

LAPORAN TUGAS BESAR
IF2110/Algoritma dan Struktur Data

Engi's Kitchen Expansion

Dipersiapkan oleh:

RenggogeniXMakaji

13517013 Aditya Putra Santoso

13517040 Ariel Ansa Razumardi


13517073 Rayza Mahendra Guntara Harsono

13517145 Muhammad Al Terra

13517151 Rakhmad Budiono

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-8-K01</i>		<i>51</i>
		<i>Revisi</i>	<i>1</i>	<i>25-11-2018</i>

Daftar Isi

1 Ringkasan.....	4
2 Penjelasan Tambahan Spesifikasi Tugas	4
2.1 Save dan Load	4
2.2 Star Customer	5
2.3 Kedatangan Pelanggan Random.....	5
3 Struktur Data (ADT).....	5
3.1 Array.....	5
3.2 Binary Tree.....	7
3.3 File Manager	10
3.4 Graph.....	11
3.5 Jam	12
3.6 Layar.....	13
3.7 List Rekursif.....	14
3.8 Matriks	15
3.9 Mesin Baris	17
3.10 Mesin Command.....	18
3.11 Mesin Karakter	19
3.12 Mesin Karakter Input.....	19
3.13 Mesin Karakter Komparasi.....	20
3.14 Mesin Kata Komparasi	21
3.15 Movement.....	22
3.16 Pelanggan.....	22
3.17 Point.....	22
3.18 Queue.....	23
3.19 Stack	24
4 Program Utama	25
5 Algoritma-Algoritma Menarik.....	25
5.1 Recursive Descent	25
6 Data Test	25
6.1 Data Test 1	26
6.2 Data Test 2	27
6.3 Data Test 3	29
6.4 Data Test 4	31
6.5 Data Test 5	35
6.6 Data Test 6	36
6.7 Data Test 7	38
6.8 Data Test 8	40
6.9 Data Test 9	42
6.10 Data Test 10.....	42
6.11 Data Test 11	44
6.12 Data Test 12.....	46
7 Test Script	47

8	Pembagian Kerja dalam Kelompok	48
9	Lampiran	48
9.1	Deskripsi Tugas Besar 2.....	48
9.2	Notulen Rapat.....	49
9.3	Log Activity Anggota Kelompok.....	49
9.4	Form Asistensi.....	51

1 Ringkasan

Engi's Kitchen Expansion merupakan tugas besar mata kuliah Algoritma dan Struktur Data yang mana juga seri lanjutan dari tugas besar Dasar Pemrograman, Engi's Kitchen. Pada tugas besar kali ini, programmer dituntut untuk menggunakan beberapa Abstract Data Type (ADT), yang diantaranya adalah ADT Jam, ADT Point, ADT Array, ADT Matrix, ADT Mesin Karakter + Mesin Kata, ADT Queue, ADT Stack, ADT List, ADT Tree, ADT Graph yang merupakan variasi multilist, dan ADT pendukung lainnya. Secara khususnya Abstract Data Type (ADT) tersebut digunakan untuk sebagai berikut. ADT Jam digunakan untuk menyimpan waktu ketika melakukan save. ADT Point digunakan sebagai representasi posisi. ADT Array digunakan sebagai representasi pesanan (order). ADT Matrix digunakan sebagai representasi sebagian ruangan yang ada pada permainan. ADT Mesin Karakter + Mesin Kata digunakan untuk membaca informasi yang diperlukan dari file eksternal, misal informasi peta, membaca command dari pengguna, dan membaca state dari permainan yang sudah pernah disimpan. ADT Queue digunakan sebagai representasi antrian pelanggan. ADT Stack digunakan sebagai representasi tumpukan makanan di tangan dan di nampan. ADT List digunakan untuk implementasi Graf. ADT Tree digunakan sebagai representasi resep. ADT Graph (variasi multilist), digunakan untuk menghubungkan masing-masing bagian dari peta agar terbentuk suatu peta besar. Tugas besar Engi's Kitchen Expansion sendiri memiliki spesifikasi yang secara umum merupakan suatu manajemen restoran, yang dimana di dalam restoran tersebut memiliki beberapa ruangan. Ruangan yang tersedia dalam restoran tersebut berupa ruang untuk pelanggan beserta meja dan kursinya, beserta ruang dapur yang merupakan tempat untuk membuat makanan. Pelanggan yang datang akan dimasukkan kedalam antrian, pemain kemudian mengarahkan pelanggan dengan prioritas paling tinggi ke meja yang masih cukup kapasitasnya. Pelanggan yang menunggu terlalu lama akan pulang dan kredibilitas restoran akan menurun. Pembuatan makanan dilakukan dengan menambahkan bahan makanan kepada piring. Pencampuran bahan makanan mengikuti urutan tertentu berdasarkan pohon makanan. Makanan yang telah terbentuk diletakkan di nampan terlebih dahulu sebelum dibawa ke pelanggan.

