi saat ada bahan yang sesuai

6.13 KULKAS

Tiga command yang digunakan untuk fitur kulkas adalah INSERTKULKAS, GETKULKAS, dan DISPLAYKULKAS. Untuk INSERTKULKAS, pemain dapat memasukkan barang kedalam kulkas, kasus normal nya adalah inventory pemain terisi dan kulkas belum penuh pada slot tersebut, dan kasus *edge case* nya adalah saat inventory pemain kosong atau kulkas terisi pada slot tersebut (jika kulkas sudah penuh langsung teratasi).

Terakhir, `DISPLAYKULKAS`, dimana hanya ada 1 kasus, yaitu mendisplay *state* dari kulkas saat ini.

[illegible]

Gambar 6.13.7 Display Kulkas


```
BNMO berada di posisi: (0,4)
Waktu: 5
Notifikasi: -
* * * * *
* . . . . S . . . . *
* . . . . T . . X . . *
* . M . . . . X . . *
* . . . . . . . X . . *
* . . . . X X X X . . *
* . X . . . . . . . *
* . X . . . . . . C . *
* . X X X . . F . . *
* . . . . . . . . . *
* . . . . . B . . . *
* * * * *
Enter Command: WAIT 0 15
BNMO berada di posisi: (0,4)
Waktu: 20
Notifikasi:
1. Mie Kering sudah diterima oleh Cooking BNMO! :D
* * * * *
* . . . . S . . . . *
* . . . . T . . X . . *
* . M . . . . X . . *
* . . . . . . . X . . *
* . . . . X X X X . . *
* . X . . . . . . . *
* . X . . . . . . C . *
* . X X X . . F . . *
* . . . . . . . . . *
* . . . . . B . . . *
* * * * *
Enter Command: █
```

Gambar 6.14.1 Tampilan saat melakukan command WAIT

6.15 UNDO

Pemain dapat membatalkan suatu aksi yang mengubah waktu alias aksi yang mengubah *state* dari permainan. Seperti contoh:

```

BNMO berada di posisi: (0,3)
Waktu: 3
Notifikasi: -
* * * * *
* . . . S . . . . *
* . . . T . . X . *
* . M . . . . X . *
* . . . . . X . . *
* . . . . . X . . *
* . . . . X X X . *
* . X . . . . . *
* . X . . . . . C . *
* . X X X . . F . . *
* . . . . . . . . *
* . . . . . B . . . *
* * * * *
Enter Command: MOVE EAST
BNMO berada di posisi: (0,4)
Waktu: 4
Notifikasi: -
* * * * *
* . . . S . . . . *
* . . . T . . X . *
* . M . . . . X . *
* . . . . . X . . *
* . . . . X X X . *
* . X . . . . . *
* . X . . . . . C . *
* . X X X . . F . . *
* . . . . . . . . *
* . . . . . B . . . *
* * * * *
Enter Command: UNDO
BNMO berada di posisi: (0,3)
Waktu: 3
Notifikasi: -
* * * * *
* . . . S . . . . *
* . . . T . . X . *
* . M . . . . X . *
* . . . . . X . . *
* . . . . X X X . *
* . X . . . . . *
* . X . . . . . C . *
* . X X X . . F . . *
* . . . . . . . . *
* . . . . . B . . . *
* * * * *
Enter Command:

```

Gambar 6.15.1 Tampilan saat dijalankan command UNDO

Setelah command UNDO, permainan akan kembali ke *state* sebelum command yang di-*undo* dijalankan. Tetapi jika pemain menggunakan sebuah command yang tidak mengubah *state* lalu melakukan command UNDO, maka yang akan ter-*undo* adalah aksi yang mengubah *state* terakhir kali. Contohnya:

```
BNMO berada di posisi: (0,3)
Waktu: 3
Notifikasi: -
*****
* . . . S . . . . *
* . . . . T . . X . *
* . M . . . . X . . *
* . . . . . X . . . *
* . . . . X X X X . *
* . X . . . . . *
* . X . . . . . C . *
* . X X X . . F . . *
* . . . . . . . *
* . . . . . B . . . *
*****
Enter Command: MOVE EAST
BNMO berada di posisi: (0,4)
Waktu: 4
Notifikasi: -
*****
* . . . . S . . . . *
* . . . . T . . X . *
* . M . . . . X . . *
* . . . . . X . . . *
* . . . . X X X X . *
* . X . . . . . *
* . X . . . . . C . *
* . X X X . . F . . *
* . . . . . . . *
* . . . . . B . . . *
*****
Enter Command: COOKBOOK

=====RESEP=====
Telur Rebus
BOIL - Telur Ayam
Ayam Geprek
MIX - Ayam Goreng - Sambal
Ayam Goreng
FRY - Ayam Tepung - Minyak Goreng
Ayam Tepung
MIX - Ayam Potong - Tepung
Ayam Potong
CHOP - Ayam Mentah
Sambal
CHOP - Cabai
Mie Goreng
FRY - Telur Mata Sapi - Mie Rebus - Minyak Goreng
Telur Mata Sapi
FRY - Minyak Goreng - Telur Ayam
Mie Rebus
BOIL - Minyak Goreng
BNMO berada di posisi: (0,4)
Waktu: 4
Notifikasi: -
*****
* . . . . S . . . . *
* . . . . T . . X . *
* . M . . . . X . . *
* . . . . . X . . . *
* . . . . X X X X . *
* . X . . . . . *
* . X . . . . . C . *
* . X X X . . F . . *
* . . . . . . . *
* . . . . . B . . . *
*****
Enter Command: UNDO
BNMO berada di posisi: (0,3)
Waktu: 3
Notifikasi: -
*****
* . . . . S . . . . *
* . . . . T . . X . *
* . M . . . . X . . *
* . . . . . X . . . *
* . . . . X X X X . *
* . X . . . . . *
* . X . . . . . C . *
* . X X X . . F . . *
* . . . . . . . *
* . . . . . B . . . *
*****
Enter Command: 
```

Gambar 6.15.2 Tampilan saat dijalankan command UNDO pada aksi yang tidak mengubah *state*

Command UNDO juga hanya bisa dijalankan jika sudah pernah dilakukan aksi yang mengubah *state*, yang akan mengisi *state stack* sehingga tidak kosong.

6.16 REDO

Pemain dapat membatalkan suatu UNDO, sehingga command REDO hanya dapat dipanggil jika aksi dari pemain sebelumnya adalah UNDO, dan tidak bisa dipanggil jika setelah UNDO, lalu aksi yang mengubah *state*, lalu REDO, tetapi, jika aksi yang dilakukan adalah aksi yang tidak mengubah *state*, maka REDO tetap bisa dilakukan.

BNMO berada di posisi: (0,6)
Waktu: 7
Notifikasi: -
* * * * *
* S . . . *
* T . . X . . *
* . M X . . *
* X . . . *
* X X X X . . *
* . X *
* . X C . . *
* . X X X . . F . . . *
* *
* B *
* * * * *
Enter Command: MOVE EAST

BNMO berada di posisi: (0,7)
Waktu: 8
Notifikasi: -
* * * * *
* S . . . *
* T . . X . . *
* . M X . . *
* X . . . *
* X X X X . . *
* . X *
* . X C . . *
* . X X X . . F . . . *
* *
* B *
* * * * *
Enter Command: UNDO

BNMO berada di posisi: (0,6)
Waktu: 7
Notifikasi: -
* * * * *
* S . . . *
* T . . X . . *
* . M X . . *
* X . . . *
* X X X X . . *
* . X *
* . X C . . *
* . X X X . . F . . . *
* *
* B *
* * * * *
Enter Command: REDO

BNMO berada di posisi: (0,7)
Waktu: 8
Notifikasi: -
* * * * *
* S . . . *
* T . . X . . *
* . M X . . *
* X . . . *
* X X X X . . *
* . X *
* . X C . . *
* . X X X . . F . . . *
* *
* B *
* * * * *
Enter Command:

Gambar 6.16.1 kasus REDO normal

```
BNMO berada di posisi: (0,7)
Waktu: 8
Notifikasi: -
* * * * *
* . . . . . S . . *
* . . . . T . . X . . *
* . M . . . . . X . . *
* . . . . . X . . *
* . . . . X X X X . . *
* . X . . . . . . . *
* . X . . . . . C . . *
* . X X X . . F . . *
* . . . . . . . . *
* . . . . . B . . . *
* * * * *
Enter Command: MOVE EAST
BNMO berada di posisi: (0,8)
Waktu: 9
Notifikasi: -
* * * * *
* . . . . . S . . *
* . . . . T . . X . . *
* . M . . . . . X . . *
* . . . . . X . . *
* . . . . X X X X . . *
* . X . . . . . . . *
* . X . . . . . C . . *
* . X X X . . F . . *
* . . . . . . . . *
* . . . . . B . . . *
* * * * *
Enter Command: UNDO

BNMO berada di posisi: (0,7)
Waktu: 8
Notifikasi: -
* * * * *
* . . . . . S . . *
* . . . . T . . X . . *
* . M . . . . . X . . *
* . . . . . X . . *
* . . . . X X X X . . *
* . X . . . . . . . *
* . X . . . . . C . . *
* . X X X . . F . . *
* . . . . . . . . *
* . . . . . B . . . *
* * * * *
Enter Command: MOVE WEST
BNMO berada di posisi: (0,6)
Waktu: 9
Notifikasi: -
* * * * *
* . . . . . S . . *
* . . . . T . . X . . *
* . M . . . . . X . . *
* . . . . . X . . *
* . . . . X X X X . . *
* . X . . . . . . . *
* . X . . . . . C . . *
* . X X X . . F . . *
* . . . . . . . . *
* . . . . . B . . . *
* * * * *
Enter Command: REDO
Redo stack kosong!
```

Gambar 6.16.2 Kasus REDO setelah aksi yang mengubah *state*

```
BNMO berada di posisi: (0,6)
Waktu: 9
Notifikasi: -
* * * * *
* . . . . . S . . *
* . . . . T . . X . . *
* . M . . . . . X . . *
* . . . . . X . . *
* . . . . X X X X . . *
* . X . . . . . . . *
* . X . . . . . C . . *
* . X X X . . F . . *
* . . . . . . . . *
* . . . . . B . . . *
* * * * *
Enter Command: MOVE EAST
BNMO berada di posisi: (0,7)
Waktu: 10
Notifikasi: -
* * * * *
* . . . . . S . . *
* . . . . T . . X . . *
* . M . . . . . X . . *
* . . . . . X . . *
* . . . . X X X X . . *
* . X . . . . . . . *
* . X . . . . . C . . *
* . X X X . . F . . *
* . . . . . . . . *
* . . . . . B . . . *
* * * * *
Enter Command: UNDO

BNMO berada di posisi: (0,6)
Waktu: 9
Notifikasi: -
* * * * *
* . . . . . S . . *
* . . . . T . . X . . *
* . M . . . . . X . . *
* . . . . . X . . *
* . . . . X X X X . . *
* . X . . . . . . . *
* . X . . . . . C . . *
* . X X X . . F . . *
* . . . . . . . . *
* . . . . . B . . . *
* * * * *
Enter Command: COOKBOOK

=====RESEP=====
Telur Rebus
BOIL - Telur Ayam
Ayam Geprek
MIX - Ayam Goreng - Sambal
Ayam Goreng
FRY - Ayam Tepung - Minyak Goreng
Ayam Tepung
MIX - Ayam Potong - Tepung
Ayam Potong
CHOP - Ayam Mentah
Sambal
CHOP - Cabai
Mie Goreng
FRY - Telur Mata Sapi - Mie Rebus - Minyak Goreng
Telur Mata Sapi
FRY - Minyak Goreng - Telur Ayam
Mie Rebus
BOIL - Mie Kering

BNMO berada di posisi: (0,6)
Waktu: 9
Notifikasi: -
* * * * *
* . . . . . S . . *
* . . . . T . . X . . *
* . M . . . . . X . . *
* . . . . . X . . *
* . . . . X X X X . . *
* . X . . . . . . . *
* . X . . . . . C . . *
* . X X X . . F . . *
* . . . . . . . . *
* . . . . . B . . . *
* * * * *
Enter Command: REDO
BNMO berada di posisi: (0,7)
Waktu: 10
Notifikasi: -
* * * * *
* . . . . . S . . *
* . . . . T . . X . . *
* . M . . . . . X . . *
* . . . . . X . . *
* . . . . X X X X . . *
* . X . . . . . . . *
* . X . . . . . C . . *
* . X X X . . F . . *
* . . . . . . . . *
* . . . . . B . . . *
* * * * *
Enter Command: █
```

