

LAPORAN TUGAS BESAR

IF2110/Algoritma dan Struktur Data


BNMO

Dipersiapkan oleh:

K01 Kelompok D

Manuella Ivana Uli Sianipar	13521051
Melvin Kent Jonathan	13521052
Yobel Dean Christopher	13521067
Fajar Maulana Herawan	13521080
Jimly Firdaus	13521102

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-01-D</i>		<i>21</i>
		<i>Revisi</i>	-	<i>20 November 2022</i>

Daftar Isi

1 Ringkasan.....	3
2 Penjelasan Tambahan Spesifikasi Tugas	4
2.1 Kulkas.....	4
3 Struktur Data (ADT)	5
3.1 Mesin Karakter dan Mesin Kata.....	5
3.2 List Statik dan List Statik Makanan	5
3.3 Matriks	5
3.4 Makanan	5
3.5 Simulator	6
3.6 Waktu	6
3.7 Point	6
3.8 Priority Queue	6
3.9 Stack	6
3.10 Tree	6
3.11 String	7
4 Program Utama	7
5 Algoritma-Algoritma Menarik.....	8
5.1 Fitur Notifikasi	8
5.2 Tampilan Peta.....	9
6 Data Test	9
6.1 Peta	9
6.2 Makanan	10
6.3 Resep	12
7 Test Script	12
8 Pembagian Kerja dalam Kelompok	16
9 Lampiran	18
9.1 Deskripsi Tugas Besar 2.....	18
9.2 Notulen Rapat.....	18
9.3 Log Activity Anggota Kelompok.....	20

1 Ringkasan

A. Deskripsi Umum

Pada tugas besar ini, kami disuruh oleh Doni untuk membantunya membuat program simulasi untuk BNMO. Program simulasi ini berbasis CLI (*command-line interface*) yang dibuat dalam bahasa C. Program ini juga memanfaatkan Struktur Data (Abstract Data Type) untuk memudahkan pemrograman. Dalam simulasi ini secara umum terdapat pengolahan untuk membuat makanan, dari membeli bahan makanan hingga menghasilkan makanan jadi.

Tujuan dari simulasi yaitu menyimulasikan pengolahan makanan. Simulator akan memiliki peta untuk menyimulasikan lokasi BNMO. Inventory untuk menyimulasikan tempat penyimpanan makanan. Di dalam simulator dapat melakukan aksi-aksi tertentu sesuai tempat *adjacent*-nya yang bertujuan untuk mengolah makanan. Pertama, BUY yaitu untuk melakukan pemesanan bahan makanan. Pemesanan hanya dapat dilakukan di sekitar telepon dan terdapat *delivery time* untuk setiap bahan yang dipesan yang dapat dilihat di Delivery. Dengan melakukan aksi-aksi yang ada di dalam simulator seperti perpindahan dan aksi terkait pengolahan makanan (BUY, MIX, CHOP, FRY, BOIL) waktu akan bertambah. Dengan demikian, waktu *delivery time* berkurang hingga masuk ke dalam inventory. Setelah memiliki bahan makanan, simulator dapat melakukan aksi – aksi terkait pengolahan makanan yang ada pada tempat *adjacent*-nya masing-masing dengan menggunakan aksi MOVE untuk mengarahkan posisi simulator seperti FRY pada F. Salah satu aksi pengolahan makanan yaitu MIX, untuk mencampurkan dua atau lebih makanan. CHOP, untuk memotong suatu bahan makanan menjadi bahan yang terpotong. FRY, menggoreng satu atau lebih bahan makanan aksi ini selalu membutuhkan minyak goreng pada inventory, BOIL, merebus suatu bahan makanan. Aksi – aksi pengolahan berkaitan dengan resep yang sedang digunakan. Aksi lain yang ada pada simulator yaitu Undo/Redo, aksi UNDO bertujuan untuk membatalkan aksi yang dilakukan oleh simulator dan mengembalikan *state* aplikasi ke sebelum aksi itu dieksekusi. Aplikasi juga dapat melakukan aksi REDO untuk membatalkan aksi UNDO.

Untuk membuat simulator lebih menarik terdapat kulkas untuk menyimpan makanan. Waktu pengolahan makanan, umumnya pengolahan makanan dengan berbagai *command* yang sudah dijelaskan semuanya dijalankan dengan satu menit sedangkan, akan diminta menambahkan waktu yang dibutuhkan untuk memproses makanan tersebut. Rekomendasi makanan yang dapat dibuat, untuk menampilkan rekomendasi makanan yang dapat dibuat. Auto-BNMO, input berupa makanan yang ingin dimasak dan output merupakan langkah BNMO dalam mencapai state tersebut.

B. Isi Umum Laporan

Secara Umum, laporan Tugas Besar IF2110 Kelompok 6 terdiri dari 9 bagian, yaitu :

STEI- ITB	< IF2110-TB-01-D >	Halaman 3 dari 21 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. Bagian 1: Berisi deskripsi umum dan penjelasan simulator BNMO, pemaparan isi laporan secara umum, serta kesimpulan mengenai Tugas Besar IF2110.
2. Bagian 2: Berisi penjelasan tambahan mengenai spesifikasi-spesifikasi fitur yang belum rinci dari Deskripsi Tugas besar.
3. Bagian 3: Berisi penjelasan dari setiap Struktur Data yang digunakan serta kegunaannya sesuai dengan spesifikasi yang ada.
4. Bagian 4: Berisi penjelasan mengenai algoritma program utama, dimulai dari *start* hingga *exit*.
5. Bagian 5: Berisi penjelasan mengenai algoritma yang dianggap menarik oleh kelompok kami beserta dengan penggunaannya dalam program.
6. Bagian 6: Berisi penjelasan mengenai fitur-fitur yang akan dicoba dan hasil yang diharapkan dari program. Dalam bagian ini juga akan ditampilkan data testnya.
7. Bagian 7: Berisi Tabel Test Script yang terdiri dari fitur yang di tes, tujuan testing, langkah-langkah testing, input data test, hasil yang diharapkan serta hasil yang keluar.
8. Bagian 8: Berisi Tabel pembagian kerja dari Kelompok D K01.
9. Bagian 9: Berisi lampiran berupa Deskripsi lengkap Tugas Besar, notulen rapat, dan Log Activity dari setiap anggota kelompok.