Laporan ini menjelaskan deskripsi persoalan secara singkat, pemaparan asumsi yang diambil dalam pembuatan Tugas Besar (beserta spesifikasi fitur-fitur tambahan yang digunakan), pemaparan mengenai struktur data yang digunakan, penjelasan mengenai langkah kerja program utama, rincian data-test yang digunakan selama sesi testing, serta tabel pembagian tugas dalam kelompok.

Program yang telah dibuat sudah memenuhi semua spesifikasi. Program kami juga bisa menyimpan permainan jika permainan yang sedang dijalankan belum selesai. Setelah itu, file yang disimpan bisa digunakan untuk dimuat pada saat dibutuhkan.

2 Penjelasan Tambahan Spesifikasi Tugas

2.1 Save dan Load

Menggunakan ADT filemanager. Implementasinya dipakai di save dan load.

2.2 Star Customer

Star customer, menggunakan implementasi ADT pelanggan, tidak memiliki fungsi khusus, namun, memanfaatkan ADT pelanggan yang ada, fitur ini sudah dapat digunakan.

2.3 Kedatangan Pelanggan Random

Interval dari kedatangan pelanggan sudah menggunakan fungsi random.

3 Struktur Data (ADT)

3.1 Array

Pada ADT Array (header : array.h, body : array.c),
Primitif yang digunakan pada ADT ini adalah sebagai berikut:

```
/* ***** KONSTRUKTOR ***** */
/* Konstruktor : create tabel kosong */
void MakeEmptyArray (TabKata * T);
/* I.S. T sembarang */
/* F.S. Terbentuk tabel T kosong dengan kapasitas IdxMax-IdxMin+1 */

/* ***** SELEKTOR (TAMBAHAN) ***** */
/* *** Banyaknya elemen *** */
int NbElmt (TabKata T);
/* Mengirimkan banyaknya elemen efektif tabel */
/* Mengirimkan nol jika tabel kosong */
/* *** Daya tampung container *** */
int MaxNbEl (TabKata T);
/* Mengirimkan maksimum elemen yang dapat ditampung oleh tabel */
/* *** Selektor INDEKS *** */
IdxType GetFirstIdx (TabKata T);
/* Prekondisi : Tabel T tidak kosong */
/* Mengirimkan indeks elemen T pertama */
IdxType GetLastIdx (TabKata T);
/* Prekondisi : Tabel T tidak kosong */
/* Mengirimkan indeks elemen T terakhir */

/* ***** Test Indeks yang valid ***** */
boolean IsIdxValid (TabKata T, IdxType i);
/* Mengirimkan true jika i adalah indeks yang valid utk ukuran tabel */
/* yaitu antara indeks yang terdefinisi utk container*/
boolean IsIdxEff (TabKata T, IdxType i);
/* Mengirimkan true jika i adalah indeks yang terdefinisi utk tabel */
/* yaitu antara FirstIdx(T)..LastIdx(T) */

/* ***** TEST KOSONG/PENUH ***** */
/* *** Test tabel kosong *** */
boolean IsEmptyArray (TabKata T);
/* Mengirimkan true jika tabel T kosong, mengirimkan false jika tidak */
/* *** Test tabel penuh *** */

boolean IsFullArray (TabKata T);
/* Mengirimkan true jika tabel T penuh, mengirimkan false jika tidak */
```

```

/* ***** TULIS dengan INPUT/OUTPUT device ***** */
void TulisIsi (TabKata T);
/* Proses : Menuliskan isi tabel dengan traversal */
/* I.S. T boleh kosong */
/* F.S. Jika T tidak kosong : indeks dan elemen tabel ditulis berderet ke
bawah */
/*      Jika T kosong : Hanya menulis "Tabel kosong" */
/* Contoh: Jika isi Tabel: [1, 20, 30, 50]
Maka tercetak di layar:
[1]1
[2]20
[3]30
[4]50
*/
void TulisIsiTab (TabKata T);
/* Proses : Menuliskan isi tabel dengan traversal, tabel ditulis di antara
kurung siku;
antara dua elemen dipisahkan dengan separator "koma", tanpa tambahan
karakter di depan,
di tengah, atau di belakang, termasuk spasi dan enter */
/* I.S. T boleh kosong */
/* F.S. Jika T tidak kosong: [e1,e2,...,en] */
/* Contoh : jika ada tiga elemen bernilai 1, 20, 30 akan dicetak: [1,20,30]
*/
/* Jika tabel kosong : menulis [] */

/* ***** SEARCHING ***** */
/* *** Perhatian : Tabel boleh kosong!! *** */
IdxType Search1 (TabKata T, ElType X);
/* Search apakah ada elemen tabel T yang bernilai X */
/* Jika ada, menghasilkan indeks i terkecil, dengan elemen ke-i = X */
/* Jika tidak ada, mengirimkan IdxUndef */
/* Menghasilkan indeks tak terdefinisi (IdxUndef) jika tabel T kosong */
/* Memakai skema search TANPA boolean */

/* ***** OPERASI LAIN ***** */
void CopyTab (TabKata Tin, TabKata * Tout);
/* I.S. Tin terdefinisi, Tout sembarang */
/* F.S. Tout berisi salinan dari Tin (elemen dan ukuran identik) */
/* Proses : Menyalin isi Tin ke Tout */
TabKata InverseTab (TabKata T);
/* Menghasilkan tabel dengan urutan tempat yang terbalik, yaitu : */
/* elemen pertama menjadi terakhir, */
/* elemen kedua menjadi elemen sebelum terakhir, dst.. */
/* Tabel kosong menghasilkan tabel kosong */

/* ***** MENAMBAH ELEMEN ***** */
/* *** Menambahkan elemen terakhir *** */
void AddAsLastEl (TabKata * T, ElType X);
/* Proses: Menambahkan X sebagai elemen terakhir tabel */
/* I.S. Tabel T boleh kosong, tetapi tidak penuh */
/* F.S. X adalah elemen terakhir T yang baru */