Gambar 6.16.3 Kasus REDO setelah aksi yang tidak mengubah *state*

6.17 INVENTORY

Command ini digunakan untuk melihat apa saja makanan yang terdapat di dalam penyimpanan. Saat membeli makanan, jika makanan sudah sampai maka akan langsung dimasukkan ke inventory. Jika barang dipakai untuk memasak atau kadaluarsa maka makanan akan dihapus dari inventory.

```
Enter Command: INVENTORY

=====INVENTORY=====
Inventory Kosong
```

Gambar 6.17.1 Tampilan saat sedang tidak ada makanan dalam inventory

```
Enter Command: INVENTORY

=====INVENTORY=====
List Makanan di Inventory
(nama -- waktu sisa kadaluarsa)
1. (11) Ayam Mentah - 23 jam 55 menit
2. (12) Telur Ayam - 23 jam 59 menit
```

Gambar 6.17.2 Tampilan saat makanan sudah dimasukkan ke dalam inventory

```
Notifikasi:
1. Ayam Mentah kadaluarsa ... :(
2. Telur Ayam kadaluarsa ... :(
* * * * *
* . . . . S . . . . *
* . . . . T . . X . . *
* . M . . . . X . . *
* . . . . . X . . *
* . . . . X X X X . *
* . X . . . . . *
* . X . . . . . C . *
* . X X X . . F . . *
* . . . . . . . *
* . . . . . B . . . *
* * * * *
Enter Command: INVENTORY

=====INVENTORY=====
Inventory Kosong
```

Gambar 6.17.3 Tampilan saat ada makanan yang kadaluarsa dan dikeluarkan dari inventory

7 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	START	Memastikan program dapat dijalankan	Mengetikkan command START dan menginput nama user	START, nama user	Muncul tampilan awal peta, posisi dan waktu awal	Gambar 6.1.1
2	EXIT	Memastikan program dapat diterminasi	Mengetikkan command EXIT	EXIT	Program berhenti	Gambar 6.2.1
3	MOVE EAST	Memastikan mekanisme <i>move east</i> dapat dilakukan	Mengetikkan command MOVE EAST saat arah tujuan (bagian kanan simulator) tidak terhalangi	MOVE EAST	Posisi simulator berubah pada informasi posisi dan pada peta	Gambar 6.3.1
4	MOVE WEST	Memastikan mekanisme <i>move west</i> dapat dilakukan	Mengetikkan command MOVE WEST saat arah tujuan (bagian kiri simulator) tidak terhalangi	MOVE WEST	Posisi simulator berubah pada informasi posisi dan pada peta	Gambar 6.3.2
5	MOVE NORTH	Memastikan mekanisme <i>move north</i> dapat dilakukan	Mengetikkan command MOVE NORTH saat arah tujuan (bagian atas simulator) tidak terhalangi	MOVE NORTH	Posisi simulator berubah pada informasi posisi dan pada peta	Gambar 6.3.4
6	MOVE SOUTH	Memastikan mekanisme <i>move south</i> dapat dilakukan	Mengetikkan command MOVE SOUTH saat arah tujuan (bagian bawah simulator) tidak terhalangi	MOVE SOUTH	Posisi simulator berubah pada informasi posisi dan pada peta	Gambar 6.3.3
7	BUY	Memastikan proses <i>buy</i> dapat dilakukan	Mengetikkan command BUY saat pemain berada di lokasi "T" pada peta dan memilih makanan yang ingin dibuat	BUY, nomor makanan yang dipilih	Makanan berhasil dibeli jika input valid dan sampai di inventory sesuai waktu pengiriman	Gambar 6.4.1 6.4.2 6.4.3
8	MIX	Memastikan apakah proses <i>mix</i> dapat dilakukan	Mengetikkan command MIX saat pemain berada di lokasi "M" pada peta dan memilih makanan yang ingin dibuat	MIX, nomor makanan yang dipilih.	List makanan yang dapat diproses dengan cara <i>mix</i> , apabila bahan ada maka akan diproses	Gambar 6.5.1 6.5.2 6.5.3
9	CHOP	Memastikan apakah proses <i>chop</i> dapat dilakukan	Mengetikkan command CHOP saat pemain berada di lokasi "C" pada peta dan memilih makanan yang ingin dibuat	CHOP, nomor makanan yang dipilih.	List makanan yang dapat diproses dengan cara <i>chop</i> , apabila bahan ada maka akan diproses	Gambar 6.6.1 6.6.2 6.6.3