C. Kesimpulan

Pada Tugas Besar IF2110, telah dimanfaatkan semua ADT yang pernah dikerjakan di praktikum serta penerapan pelajaran membuat ADT sendiri seperti Simulator dan banyak lagi. Simulasi BNMO membutuhkan cukup banyak ADT untuk menunjang efisiensi dari pemrograman sehingga setiap pemrogram mendapatkan pekerjaan beberapa ADT yang berbeda. Hasil dari Tugas Besar kelompok D ini cukup baik, semua program sudah berjalan seperti yang seharusnya.

2 Penjelasan Tambahan Spesifikasi Tugas

2.1 Kulkas

Kulkas dapat menyimpan makanan sehingga *Expired Time Makanan* yang ditaruh di *kulkas* tersebut tidak berkurang. Makanan yang disimpan di dalam *kulkas* dihapus dari *inventory*. *Kulkas* berupa matriks dengan ukuran 10x20. Peletakkan makanan pada *kulkas* dalam matriks tersebut sesuai dengan ukuran masing-masing makanan yang sudah didefinisikan pada ADT Makanan. Makanan diletakkan secara bersebelahan dan tidak menumpuk. Pada awalnya akan di cek terlebih dahulu apakah *kulkas* penuh atau tidak, apabila ada tempat di dalam *kulkas*, bahan makanan akan disimpan di list *kulkas* dan matriks *kulkas*. Cara memasukkannya adalah dengan melakukan *for looping* terhadap masing-masing elemen dari matriks *kulkas*. Elemen dari matriks *kulkas* tersebut kemudian akan dimasukkan ke sebuah fungsi yang akan mengembalikan *boolean* apakah barang dapat disimpan dalam elemen matriks tersebut memanjang ke kanan dan ke bawah. Apabila *kulkas* penuh, akan ditampilkan pesan “Kulkas penuh.” *Kulkas* dibuat dengan sistem FIFO (*First In First Out*), yaitu makanan yang dikeluarkan dari *kulkas* merupakan makanan yang pertama kali dimasukkan ke dalam *kulkas*. Begitu juga makanan yang akan dimasukkan ke dalam *kulkas* merupakan makanan dengan waktu kedaluwarsa yang terkecil di *inventory*.

STEI- ITB	< IF2110-TB-01-D >	Halaman 4 dari 21 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

3 Struktur Data (ADT)

Berikut beberapa ADT (Abstract Data Type) yang digunakan dalam pembuatan program beserta penjelasan.

3.1 Mesin Karakter dan Mesin Kata

ADT Mesin Karakter dan Mesin Kata dimanfaatkan untuk membaca masukan *user* dari *input stdin / keyboard*. ADT Mesin Karakter dan Mesin Kata juga digunakan untuk membaca dan mem-*parsing* konfigurasi file, berupa konfigurasi *map*, *makanan*, dan *resep*. Pada ADT Mesin Karakter dan Mesin Kata, sudah terdapat konstruktor, selektor, dan beberapa primitif tambahan. Definisi dan implementasi dari ADT Mesin Karakter dapat dilihat pada *charmachine.c* dan untuk definisi dan implementasi dari ADT Mesin Kata dapat dilihat pada *wordmachine.c*.

3.2 List Statik dan List Statik Makanan

ADT List Statik merupakan sebuah struktur data yang digunakan untuk memesan memori untuk menyimpan tipe elemen berupa integer. Pada ADT ini, terdapat konstruktor, selektor, dan beberapa primitif tambahan. Kami juga membuat modifikasi dari ADT List Statik ini agar dapat menampung tipe elemen selain integer, yakni ADT List Statik Resep yang menyimpan tipe elemen *tree* dan juga ADT List Statik Makanan (pada ADT Makanan) yang menyimpan tipe elemen makanan. Adapun operasi pada modifikasi list statik tersebut serupa, tetapi pada tipe elemen yang berbeda. ADT List Statik serta modifikasinya diimplementasikan pada *liststatik.c*, *liststatikresep.c*, dan *makanan.c*.

3.3 Matriks

ADT Matriks atau sering juga disebut array berdimensi 2 adalah struktur data yang mengacu pada sebuah/sekumpulan elemen yang di akses. Struktur data ini bersifat statik, yaitu ukuran maksimum memorinya ditentukan dari awal. Batas indeks baris dan kolom harus terdefinisi dengan pasti saat dideklarasikan dan tak dapat diubah-ubah. Sebelum matriks digunakan untuk menyimpan data, terlebih dahulu matriks harus dideklarasikan. Mendeklarasikan matriks artinya menentukan nama matriks, tipe datanya dan ukuran matriks. Matriks diimplementasikan pada *moves.c* dan *kulkas.c*.

3.4 Makanan

ADT Makanan pada program ini digunakan untuk mengoperasikan tipe Makanan pada program ini. Adapun komponen penyusun tipe Makanan adalah *id* (int), *nama* (string), *waktu_kedaluwarsa* (TIME), *lokasi_aksi* (string), *lama_pengiriman* (TIME), *size_x* (int), dan *size_y* (int). ADT ini juga dilengkapi dengan struktur data ListMakanan serta konstruktor dan selektornya. ADT Makanan diimplementasikan pada *makanan.c*.