```

```

void AddEli (TabKata * T, ElType X, IdxType i);
/* Menambahkan X sebagai elemen ke-i tabel tanpa mengganggu kontiguitas
   terhadap elemen yang sudah ada */
/* I.S. Tabel tidak kosong dan tidak penuh */
/*      i adalah indeks yang valid. */
/* F.S. X adalah elemen ke-i T yang baru */
/* Proses : Geser elemen ke-i+1 s.d. terakhir */
/*      Isi elemen ke-i dengan X */

/* ***** MENGHAPUS ELEMEN ***** */
void DelLastEl (TabKata * T, ElType * X);
/* Proses : Menghapus elemen terakhir tabel */
/* I.S. Tabel tidak kosong */
/* F.S. X adalah nilai elemen terakhir T sebelum penghapusan, */
/*      Banyaknya elemen tabel berkurang satu */
/*      Tabel T mungkin menjadi kosong */
void DelEli (TabKata * T, IdxType i, ElType * X);
/* Menghapus elemen ke-i tabel tanpa mengganggu kontiguitas */
/* I.S. Tabel tidak kosong, i adalah indeks efektif yang valid */
/* F.S. X adalah nilai elemen ke-i T sebelum penghapusan */
/*      Banyaknya elemen tabel berkurang satu */
/*      Tabel T mungkin menjadi kosong */
/* Proses : Geser elemen ke-i+1 s.d. elemen terakhir */
/*      Kurangi elemen efektif tabel */

```

ADT array yang kami gunakan untuk memenuhi Tugas Besar 2 ini merupakan adaptasi ulang dari ADT array yang telah diberikan pada masa lampau untuk menangani sebuah array integer. Pada Tugas Besar 2 ini struktur data yang menggunakan TabInt sebagai array dari integer diubah menjadi TabKata yang menerima array dari kata. Pemilihan struktur data ini diperlukan sebagai salah satu cara untuk mengolah proses pemesanan.