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
10	FRY	Memastikan apakah proses <i>fry</i> dapat dilakukan	Mengetikkan command FRY saat pemain berada di lokasi “F” pada peta dan memilih makanan yang ingin dibuat	FRY, nomor makanan yang dipilih.	List makanan yang dapat diproses dengan cara <i>fry</i> , apabila bahan ada maka akan diproses	Gambar 6.7.1 6.7.2 6.7.3
11	BOIL	Memastikan apakah proses <i>boil</i> dapat dilakukan	Mengetikkan command BOIL saat pemain berada di lokasi “B” pada peta dan memilih makanan yang ingin dibuat	BOIL, nomor makanan yang dipilih.	List makanan yang dapat diproses dengan cara <i>boil</i> , apabila bahan ada maka akan diproses	Gambar 6.8.1 6.8.2 6.8.3
12	DELIVERY	Memastikan list delivery sesuai dengan bahan yang dibeli	Membeli beberapa bahan makanan dan mengetikkan command DELIVERY	DELIVERY	Delivery list yang sesuai dengan bahan yang sudah dibeli	Gambar 6.9.1 6.9.2 6.9.3
13	CATALOG	Memastikan catalog makanan sesuai dengan file konfigurasi	Mengetikkan command CATALOG setelah command START	CATALOG	List makanan yang tampil sesuai dengan file konfigurasi dan terurut secara ID	Gambar 6.10.1
14	COOKBOOK	Memastikan resep makanan sesuai dengan file konfigurasi	Mengetikkan command COOKBOOK setelah command START	COOKBOOK	Daftar resep makanan yang tampil sesuai dengan file konfigurasi	Gambar 6.11.1
15	REKOMENDASI	Memastikan rekomendasi yang diberikan sesuai dengan bahan makanan yang dimiliki	Membeli beberapa bahan makanan dan mengetikkan command REKOMENDASI	REKOMENDASI	Rekomendasi yang muncul sesuai dengan bahan yang dimiliki	Gambar 6.12.2
16	INSERT KULKAS	Memastikan apakah bisa memasukkan barang ke dalam kulkas beserta <i>edge case</i> dimana inventory kosong atau kulkas penuh	Normal: inventory terisi dan kulkas kosong, menetik command INSERTKULKAS dan memilih id makanan yang ingin dimasukkan, indeks di kulkas, dan id di kulkas <i>edge</i> : inventory kosong atau kulkas penuh	INSERTKULKAS	Normal: diterima jika input id makanan valid <i>edge</i> : ditolak	Gambar 6.13.1 6.13.2 6.13.3
17	GET KULKAS	Memastikan apakah barang yang terdapat di kulkas bisa diambil	Normal: Kulkas tidak kosong, menetik command GETKULKAS, lalu memilih ID makanan di	GETKULKAS	Normal: makanan diambil dari kulkas dan dimasukkan	Gambar 6.13.4 6.13.5 6.13.6