3.5 Simulator

ADT Simulator merupakan struktur data yang digunakan untuk menyimpan *state* dari program. Komponen penyusun dari struktur data Simulator adalah nama (string), posisi (POINT), waktu (TIME), inventory (PrioQueue), delivery (PrioQueue), dan kulkas (Matrix). ADT ini dilengkapi dengan konstruktor, selektor, dan beberapa operasi primitif lain. ADT Simulator diimplementasikan pada *makanan.c*

3.6 Waktu

ADT Waktu pada program ini menyimpan 3 komponen penyusun, yakni DD (int), HH (int), MM (int). DD menyimpan jumlah hari dari sebuah waktu dengan aturan $0 \leq DD \leq 29$. HH menyimpan jumlah jam dari sebuah waktu dengan aturan $0 \leq HH \leq 23$. MM menyimpan jumlah menit dari sebuah waktu dengan aturan $0 \leq MM \leq 59$. ADT ini dilengkapi dengan konstruktor, selektor, dan beberapa operasi primitif lain seperti validasi, komparasi, konversi, dan manipulasi. ADT Waktu diimplementasikan pada *waktu.c*

3.7 Point

ADT Point pada program ini menyimpan 2 komponen penyusun, yakni X (int), dan Y (int). X mewakili absis, sedangkan Y mewakili ordinat dari sebuah titik. ADT ini dilengkapi dengan konstruktor dan selektor. ADT Point diimplementasikan pada *point.c*

3.8 Priority Queue

ADT Priority Queue adalah sebuah ADT yang menggunakan metode sama seperti list, namun pada Priority Queue terdapat sifat FIFO (First In First Out) yang artinya elemen yang masuk pertama ke Priority Queue akan selalu diproses pertama dan diurutkan berdasarkan waktu. Priority Queue digunakan untuk mengimplementasikan pesanan bahan makanan yang masuk dan bahan makanan dalam inventory, ADT ini dipilih karena sesuai dengan konsep pesanan yaitu pesanan yang datang duluan juga akan dilayani duluan dan makanan yang masuk ke inventory, tetapi terurut berdasarkan waktu. ADT primitif Priority Queue diimplementasikan pada *prioqueue.c*. Sedangkan ADT inventory dan delivery diimplementasikan pada *inventory_delivery.c*

3.9 Stack

ADT Stack adalah sebuah ADT yang menggunakan metode sama seperti list, namun pada Stack terdapat sifat LIFO (Last In First Out) yang artinya elemen yang masuk terakhir ke Stack akan selalu diproses pertama. Stack digunakan untuk mengimplementasikan undo redo dalam simulator. State Simulator yang sekarang yaitu elemen teratas pada Undo Stack. Kemudian, state sebelumnya yaitu elemen di bawah elemen teratas. ADT primitif Stack diimplementasikan pada *stack.c* sedangkan ADT Undo Redo diimplementasikan pada *undoredo.c*.

3.10 Tree

STEI- ITB	< IF2110-TB-01-D >	Halaman 6 dari 21 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.		

ADT Tree digunakan untuk menyimpan resep makanan. Tree terdiri atas *node-node*. Misalkan ada sebuah *node A*. *Node B* berada di bawah *node A* dan terhubung. *Node B* adalah *child* dari *node A* dan *node A* adalah *parent* dari *node B*.

- ROOT(n) : nilai yang tersimpan pada *node* tersebut
- NEXTNODE(n) : menunjuk ke *node* elemen dengan *parent* yang sama
- CHILDNODE(n) : menunjuk ke *child* pertama dari sebuah *node*

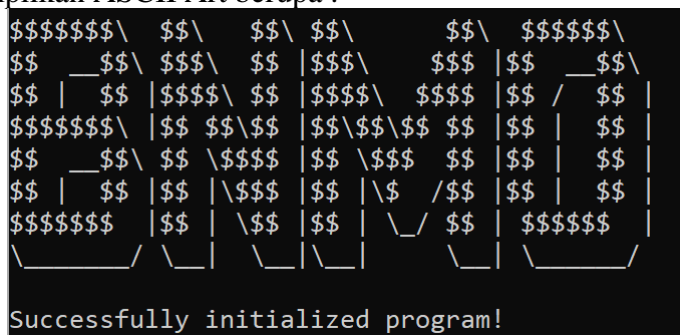
Struktur ini tepat untuk menyimpan resep karena tiap makanan bisa saja terdiri dari makanan-makanan lain.

3.11 String

ADT String merupakan struktur data yang terdiri atas 2 bagian yaitu *content* dan *length*. Bagian *content* bertindak sebagai *array of characters* yang disimpan dalam bentuk *char pointers*. Untuk bagian *length* menandakan panjang dari *array of characters* yang digunakan. Konsep *content* pada ADT String ini memiliki ukuran yang dinamis sehingga ukuran *array of characters* yang hendak digunakan dapat disesuaikan. ADT String dibuat karena untuk memudahkan pengaksesan kata dalam program karena ADT String memberikan kita kemampuan untuk dapat membaca sebuah kata sebagai 1 kesatuan, berbeda dengan kata yang dihasilkan pada ADT Mesin Kata, dimana kata disimpan dalam bentuk *TabWord* sehingga tidak bisa mengakses keseluruhan kata sebagai 1 kesatuan. Pada ADT String, sudah terdapat konstruktor, selektor, dan beberapa primitif tambahan untuk menunjang *use case* dari ADT String yang telah dibuat. Defisini dan implementasi dari ADT String dapat dilihat pada *string.c*

4 Program Utama

Program utama simulator BNMO terdapat pada file *main.c* kami. Ketika dijalankan, pertama kali program akan menampilkan ASCII Art berupa :



Gambar 4.1. Tampilan *splash screen* ASCII Art.