3.2 Binary Tree

Pada ADT Binary Tree (header : bintree.h, body : bintree.c), Primitif yang digunakan pada ADT ini adalah sebagai berikut

```

/* *** Konstruktor *** */
BinTree Tree(infotypeLR Akar, BinTree L, BinTree R);
/* Menghasilkan sebuah pohon biner dari A, L, dan R, jika alokasi berhasil */
/* Menghasilkan pohon kosong (Nil) jika alokasi gagal */
void MakeTree(infotypeLR Akar, BinTree L, BinTree R, BinTree *P);
/* I.S. Akar, L, R terdefinisi. P Sembarang */
/* F.S. Membentuk pohon P dengan Akar(P)=Akar, Left(P)=L, dan Right(P)=R
   jika alokasi berhasil. P = Nil jika alokasi gagal. */
BinTree PreOrderToTree(char s[500],int *pos);
/* Manajemen Memory */
addrNode AlokNode(infotypeLR X);
/* Mengirimkan addrNode hasil alokasi sebuah elemen */
/* Jika alokasi berhasil, maka addrNode tidak Nil, dan misalnya menghasilkan
P,
   maka Akar(P) = X, Left(P) = Nil, Right(P)=Nil */
/* Jika alokasi gagal, mengirimkan Nil */
void DealokNode(addrNode P);
/* I.S. P terdefinisi */

```

```

/* F.S. P dikembalikan ke sistem */
/* Melakukan dealokasi/pengembalian addrNode P */

/* *** Predikat-Predikat Penting *** */
boolean IsTreeEmpty(BinTree P);
/* Mengirimkan true jika P adalah pohon biner kosong */
boolean IsTreeOneElmt(BinTree P);
/* Mengirimkan true jika P adalah pohon biner tidak kosong dan hanya memiliki
1 elemen */
boolean IsUnerLeft(BinTree P);
/* Mengirimkan true jika pohon biner tidak kosong P adalah pohon unerleft:
hanya mempunyai subpohon kiri */
boolean IsUnerRight(BinTree P);
/* Mengirimkan true jika pohon biner tidak kosong P adalah pohon unerright:
hanya mempunyai subpohon kanan*/
boolean IsBiner(BinTree P);
/* Mengirimkan true jika pohon biner tidak kosong P adalah pohon biner:
mempunyai subpohon kiri dan subpohon kanan*/

/* *** Traversal *** */
void PrintPreorder(BinTree P);
/* I.S. P terdefinisi */
/* F.S. Semua simpul P sudah dicetak secara preorder: akar, pohon kiri, dan
pohon kanan.
    Setiap pohon ditandai dengan tanda kurung buka dan kurung tutup ().
    Pohon kosong ditandai dengan ().
    Tidak ada tambahan karakter apa pun di depan, tengah, atau akhir. */
/* Contoh:
    (A()) adalah pohon dengan 1 elemen dengan akar A
    (A(B())(C())) adalah pohon dengan akar A dan subpohon kiri (B()) dan
subpohon kanan (C()) */
char* GetPreOrderForm(BinTree P);
void PrintInorder(BinTree P);
/* I.S. P terdefinisi */
/* F.S. Semua simpul P sudah dicetak secara inorder: pohon kiri, akar, dan
pohon kanan.
    Setiap pohon ditandai dengan tanda kurung buka dan kurung tutup ().
    Pohon kosong ditandai dengan ().
    Tidak ada tambahan karakter apa pun di depan, tengah, atau akhir. */
/* Contoh:
    (())A() adalah pohon dengan 1 elemen dengan akar A
    (((B())A())C()) adalah pohon dengan akar A dan subpohon kiri ((B())) dan
subpohon kanan ((C())) */
void PrintPostorder(BinTree P);
/* I.S. P terdefinisi */
/* F.S. Semua simpul P sudah dicetak secara postorder: pohon kiri, pohon
kanan, dan akar.
    Setiap pohon ditandai dengan tanda kurung buka dan kurung tutup ().
    Pohon kosong ditandai dengan ().
    Tidak ada tambahan karakter apa pun di depan, tengah, atau akhir. */
/* Contoh:
    (())A adalah pohon dengan 1 elemen dengan akar A
    (((())B)((())C)A) adalah pohon dengan akar A dan subpohon kiri ((())B) dan
subpohon kanan ((())C) */
void PrintTree(BinTree P, int h);