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
		kembali beserta <i>edge case</i> dimana kulkas kosong atau id makanan salah	kulkas yang valid. <i>edge</i> : Kulkas kosong, mengetik command GETKULKAS		kedalam inventory <i>edge</i> : ditolak	
18	DISPLAY KULKAS	Memastikan apakah kulkas dapat ditampilkan beserta isinya dengan baik	Mengetikkan command DISPLAYKULKAS saat kulkas kosong ataupun terisi	DISPLAYKULKAS	Isi dari kulkas baik kosong maupun terisi ditampilkan ke layar	Gambar 6.13.7 6.13.8
19	WAIT	Memastikan agar fungsi wait berjalan sesuai fungsinya	Mengetikkan command WAIT X Y, dengan X adalah parameter jam dan Y adalah parameter menit	WAIT, Jumlah jam yang diinginkan, Jumlah menit yang diinginkan	Waktu simulator bertambah sesuai dengan <i>input</i>	Gambar 6.14.1
20	UNDO	Memastikan apakah undo dapat dijalankan dengan baik, sesuai dengan <i>edge case</i>	Command UNDO saat stack kosong, saat setelah aksi yang mengubah <i>state</i> dan aksi yang tidak mengubah <i>state</i>	UNDO	<i>State</i> ter-undo atau tidak berdasarkan kasus dengan baik	Gambar 6.15
21	REDO	Memastikan apakah redo hanya dapat dijalankan setelah undo, dan undo lalu aksi yang tidak mengubah <i>state</i> , tetapi tidak bisa saat undo lalu aksi yang mengubah <i>state</i>	Mengetikkan command UNDO setelah suatu aksi yang mengubah <i>state</i> , lalu mengetikkan command REDO, atau setelah UNDO melakukan suatu aksi, lalu command REDO	UNDO, REDO	Redo hanya bisa dipanggil untuk kasus setelah undo, dan aksi yang tidak mengubah <i>state</i>	Gambar 6.16
22	INVENTOR Y	Memastikan apakah bahan yang sudah diterima terdapat di dalam inventory	Membeli beberapa bahan, mengetikkan command INVENTORY apabila sudah ada notifikasi bahan sudah diterima	INVENTOR Y	Bahan yang sudah diterima terdapat di dalam inventory	Gambar 6.17.1 6.17.2 6.17.3

8 Pembagian Kerja dalam Kelompok

Pembagian Kerja didalam kelompok kami dibagi menjadi 2 bagian, yaitu ADT dan Fitur, untuk pembagian kerja adalah sebagai berikut:

STEI- ITB	IF2110-TB-I-01	Halaman 38 dari 45 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.		

NIM	Nama	Pembagian ADT	Pembagian Fitur
13521061	Alex Sander	ADT Matrix ADT Point	Peta Mekanisme waktu Waktu pengolahan makanan
13521092	Frankie Huang	ADT Tree ADT Mesin Karakter dan Mesin Kata	Inisiasi List Resep Cookbook
13521089	Kenneth Ezekiel	ADT List Statik ADT Queue	Inventory Pemesanan Bahan Makanan dan Delivery Kulkas Undo/Redo
13521042	Kevin John Wesley Hutabarat	ADT Stack ADT Time ADT Set	Pengolahan Makanan Undo/Redo Rekomendasi
13521094	Angela Livia Arumsari	ADT Makanan ADT Simulator	Simulator List Makanan Notifikasi Rekomendasi Makanan

9 Lampiran

9.1 Deskripsi Tugas Besar

BNMO (dibaca:Binomo) adalah sebuah robot game milik Indra dan Doni. Akhir-akhir ini, Indra baru saja menjalin hubungan spesial dengan perempuan bernama Siska Kol. Dan dalam dekat waktu, Indra akan mengajak Siska Kol kerumah untuk makan malam bersama Doni dan BNMO. Oleh karena itu, Indra meminta bantuan BNMO dan Doni untuk membantu mempersiapkan makan malam spesial tersebut. Saat itu juga, BNMO langsung tertarik untuk mengerjakan bagian masak karena ia sangat sering melihat video memasak diaplikasi tok tok dan sangat terngiang-ngiang dengan “mari kita cobaaa”. Namun, ada masalah. 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.