Selanjutnya program akan mengeluarkan pesan "Successfully initialized program!" yang menandakan bahwa program sudah siap digunakan oleh *user*. Selanjutnya, *user* akan diminta untuk melakukan input *command* berupa START atau EXIT. Jika *user* menginput EXIT, maka program akan terminasi dan mengeluarkan pesan "Bye... See you again next time!" yang menandakan bahwa program sudah berhasil diterminasi. Jika *user* menginput START, maka program akan mengeluarkan pesan "Starting Program..." kemudian *user* akan diminta untuk menginput nama untuk simulator. Selanjutnya program akan mencetak informasi berupa:

1. Nama Simulator

STEI- ITB	< IF2110-TB-01-D >	Halaman 7 dari 21 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. Posisi Simulator
3. Waktu Sekarang
4. Notifikasi
5. Peta

Setelah semua itu, program akan kembali meminta masukan berupa *command* yang valid dari *user*. Jika *user* menginput *command* yang tidak terdaftar pada simulator maka pesan validasi akan dilempar oleh program dan program akan kembali meminta input *command* dari *user*. Program utama akan memanggil prosedur / fungsi yang berkorespondensi dengan inputan *user* saat itu dan program ini akan selalu berada pada *state running* sampai *command* EXIT diberikan oleh *user* kepada program.

```
Successfully initialized program!
COMMAND (START/EXIT): START
Starting Program...
Nama simulator: BNMO
BNMO di posisi: (5,4)
Waktu : 0.0
Notifikasi : -
* * * * *
*           X           *
*      T       X       M      *
*           S       X       *
*      X X X X   X X X   *
*   X           *           *
*   X           C           *
*   X X X       F           *
*   X X X X X X X X X X X *
*      M       X           *
* X X X X       X       T   *
*           X           *
*   B       X           *
*           S           *
* * * * *
```

Gambar 4.2. Contoh Tampilan Program Utama.

5 Algoritma-Algoritma Menarik

5.1 Fitur Notifikasi

Fitur notifikasi menggunakan algoritma yang cukup menarik bagi kelompok kami karena tantangan yang kami hadapi dalam menyusun fitur ini. Perlu diperhatikan bahwa notifikasi haruslah dapat ditampilkan di satu kolom spesifik untuk notifikasi dan berlaku untuk segala jenis notifikasi, baik yang timbul akibat perubahan pada delivery, inventory, undo, maupun redo. Tantangan ini semakin berat karena notifikasi untuk setiap event haruslah unik dan memerhatikan item yang berubah pada simulator. Adapun untuk menentukan jenis notifikasi yang harus ditampilkan program harus mendeteksi jenis perubahan yang terjadi pada simulator. Perubahan yang terjadi ini dideteksi oleh prosedur/fungsi yang terpisah-pisah sehingga

STEI- ITB	< IF2110-TB-01-D >	Halaman 8 dari 21 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.		

menyulitkan kami untuk menyatukan notifikasi di satu tempat dan ditampilkan secara berbarengan. Selain itu, notifikasi undo/redo juga harus dibuat secara terpisah karena dua faktor. Faktor pertama adalah kalimat notifikasi yang ditampilkan haruslah berbeda dari kasus ketika program dijalankan secara normal. Faktor kedua adalah perubahan state program dalam pemanggilan undo/redo tidak melewati prosedur/fungsi yang dapat mendeteksi perubahan simulator karena fitur undo/redo bersifat langsung melakukan *overwrite* state simulator.

Setelah berdiskusi dengan *stakeholder* fungsi/prosedur yang terlibat, kami akhirnya memutuskan untuk membuat sebuah 10 ListMakanan unik yang dapat menampung item yang berubah pada simulator ketika suatu command dijalankan. Ketika terdapat perubahan pada simulator, prosedur/fungsi yang berlainan akan mendeteksi perubahan tersebut secara terpisah dan melakukan `appendListMakanan` pada ListMakanan terkait. Selanjutnya, fitur notifikasi akan dipanggil untuk mengiterasi setiap ListMakanan. Apabila ListMakanan tertentu tidak kosong, fitur notifikasi akan melakukan iterasi pada ListMakanan tersebut dan menampilkan notifikasi sesuai dengan kasus yang terjadi. Lalu, ListMakanan tersebut akan dikosongkan agar kemudian dapat diisi kembali dengan perubahan baru yang terjadi pada simulator.

5.2 Tampilan Peta

Bagian tampilan peta pada program menggunakan algoritma yang menarik bagi kelompok kami karena improvisasi yang harus kami lakukan untuk memenuhi spesifikasi. Matriks pada peta dibaca oleh fungsi parser dan file konfigurasi peta sudah menyertakan lokasi awal simulator. Ketika terjadi command MOVE harus dilakukan validasi dan juga pengubahan lokasi simulator. Namun, ADT Simulator kami tidak menyimpan *state* dari matrix peta dan matrix peta hanya disimpan di variabel pada main program. Workaround yang kami lakukan adalah dengan membaca lokasi S hanya ketika pertama kali inisialisasi peta. Parser *me-locate* simulator dan point yang didapat diteruskan ke simulator sebagai lokasi awal. Setelah membaca lokasi awal, parser akan menghapus simbol 'S' dari peta sehingga matrix yang tersimpan pada main program tidak mengandung simbol 'S'. Untuk menampilkan peta pada keberjalanan program, program akan membaca point lokasi dari simulator dan kemudian diteruskan ke matrix. Untuk keberjalanan perintah move, program akan mengkomparasi point lokasi simulator dan index pada matrix agar dapat mengidentifikasi adanya *obstacle*.

6 Data Test

6.1 Peta

Data test untuk Peta disimpan dalam format .txt file (textfile). Fitur yang akan di tes dengan data test ini adalah fitur baca peta. Fitur baca peta akan membaca dan memarsing textfile ke dalam bentuk matriks untuk mempermudah melacak keberadaan tiap - tiap komponen yang ada di dalam peta, seperti lokasi S, T, B, C, F, M, X.