```



```

/* I.S. P terdefinisi, h adalah jarak indentasi (spasi) */
/* F.S. Semua simpul P sudah ditulis dengan indentasi (spasi) */
/* Penulisan akar selalu pada baris baru (diakhiri newline) */
/* Contoh, jika h = 2:
1) Pohon preorder: (A()()) akan ditulis sbb:
A
2) Pohon preorder: (A(B()())(C()())) akan ditulis sbb:
A
  B
  C
3) Pohon preorder: (A(B(D()())())(C()(E()()))) akan ditulis sbb:
A
  B
    D
  C
    E
*/

/* *** Searching *** */
boolean SearchTree(BinTree P, infotypeLR X);
/* Mengirimkan true jika ada node dari P yang bernilai X */

/* *** Fungsi-Fungsi Lain *** */
int NbElmtTree(BinTree P);
/* Mengirimkan banyaknya elemen (node) pohon biner P */
int NbDaun(BinTree P);
/* Mengirimkan banyaknya daun (node) pohon biner P */
/* Prekondisi: P tidak kosong */
boolean IsSkewLeft(BinTree P);
/* Mengirimkan true jika P adalah pohon condong kiri */
/* Pohon kosong adalah pohon condong kiri */
boolean IsSkewRight(BinTree P);
/* Mengirimkan true jika P adalah pohon condong kanan */
/* Pohon kosong adalah pohon condong kanan */
int Level(BinTree P, infotypeLR X);
/* Mengirimkan level dari node X yang merupakan salah satu simpul dari pohon biner P.
   Akar(P) level-nya adalah 1. Pohon P tidak kosong. */
int Tinggi(BinTree P);
/* Pohon Biner mungkin kosong. Tinggi pohon kosong = 0.
   Mengirim "height" yaitu tinggi dari pohon */

/* *** Operasi lain *** */
void AddDaunTerkiri(BinTree *P, infotypeLR X);
/* I.S. P boleh kosong */
/* F.S. P bertambah simpulnya, dengan X sebagai simpul daun terkiri */
void AddDaun(BinTree *P, infotypeLR X, infotypeLR Y, boolean Kiri);
/* I.S. P tidak kosong, X adalah salah satu daun Pohon Biner P */
/* F.S. P bertambah simpulnya, dengan Y sebagai anak kiri X (jika Kiri =
true), atau
   sebagai anak Kanan X (jika Kiri = false) */
/*      Jika ada > 1 daun bernilai X, diambil daun yang paling kiri */
void DelDaunTerkiri(BinTree *P, infotypeLR *X);
/* I.S. P tidak kosong */

```

```

/* F.S. P dihapus daun terkirinya, dan didealokasi, dengan X adalah info yang semula
    disimpan pada daun terkiri yang dihapus */
void DelDaun(BinTree *P, infotypeLR X);
/* I.S. P tidak kosong, minimum ada 1 daun bernilai X. */
/* F.S. Semua daun bernilai X dihapus dari P. */
ListRek MakeListDaun(BinTree P);
/* Jika P adalah pohon kosong, maka menghasilkan list kosong. */
/* Jika P bukan pohon kosong: menghasilkan list yang elemennya adalah semua daun pohon P,
    jika semua alokasi list berhasil.
    Daun terkiri menjadi elemen pertama dari list, diikuti elemen kanannya, dst.
    Menghasilkan list kosong jika ada alokasi yang gagal. */
ListRek MakeListPreorder(BinTree P);
/* Jika P adalah pohon kosong, maka menghasilkan list kosong. */
/* Jika P bukan pohon kosong: menghasilkan list yang elemennya adalah semua elemen pohon P
    dengan urutan preorder, jika semua alokasi berhasil.
    Menghasilkan list kosong jika ada alokasi yang gagal. */
ListRek MakeListLevel(BinTree P, int N);
/* Jika P adalah pohon kosong, maka menghasilkan list kosong. */
/* Jika P bukan pohon kosong: menghasilkan list yang elemennya adalah semua elemen pohon P
    yang levelnya=N, jika semua alokasi berhasil.
    Elemen terkiri menjadi elemen pertama dari list, diikuti elemen kanannya, dst.
    Menghasilkan list kosong jika ada alokasi yang gagal. */
infotypeLR GetDaunAcak(BinTree P);
/* *** Binary Search Tree ***
boolean BSearch(BinTree P, infotypeLR X);
/* Mengirimkan true jika ada node dari P yang bernilai X

void InsSearch(BinTree *P, infotypeLR X);
/* Menghasilkan sebuah pohon Binary Search Tree P
/* dengan tambahan simpul X. Belum ada simpul P yang bernilai X.

void DelBtree(BinTree *P, infotypeLR X);
/* I.S. Pohon P tidak kosong
/* F.S. Nilai X yang dihapus pasti ada
/* Sebuah node dengan nilai X dihapus */