9.2 Notulen Rapat

PEMBAGIAN TUGAS

A. Timeline Kasar

Mulai	Selesai	Tugas
20 Oktober 2022	24 Oktober 2022	Kode/Laporan ADT
27 Oktober 2022	5 November 2022	Kode Program

STEI- ITB	IF2110-TB-I-01	Halaman 39 dari 45 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.		

1 November 2022	1 November 2022	Milestone 1
13 November 2022	20 November 2022	Laporan program

B. Meet 1 (Jumat, 21 Oktober 2022)

Asistensi 1 : Selasa, 25 Oktober 2022

Deadline : Rabu, 26 Oktober 2022

PJ	Tugas
Alex	ADT Matriks ADT Point
Frankie	ADT Tree ADT Mesin Karakter dan Mesin Kata
Kenneth	ADT List Statik ADT Queue + PrioQueue
Kevin	ADT Stack ADT Waktu
Livia	ADT Makanan ADT Simulator

C. Meet 2 (Kamis, 27 Oktober 2022)

Kebutuhan Program	ADT Yang Digunakan	PJ
Inisiasi	Mesin Karakter dan Mesin Kata	Frank
Simulator	Simulator	Livia
Makanan	Makanan List Statik	Livia
Resep	Tree List Statik Mesin Karakter dan Mesin Kata	Frank
Inventory	Priority Queue	Ken
Pemesanan Bahan Makanan dan Delivery	Matriks Priority Queue	Ken
Peta	Matriks Mesin Karakter dan Mesin Kata	Alex
Pengolahan Makanan	Mesin Karakter dan Mesin Kata Matriks Queue	Kevin

	List Statik	
Mekanisme Waktu	Simulator	Alex
Command-command Lain <ul style="list-style-type: none"> - Undo/Redo - Catalog - Cookbook 	Simulator Stack List Statik dan Makanan Tree	Kevin, Ken Livia Frank

Untuk variabel global:

- Waktu
- Inventory
- Simulator

Untuk variabel konstanta:

- List makanan
- Tree resep
- Peta

Dibutuhkan cepat: (2 hari)

- Inisiasi
- Makanan
- Inventori
- Peta
- Mekanisme waktu

D. Meet 3 (Senin, 31 Oktober 2021)

Asistensi 2 : 1 November 2022

Meet 4 -> OFFLINE @ SANSCO 5 NOVEMBER 2022

Bantu ngerjain fitur teman yang belum selesai.

MILESTONE

Tanggal		DD-MM-YYYY
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	
	d. ADT Simulator	100%
4.	Makanan	
	e. Membaca makanan dari file	100%
	f. ADT Makanan	100%
	g. Command CATALOG	100%
6.	Peta	
	h. Membaca peta dari file	100%
	i. Command MOVE NORTH/EAST/SOUTH/WEST	100%
7.	Mekanisme Waktu	
	j. ADT Waktu	100%
	k. Waktu bertambah seiring command yg valid	100%
8.	Laporan (50%)	100%

Laporan :

STEI- ITB	IF2110-TB-I-01	Halaman 42 dari 45 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.		