Berikut tampilan struktur peta:

Ukuran :	15 baris x 15 kolom
----------	---------------------

STEI- ITB	< IF2110-TB-01-D >	Halaman 9 dari 21 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.		

15 15 ##### #####X##### ##T####X###M### ####S##X##### ####XXXX#XXX### #X##### #X#####C## #XXX##F##### #XXXXXXXXXXXX### ##M###X##### XXXX##X####T### #####X##### #B####X##### ##### #####S#####	# : Daerah yang dapat dilalui / dilewati oleh Simulator. X : Tembok. Tidak bisa dilewati oleh Simulator. T : Telepon. Tempat Simulator dapat melakukan aksi BUY. Tidak bisa dilewati oleh Simulator. S : Posisi Simulator. C : Chop. Tempat Simulator dapat melakukan aksi CHOP. Tidak bisa dilewati oleh Simulator. F : Fry. Tempat Simulator dapat melakukan aksi FRY. Tidak bisa dilewati oleh Simulator. M : Mix. Tempat Simulator dapat melakukan aksi MIX. Tidak bisa dilewati oleh Simulator. B : Boil. Tempat Simulator dapat melakukan aksi BOIL. Tidak bisa dilewati oleh Simulator.
--	---

6.2 Makanan

Data test untuk Makanan disimpan dalam format .txt file (textfile). Fitur yang akan di tes dengan data test ini adalah fitur catalog, pemesanan, dan pengolahan pada makanan. Untuk fitur catalog, program akan menampilkan daftar makanan pada konfigurasi yang merupakan hasil parsing dari textfile konfigurasi makanan. Untuk fitur pemesanan program akan menampilkan list makanan yang sedang dalam pengantaran beserta lama waktu pengantaran. Untuk pengolahan makanan, program akan mengatur lama pemrosesan makanan berdasarkan konfigurasi makanan.

Berikut tampilan struktur makanan:

12 1 Ayam Mentah 1 0 0 0 0 3 Buy 2 Ayam Potong 0 2 0 0 0 0	Menandakan jumlah makanan : 12 ID Makanan Nama Makanan Expiry Time Lama Pemrosesan Makanan Aksi
---	--

STEI- ITB	< IF2110-TB-01-D >	Halaman 10 dari 21 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.		

Chop 3 Tepung 0 2 0 0 0 2 Buy 4 Ayam Gulai 1 0 0 0 0 15 Mix 5 Minyak Goreng 0 3 0 0 0 5 Buy 6 Ayam Goreng 1 0 0 0 0 10 Fry 7 Air Putih 2 0 0 0 0 2 Buy 8 Bumbu Gulai 1 0 0 0 1 0 Mix 9 Rempah 1 0 0 0 0 3 Buy 10 Kecap 2 0 0 0 0 3 Buy 11 Ayam Kecap 0 8 0	
---	--

0 0 30 Mix 12 Ayam Tepung 0 9 0 0 0 15 Mix	
--	--

6.3 Resep

Data test untuk Resep disimpan dalam format .txt file (textfile). Fitur yang akan di tes dengan data test ini adalah fitur cookbook. Untuk fitur cookbook, program akan menampilkan resep apa saja yang terdapat pada simulator, yang merupakan hasil dari parsing filetext konfigurasi resep.

Berikut tampilan struktur resep:

6 2 1 1 12 2 2 3 6 2 2 5 8 2 7 9 4 2 6 8 11 3 6 7 10	Menandakan jumlah resep : 6 (pada baris pertama) Angka pertama merupakan ID Makanan yang hendak dibuat, angka kedua merupakan jumlah bahan makanan yang hendak digunakan, angka ketiga dan seterusnya merupakan ID Makanan yang harus ada untuk membuat makanan tersebut. (pada baris kedua hingga baris terakhir).
--	--

7 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Inisiasi Program	Memastikan program dapat dijalankan dengan benar dan dapat diterminasi.	1. Menginput <i>command</i> START. 2. Menginput <i>command</i> EXIT.	-	Program dapat menjalankan <i>splash screen</i> dan dapat menerima <i>commands</i> selanjutnya.	Program mengeluarkan <i>splash screen</i> dan program meminta <i>commands</i> dari user untuk aksi selanjutnya.
2	Simulator	Memastikan segala perubahan yang terjadi pada program akibat <i>side effect</i> dari	1. Memulai program 2. Melakukan prosedur MOVE, BUY, pengolahan makanan, WAIT.	<i>All Commands</i>	State dari program dapat berubah, yakni pada elemen yang berkoresponden	State dari program berubah, yakni pada elemen yang berkoresponden dengan <i>command</i>

STEI- ITB	< IF2110-TB-01-D >	Halaman 12 dari 21 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.		