```

ADT bintree yang kami gunakan untuk memenuhi Tugas Besar 2 ini merupakan adaptasi ulang dari ADT bintree yang telah diberikan pada masa lampau untuk menangani tree dengan akar-akar integer. Selain menggunakan primitif-primitif pada awal ADT yang didapat, bintree yang digunakan juga mengandung primitif tambahan seperti PreOrderToTree, GetPreOrderForm dan GetDaunAcak. Primitif-primitif tersebut kami buat untuk menambah kemampuan ADT agar dapat diimplementasikan saat menangani mekanisme pengambilan order dan bagaimana untuk mengeluarkan pesan yang bisa memiliki banyak kemungkinan.

3.3 File Manager

MATRIKS BacaMap(char namaFile[100]);

STEI- ITB	I	Halaman 10 dari 53 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.		

```

BinTree BacaPohonMakanan(char namaFile[100]);
TabKata BacaArray(char namaFile[100]);
void TulisSaveGame(char namaFile[100], Queue ANS, Queue AS, long Uang, long
Nyawa, JAM Waktu, POINT Posisi,
                int Stage, TabKata Orderan, TabKata BOO, Queue PDM,
MATRIKS P1, MATRIKS P2, MATRIKS P3, MATRIKS P4,
                Graph Pintu, BinTree PohM, Stack Food, Stack Hand);
void BacaSaveGame(char namaFile[100], Queue *ANS, Queue *AS, long int *Uang,
long int *Nyawa, JAM *Waktu,
                POINT *Posisi, int *Stage, TabKata *Orderan, TabKata
*BOO, Queue *PDM, MATRIKS *P1, MATRIKS *P2,
                MATRIKS *P3, MATRIKS *P4, Graph *Pintu, BinTree *PohM,
Stack *Food, Stack *Hand);
Queue BacaQueueMejaPelanggan(char namaFile[100]);
Queue BacaQueuePelanggan(char namaFile[100]);
Stack BacaStackFood(char namaFile[100]);
Stack BacaStackHand(char namaFile[100]);
Graph BacaGraphPintu(char namaFile[100]);

```

ADT file manager yang kami gunakan untuk memenuhi Tugas Besar 2 ini merupakan implementasi unik yang dimanfaatkan untuk melakukan manajemen pembacaan file external yang digunakan untuk melakukan start game awal. Pemilihan ini dilakukan karena beragamnya data yang digunakan dan akan lebih mudah apabila semua fungsi dapat digabungkan.

3.4 Graph

```

//Konstruktor
void CreateGraph(int X, Graph* G);

//Manajemen Memory
adrNode AlokNodeGraph(int X);
void DealokNodeGraph(adrNode P);
adrSuccNode AlokSuccNode(adrNode Pn, POINT asal, POINT tujuan);
void DealokSuccNode(adrSuccNode P);

adrNode SearchNode(Graph G, int X);
adrSuccNode SearchEdge(Graph G, int prec, int succ);
void InsertNode(Graph* G, int X, adrNode* Pn);
void InsertEdge(Graph* G, int prec, int succ, POINT asal, POINT tujuan);

/* Tubes */
void CreatePintu(Graph* G, int room1, int room2, POINT pintuRoom1, POINT
pintuRoom2);
void CariEdgePintu(Graph G, POINT asal, int roomAsal, int* roomTujuan, POINT*
tujuan);

```

ADT graph yang kami gunakan untuk memenuhi Tugas Besar 2 ini merupakan implementasi unik berdasarkan multilist dari listrek yang telah diberikan pada masa lampau untuk menangani suatu graf integer yang setiap nodenya dapat menunjuk ke tidak hanya 1 arah tetapi berbagai arah. Pada Tugas Besar 2 ini, struktur data yang menggunakan bentuk graf ini kami gunakan sebagai manajemen ruangan dan bagaimana mekanisme pintu ruangan dari map dapat dilakukan.