1. Ringkasan - Ken (DONE) tambahin
2. Spesifikasi - Livia (DONE)
3. ADT (DONE)
8. Pembagian Kerja (DONE)

Validasi dan error handling : Frankie

Notifikasi : Livia

TODO : Pindahin fitur utama ke program/lib (1 orang) -> Ken (DONE)

BONUS

BONUS	ADT	Nama
Kulkas	Matrix Makanan	Ken
Waktu Pengolahan Makanan	Makanan	Alex
Rekomendasi Makanan	Set	Kevin, Livia

Penjelasan Tambahan - per bagian

Program - belum samsek

Algoritma menarik - per bagian

Data Test - belum samsek

Test Script - belum samsek

! Konfigurasi makanan yang lebih banyak

9.3 Log Activity Anggota Kelompok

No.	Hari, tanggal	Deskripsi	Anggota
1.	Jumat, 21 Oktober 2022	Pembahasan mengenai pembagian tugas ADT.	Ken, Frank, Alex, Kevin, Livia

STEI- ITB	IF2110-TB-I-01	Halaman 43 dari 45 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.		

2.	Sabtu, 22 Oktober 2022	Mengerjakan ADT Tree, Mesin Karakter dan Mesin Kata.	Frank
3.	Sabtu, 22 Oktober 2022	Mengerjakan ADT Queue, Priority Queue, String dan List Statik.	Ken
4.	Senin, 24 Oktober 2022	Mengerjakan ADT Matriks dan Point.	Alex
5.	Senin, 24 Oktober 2022	Mengerjakan ADT Waktu dan Stack.	Kevin
6.	Selasa, 25 Oktober 2022	Mengerjakan ADT Makanan dan Simulator	Livia
7.	Selasa, 25 Oktober 2022	Asistensi 1	Ken, Frank, Alex, Kevin, Livia
8.	Rabu, 26 Oktober 2022	Pembahasan mengenai pembagian tugas fitur.	Ken, Frank, Alex, Kevin, Livia
9.	Kamis, 27 Oktober 2022	Mengerjakan bagian peta.	Alex
10.	Kamis, 27 Oktober 2022	Mengerjakan bagian inisiasi dan membuat file main.	Frank
11.	Jumat, 28 Oktober 2022	Menambahkan fungsi word to string dan mengerjakan bagian delivery dan inventory.	Ken
12.	Jumat, 28 Oktober 2022	Menambahkan ADT List Makanan.	Livia
13.	Sabtu, 29 Oktober 2022	Menambahkan file resep dan menambahkan fungsi string to word.	Frank
14.	Minggu, 30 Oktober 2022	Update ADT List Makanan, String, dan inventory	Ken
15.	Minggu, 30 Oktober 2022	Mengerjakan bagian mekanisme waktu	Alex
16.	Senin, 31 Oktober 2022	Pembahasan mengenai progress, pembagian tugas bonus.	Ken, Frank, Alex, Kevin, Livia
17.	Senin, 31 Oktober 2022	Mengerjakan bagian pengolahan makanan	Kevin
18.	Selasa, 1 November 2022	Menambahkan splash screen dan mengubah file konfigurasi.	Livia
19.	Selasa, 1 November 2022	Asistensi 2	Ken, Frank, Alex, Kevin, Livia
20.	Selasa, 1 November 2022	Menambahkan fungsi cookbook	Frank
21.	Selasa, 1 November 2022	Update ADT Time	Kevin
22.	Selasa, 1 November 2022	Menambahkan ADT State, Undo, dan Redo	Ken
23.	Jumat, 4 November 2022	Mengerjakan bagian notifikasi	Livia
	Jumat, 4 November 2022	Menyelesaikan bonus 1 kulkas	Ken

24.	Sabtu, 5 November 2022	Debugging	Ken, Frank, Alex, Kevin, Livia
25.	Minggu, 6 November 2022 – Sabtu, 18 November 2022	Melanjutkan debugging dan menyelesaikan laporan	Ken, Frank, Alex, Kevin, Livia
26.	Kamis, 10 November 2022	Menyelesaikan bonus rekomendasi makanan	Livia
27.	Rabu, 16 November 2022	Implementasi ulang ADT Undo Stack dan Redo Stack menjadi dinamik	Ken
28.	Jumat, 18 November 2022	Revisi dan implementasi fungsi baru untuk command BUY, CHOP, BOIL, FRY, dan MIX	Livia
29.	Sabtu, 19 November 2022	Penyelesaian Laporan	Ken, Frank, Alex, Kevin, Livia