No.	Fitur yang Dites	Tujuan Testing	Langkah- Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
		<i>commands</i> dapat disimpan dan mengubah <i>state</i> dari program.	KULKAS. 3. Mengecek perubahan waktu, lokasi, Prioqueue inventory, Prioqueue delivery, dan Kulkas untuk setiap <i>command</i> atau prosedur terkait.		dengan <i>command</i> atau prosedur yang dilakukan.	atau prosedur yang dilakukan.
3	Makanan	Memastikan fitur <i>search by ID</i> pada makanan dapat digunakan untuk kebutuhan program.	1. Memanggil fungsi <i>search by ID</i> . 2. Mengetes driver.	-	Fungsi <i>search by ID</i> dapat berfungsi dengan baik.	Fungsi <i>search by ID</i> mengembalikan data makanan berdasarkan <i>ID</i> yang diberikan.
4	Inventory	Memastikan program dapat menyimpan item makanan di Prioqueue Inventory	1. Memulai program 2. Menginput <i>command</i> BUY 3. Menginput <i>command</i> WAIT <jam> <menit> sebanyak waktu <i>time delivery</i> 4. Menginput <i>command</i> INVENTORY	<i>Command:</i> INVENTORY	Program dapat menampilkan item yang telah masuk dari antrian pengiriman serta waktu <i>expired</i> makanan	Program menampilkan item yang telah masuk dari antrian pengiriman serta waktu <i>expired</i> makanan
5	Pemesanan	Memastikan program dapat membeli makanan yang memiliki lokasi aksi "Buy" dan memasukannya ke dalam prioqueue Delivery	1. Memulai program 2. MOVE menuju adjacent lokasi T (telepon) 3. Menginput <i>command</i> BUY 4. Menginput <i>command</i> 1 – sekian berdasarkan item yang ingin dibeli dan 0 untuk keluar 5. Menginput <i>command</i> DELIVERY untuk melihat apakah item sudah masuk ke Prioqueue Delivery	<i>Command:</i> BUY	Program dapat menampilkan item yang dapat dibeli (lokasi aksi "Buy") serta dapat menerima item yang ingin dibeli dan memasukannya ke dalam Prioqueue Delivery.	Program menampilkan item yang dapat dibeli serta dapat menerima item yang ingin dibeli user dan memasukannya ke dalam Prioqueue Delivery.

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
6	Delivery	Memastikan program dapat menyimpan item makanan di Prioqueue delivery	1. Memulai program 2. Menginput <i>command</i> BUY 3. Menginput <i>command</i> DELIVERY	<i>Command:</i> BUY & DELIVERY	Program dapat menampilkan item yang berada pada antrian pengiriman secaraurut berdasarkan sisa waktu pengiriman sekaligus menampilkan sisa waktu pengiriman.	Program menampilkan item yang berada pada antrian pengiriman secaraurut berdasarkan sisa waktu pengiriman sekaligus menampilkan sisa waktu pengiriman.
7	Peta	Memastikan program dapat membaca dan menggunakan konfigurasi peta dari filetext, memastikan gerak simulator tidak bisa terjadi <i>collision</i> .	1. Memulai program. 2. Menginput <i>command</i> START. 3. Menginput <i>command</i> MOVE <arah>.	Data test pada bagian 6.1.	Program dapat menampilkan peta, simulator tidak dapat keluar dari batasan peta dan tidak bisa menabrak properti peta seperti X, B, C, S, F, T.	Program menampilkan peta, simulator hanya dapat bergerak di dalam batasan peta dan tidak bisa terjadi <i>collision</i> antara simulator dengan properti peta.
8	Pengolahan Makanan	Memastikan program dapat melakukan pengolahan makanan.	1. Memulai program. 2. Menginput <i>command</i> START. 3. Menginput <i>command</i> BUY dan membeli beberapa item. 4. Bergerak kearah tempat Mix (M). 5. Memilih makanan yang akan di-mix.	Data test pada bagian 6.2 dan 6.3.	Program menampilkan apakah makanan berhasil dibuat atau tidak. Jika tidak, program akan menunjukkan bahan yang kurang untuk membuat makanan yang diinginkan.	Program menampilkan apakah makanan berhasil dibuat atau tidak. Jika tidak, program akan menunjukkan bahan yang kurang untuk membuat makanan yang diinginkan.
9	Mekanisme Waktu	Memastikan <i>commands</i> WAIT X Y dapat diproses oleh program.	1. Memulai program. 2. Menginput <i>command</i> START. 3. Menginput <i>command</i> WAIT <jam> <menit>.	<i>Command:</i> WAIT 1 30	Program dapat membaca argumen dari WAIT dan simulator tidak melakukan apa-apa selama 1 jam 30 menit.	Waktu pada simulator bertambah, waktu expiry berkurang, waktu delivery berkurang sebanyak argumen yang diberikan <i>command</i> WAIT 1 30

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
10	Catalog & Cookbook	Memastikan program dapat membaca file konfigurasi dan dapat menampilkan list makanan apa saja yang ada pada program dan resep apa saja yang dapat dibuat.	1. Memulai program. 2. Menginput <i>command</i> START. 3. Menginput <i>command</i> CATALOG. 4. Menginput <i>command</i> COOKBOOK.	Data test pada bagian 6.2 & 6.3	Program dapat menampilkan keterangan semua list makanan yang ada dan semua resep yang dapat dibuat.	Program menampilkan semua makanan yang ada dari <i>command</i> CATALOG dan menampilkan semua resep dari <i>command</i> COOKBOOK.
11	Validasi & Error Handling	Memastikan program dapat memberikan pesan kesalahan jika ada <i>commands</i> yang tidak sesuai.	1. Memulai program. 2. Menginput <i>command</i> START. 3. Menginput <i>command</i> yang tidak valid.	<i>Command:</i> MOVE UP	Program dapat menampilkan pesan kesalahan input dan meminta input ulang dari <i>user</i> .	Program memberikan pesan kesalahan dan mengulangi meminta input dari <i>user</i> .
12	Notifikasi	Memastikan program dapat menampilkan notifikasi jika terdapat perubahan yang terjadi pada simulator.	1. Memulai program. 2. Menginput <i>command</i> BUY dan dilakukan untuk beberapa item. 3. Melakukan MOVE agar terjadi penambahan waktu 4. Melakukan UNDO dan REDO	<i>Command:</i> MOVE & BUY	Program dapat menampilkan notifikasi untuk <i>expired</i> makanan, tibanya makanan di inventory, perubahan untuk UNDO dan REDO.	Program menampilkan notifikasi sesuai dengan <i>event</i> yang terjadi.
13	Undo/Redo	Memastikan program dapat mengembalikan state program ke sebelumnya dan membatalkan <i>command</i> UNDO	1. Memulai program 2. Menginput BUY 3. Menginput <i>command</i> UNDO	<i>Command:</i> UNDO / REDO	Program dapat mengembalikan state simulator sebelumnya	Program mengembalikan state simulator sebelumnya
14.	Kulkas	Memastikan program dapat mengeluarkan Makanan dari	1. Memulai program 2. Menginput KULKAS	<i>Command:</i> KULKAS	Program dapat meletakkan Makanan ke dalam kulkas dari	Program dapat meletakkan Makanan ke dalam kulkas dari

No.	Fitur yang Dites	Tujuan Testing	Langkah- Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
		inventory dan diletakkan ke dalam kulkas, dan sebaliknya.			inventory dan mengeluarkannya kembali ke inventory.	inventory dan mengeluarkannya kembali ke inventory.