Pemilihan struktur data ini diperlukan sebagai salah satu cara untuk mempermudah dalam melakukan pemetaan pintu dan pengolahan ruangan.

3.5 Jam

```
/* ***** */
/* DEFINISI PRIMITIF */
/* ***** */
/* KELOMPOK VALIDASI TERHADAP TYPE */
/* ***** */
boolean IsJAMValid (int H, int M, int S);
/* Mengirim true jika H,M,S dapat membentuk J yang valid */
/* dipakai untuk mentest SEBELUM membentuk sebuah Jam */

/* *** Konstruktor: Membentuk sebuah JAM dari komponen-komponennya *** */
JAM MakeJAM (int HH, int MM, int SS);
/* Membentuk sebuah JAM dari komponen-komponennya yang valid */
/* Prekondisi : HH, MM, SS valid untuk membentuk JAM */

/* ***** */
/* KELOMPOK BACA/TULIS */
/* ***** */
void BacaJAM (JAM * J);
/* I.S. : J tidak terdefinisi */
/* F.S. : J terdefinisi dan merupakan jam yang valid */
/* Proses : mengulangi membaca komponen HH, MM, SS sehingga membentuk J */
/* yang valid. Tidak mungkin menghasilkan J yang tidak valid. */
/* Pembacaan dilakukan dengan mengetikkan komponen HH, MM, SS
dalam satu baris, masing-masing dipisahkan 1 spasi, diakhiri enter. */
/* Jika JAM tidak valid maka diberikan pesan: "Jam tidak valid", dan
pembacaan
diulangi hingga didapatkan jam yang valid. */
/* Contoh:
60 3 4
Jam tidak valid
1 3 4
--> akan terbentuk JAM <1,3,4> */

void TulisJAM (JAM J);
/* I.S. : J sembarang */
/* F.S. : Nilai J ditulis dg format HH:MM:SS */
/* Proses : menulis nilai setiap komponen J ke layar dalam format HH:MM:SS
tanpa karakter apa pun di depan atau belakangnya, termasuk spasi, enter,
dll.*/

/* ***** */
/* KELOMPOK KONVERSI TERHADAP TYPE */
/* ***** */
long JAMToDetik (JAM J);
/* Diberikan sebuah JAM, mengkonversi menjadi jumlah detik dari pukul 0:0:0
*/
/* Rumus : detik = 3600*HH + 60*MM + SS */
/* Nilai maksimum = 3600*23+59*60+59 */

JAM DetikToJAM (long N);
```

```

/* Mengirim konversi detik ke JAM */
/* Catatan: Jika N >= 86400, maka harus dikonversi dulu menjadi jumlah detik
yang
mewakili jumlah detik yang mungkin dalam 1 hari, yaitu dengan rumus:
N1 = N mod 86400, baru N1 dikonversi menjadi JAM */

/* ***** */
/* KELOMPOK OPERASI TERHADAP TYPE */
/* ***** */
/* *** Kelompok Operator Relational *** */
boolean JEQ (JAM J1, JAM J2);
/* Mengirimkan true jika J1=J2, false jika tidak */
boolean JNEQ (JAM J1, JAM J2);
/* Mengirimkan true jika J1 tidak sama dengan J2 */
boolean JLT (JAM J1, JAM J2);
/* Mengirimkan true jika J1<J2, false jika tidak */
boolean JGT (JAM J1, JAM J2);
/* Mengirimkan true jika J1>J2, false jika tidak */
/* *** Operator aritmatika JAM *** */
JAM NextDetik (JAM J);
/* Mengirim 1 detik setelah J dalam bentuk JAM */
JAM NextNDetik (JAM J, int N);
/* Mengirim N detik setelah J dalam bentuk JAM */
JAM PrevDetik (JAM J);
/* Mengirim 1 detik sebelum J dalam bentuk JAM */
JAM PrevNDetik (JAM J, int N);
/* Mengirim N detik sebelum J dalam bentuk JAM */
/* *** Kelompok Operator Aritmetika *** */
long Durasi (JAM JAw, JAM JAkh);
/* Mengirim JAkh-JAw dlm Detik, dengan kalkulasi */
/* Jika JAw > JAkh, maka JAkh adalah 1 hari setelah JAw */