8 Pembagian Kerja dalam Kelompok

FITUR	NAMA ANGGOTA
BAGIAN ADT	
SEDERHANA (Waktu, Point)	Melvin Kent Jonathan
SIMULATOR	Melvin Kent Jonathan
MAKANAN	Melvin Kent Jonathan
LIST STATIK	Melvin Kent Jonathan
MATRIKS	Yobel Dean Christopher
PRIQUEUE & STACK	Fajar Maulana H
TREE	Manuella Ivana Uli Sianipar
MESIN KARAKTER & MESIN KATA	Jimly Firdaus
STRING	
PROGRAM	
INISIASI	Jimly Firdaus, Melvin Kent Jonathan
NOTIFIKASI	Fajar Maulana H, Melvin Kent Jonathan
SIMULATOR	Melvin Kent Jonathan
MOVE	Yobel Dean Christopher
PEMESANAN MAKANAN	Melvin Kent Jonathan
DELIVERY	Fajar Maulana H
PENGOLAHAN	Manuella Ivana Uli Sianipar
KULKAS	Yobel Dean Christopher
UNDO/REDO	Fajar Maulana H

PARSER	Jimly Firdaus
PETA	
CATALOG & COOKBOOK	
MAIN PROGRAM	Jimly Firdaus, Melvin Kent Jonathan
LAPORAN	
BAB 1	Fajar Maulana H
BAB 2	Yobel Dean Christopher
BAB 3	Semua anggota kelompok
BAB 4	Jimly Firdaus
BAB 5	Melvin Kent Jonathan
BAB 6	Jimly Firdaus
BAB 7	Semua anggota kelompok
BAB 8	Semua anggota kelompok
BAB 9	Semua anggota kelompok

9 Lampiran

9.1 Deskripsi Tugas Besar 2



“BNMO sedang memasak mengikuti program simulasi yang telah direkam Doni”

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 ke rumah 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](#) di aplikasi toktok dan sangat tergiang-giang 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

Rapat 1 – 22 Oktober 2022

- Membahas ADT yang dibutuhkan

STEI- ITB	< IF2110-TB-01-D >	Halaman 18 dari 21 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.		

- Memabahas pembagian tugas untuk pembuatan ADT

Asistensi 1 –28 Oktober 2022

Catatan Asistensi:

Q1 : Waktu print out buat apa?

R : - Buat user liat saja, tapi overall operasi2
di yang lain berupa count down

Q2 : Input nama file atau dari program langsung?

R : - Dibebaskan, data dari file konfigurasi juga dibuat sendiri asalkan spesifikasi minimumnya terpenuhi

Q3 : ADT tree bagaimana?

R : - Harus n ary dan dinamik

Q4 : Untuk waktu pengolahan makanan yang bonus bagaimana?

R : - Boleh langsung spontan berubah atau lewat queue

- Boleh tambahin properti di ADT makanan

- Boleh tambahin queue di property simulator (perlu konfirmasi lagi) Ga bisa keluar dari bintang (divalidasi aja)

- Properti lokasi aksi itu untuk memperoleh makanannya

Q5 : Untuk UNDO/REDO bagaimana ?

R : Dibebasin, yang penting dijelaskan di laporan kenapa approach yang diambil gini.

Saran : yang disimpan ke stacknya itu commandnya aja, biar ngak berat di memory buat nyimpan state. Balik lagi ke kalian enaknya mau gimana.

Rapat 2 – 30 Oktober 2022

- Membahas kelengkapan ADT
- Memabahas rancangan program utama
- Melakukan pembagian tugas

Asistensi 2 – 2 November 2022

Catatan Asistensi:

Progress sesuai dengan lembar Milestone, tetapi di bagian main harus diperbaiki karena ada error/conflict. Tambahan progress di luar lembar Milestone ada di ADT Tree dan juga resep yang sudah dibuat.

Rapat 3 – 18 November 2022

- Membahas keberjalanan program utama
- Membagi tugas untuk penyelesaian akhir

STEI- ITB	< IF2110-TB-01-D >	Halaman 19 dari 21 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.		

9.3 Log Activity Anggota Kelompok

LOG ACTIVITY		
Tanggal	Nama	Deskripsi
22/10/2022	Seluruh anggota	Bagi tugas ADT
23/10/2022	Manuella Ivana Uli Sianipar	Nenvyar ADT Tree
24/10/2022	Fajar Maulana H	Membuat ADT Stack dan Prioqueue
24/10/2022	Melvin Kent Jonathan	Membuat ADT Sederhana (Waktu, Point, Makanan, Simulator) dan ADT List Statik
24/10/2022	Jimly Firdaus	Membuat ADT Mesin Kata, Mesin Karakter, Parser Peta
26/10/2022	Jimly Firdaus	Membuat ADT String
27/10/2022	Jimly Firdaus	Membuat Parser Makanan dan memperbaiki <i>bug</i> pada ADT String
29/10/2022	Yobel Dean Christopher	Membuat ADT Matriks
30/10/2022	Seluruh anggota	Kerja kelompok untuk dekomposisi program.
30/10/2022	Yobel Dean Christopher	Membuat Move
30/10/2022	Manuella Ivana Uli Sianipar	Memperbaiki ADT lainnya
31/10/2022	Fajar Maulana	Membuat ADT Undo/Redo dan Inventory/Delivery
31/10/2022	Jimly Firdaus	Membuat prototype main
31/10/2022	Manuella Ivana Uli Sianipar	Membuat pengolahan
1/11/2022	Jimly Firdaus	Membuat parser resep, cookbook, catalog
2/11/2022	Jimly Firdaus	<i>Fix bug</i> pada input <i>command</i> , memperbaiki prototype main, membuat <i>parsing command</i> untuk WAIT X Y, menambahkan validasi input.
4/11/2022	Manuella Ivana Uli Sianipar	Memperbaiki ADT lainnya dan memperbaiki pengolahan
7/11/2022	Jimly Firdaus	Menambahkan validasi <i>command</i> MOVE, membuat fitur mengecek <i>adjacent</i> simulator, membuat test program.
12/11/2022	Manuella Ivana Uli Sinipar	Membuat fungsi untuk mengubah ListResep (berbentuk list dinamis) ke ListStatikResep (berbentuk list static dengan elemen tree)
13/11/2022	Melvin Kent Jonathan	Membuat prosedur pemesanan makanan, memulai konstruksi main program dari fungsi dan prosedur yang ada.
14/11/2022	Jimly Firdaus	<i>Resolve include</i> pada main program, menambahkan <i>exit phase</i> pada main program.
17/11/2022	Melvin Kent Jonathan	Memisahkan prosedur move dari ADT

STEI- ITB	< IF2110-TB-01-D >	Halaman 20 dari 21 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.		

		Matrix
18/11/2022	Jimly Firdaus	Mengubah state peta dan prosedur MOVE, menambahkan <i>command</i> INVENTORY, <i>fix bug</i> untuk semua input.
18/11/2022	Seluruh anggota	Kerja kelompok untuk finalisasi dan debugging program.
19/11/2022	Melvin Kent Jonathan, Fajar Maulana H	Membuat fitur notifikasi secara terpisah.
19/11/2022	Yobel Dean Christopher	Membuat bonus yaitu Kulkas.
19/11/2022	Jimly Firdaus	<i>Fix bug</i> pada driver ADT Mesin Kata dan Mesin Karakter, melengkapi driver ADT String, melakukan perubahan untuk input di beberapa fitur, menambahkan README <i>project</i> .
19/11/2022	Melvin Kent Jonathan	Memastikan seluruh fungsi waktu berjalan pada setiap prosedur/fungsi terkait

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

No. Kelompok/Kelas : D / K01
Nama Kelompok : Kelompok D
Anggota Kelompok (Nama/NIM) :
1. Manuella Ivana Uli Sianipar / 13521051
2. Melvin Kent Jonathan / 13521052
3. Yobel Dean Christopher / 13521067
4. Fajar Maulana Herawan / 13521080
5. Jimly Firdaus / 13521102

Asisten Pembimbing : Rezda Abdullah Fachrezzi - 13519194

Asistensi I

Tanggal : 28 Oktober 2022	Catatan Asistensi:
Tempat : ZOOM Meeting	

Kehadiran Anggota Kelompok:

No
NIM
Tanda tangan

1
13521051



2
13521052



3
13521067



4
13521080



5
13521102



Waktu print out buat apa?

- Buat user liat saja, tapi overall operasi di yang lain berupa count down

Input nama file atau dari program langsung?

- dibebaskan, data dari file konfigurasi juga dibuat sendiri asalkan spesifikasi minimumnya terpenuhi

ADT tree bagaimana?

- Harus n ary dan dinamik

Untuk waktu pengolahan makanan yang bonus bagaimana?


- Boleh langsung spontan berubah atau lewat queue
- Boleh tambahin properti di ADT makanan
- Boleh tambahin queue di property simulator (perlu konfirmasi lagi)

Ga bisa keluar dari bintang (divalidasi aja)

Properti lokasi aksi itu untuk memperoleh makanannya

Untuk UNDO/REDO : Dibebasin, yang penting dijelasin di laporan kenapa approach yang diambil gini.

Saran : yang disimpan ke stacknya itu commandnya aja, biar ngak berat di memory buat nyimpan state. Balik lagi ke kalian enaknya mau gimana.

	Tanda Tangan Asisten: 
--	---

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

No. Kelompok/Kelas : D / K01
Nama Kelompok : Kelompok D
Anggota Kelompok (Nama/NIM) :
1. Manuella Ivana Uli Sianipar / 13521051
2. Melvin Kent Jonathan / 13521052
3. Yobel Dean Christopher / 13521067
4. Fajar Maulana Herawan / 13521080
5. Jimly Firdaus / 13521102

Asisten Pembimbing : Rezda Abdullah Fachrezzi - 13519194

Asistensi I

Tanggal : 3 November 2022	Catatan Asistensi:
Tempat : ZOOM Meeting	

Kehadiran Anggota Kelompok:

No
NIM
Tanda tangan

1
13521051



2
13521052



3
13521067



4
13521080




5
13521102



Progress sesuai dengan lembar Milestone, tetapi di bagian main harus diperbaiki karena ada error/conflict. Tambahan progress di luar lembar Milestone ada di ADT Tree dan juga resep yang sudah dibuat

Tanda Tangan Asisten:

	
--	---

Lampiran :

Tanggal		03-11-2022
No	Fitur	Progress (0-100%)
1.	Command Parser	100
2.	Inisiasi	80
	a. Splash Screen	0
	b. Command START	100
	c. Command EXIT	100
3.	Simulator	
	d. ADT Simulator	75
4.	Makanan	
	e. Membaca makanan dari file	100
	f. ADT Makanan	75
	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	30
8.	Laporan (50%)	0