```

ADT jam yang kami gunakan untuk memenuhi Tugas Besar 2 ini merupakan adaptasi ulang dari ADT jam yang telah diberikan pada masa lampau untuk menangani sebuah data tipe jam. ADT mengolah 3 buah integer menjadi suatu bentuk JAM yang telah divalidasi agar tidak ada bentuk jam tak sesuai format dan dapat dilakukan operasi padanya. Pada Tugas Besar 2 ini struktur data yang menggunakan bentuk jam ini kami gunakan sebagai penghitung tick saat setiap aksi dilakukan. Pemilihan struktur data ini diperlukan sebagai salah satu cara untuk mengolah mekanika utama permainan yaitu saat seseorang akan pergi dan apabila kesabarannya sudah habis.

3.6 Layar

```

void CreateEmptyLayar(LAYAR *L, int width, int height);
//Membuat layar kosong dengan ukuran width * height

void CreateGarisHorizontal(LAYAR *L, int startX, int finishX, int y);
//Membuat garis dari koordinat (startX,y) hingga (finishX, y)

void CreateGarisVertikal(LAYAR *L, int x, int startY, int finishY);
//Membuat garis dari koordinat (x, startY) hingga (x, finishY)

void CreateTabel(LAYAR *L, int startX, int startY, int finishX, int finishY,
int rows, int cols);

```

```

//Membuat tabel dari koordinat (startX,startY) hingga (finishX,finishY)
dengan rinician
//Baris sebanyak rows, kolom sebanyak cols
//Lebar baris dan kolom disesuaikan dari deltaX/cols dan deltaY/rows
/* Rincian tambahan :
* Garis atas tabel ada di (startX,startY) hingga (finishX,startY)
* Garis bawah ada di (startX,finishY) hingga (finishX,finishY)
* Garis kiri ada di (startX,startY) hingga (startX,finishY)
* Garis kanan ada di (finishX,startY) hingga (finishX,finishY)
*/

void PutTextAt(LAYAR *L, int x, int y, char str[20]);
//Menaruh String str di posisi (x,y) di layar

//void initLayarGame(LAYAR *L, char nama[20], int money, int life, int time,
int waitingCust, Queue order, Stack food, Stack hand, MATRIKS ruangan);
//Mengisi layar dengan semua elemen UI dari game seperti nama, uang, nyawa,
waktu, banyak kustomer, orderan
//Food stack, makanan di tangan

void printLayar(LAYAR L);
//Print layar ke console

```

ADT layar yang kami gunakan untuk memenuhi Tugas Besar 2 ini merupakan adaptasi unik yang kami lakukan agar pada saat program dijalankan, tampilan pada program dapat dikelola dan berubah dari waktu ke waktu. Pada Tugas Besar 2 ini struktur data yang menggunakan bentuk jam ini kami gunakan sebagai penggambar mengenai layout utama dari permainan. Pemilihan struktur data ini diperlukan sebagai salah satu cara untuk menggambarkan status pengguna dan lokasinya di map. Primitif-primitif pada layar banyak berkaitan dengan CreateEmpty, PutTextAt, initLayarGame dan sebagainya sebagai konstruktor layar.

3.7 List Rekursif

```

/* *** Manajemen Memori *** */
addressListRek AlokasiListRek (infotypeLR X);
/* Mengirimkan addressListRek hasil alokasi sebuah elemen */
/* Jika alokasi berhasil, maka addressListRek tidak Nil, dan misalnya
menghasilkan P,
    maka Info(P) = X, Next(P) = Nil */
/* Jika alokasi gagal, mengirimkan Nil */
void DealokasiListRek (addressListRek P);
/* I.S. P terdefinisi */
/* F.S. P dikembalikan ke sistem */
/* Melakukan dealokasi/pengembalian addressListRek P */

/* *** Primitif-primitif yang harus direalisasikan *** */

/* Pemeriksaan Kondisi ListRek */
int IsEmptyListRek(ListRek L);
/* Mengirimkan 1 jika L kosong dan 0 jika L tidak kosong */
int IsOneElmtListRek(ListRek L);
/* Mengirimkan 1 jika L berisi 1 elemen dan 0 jika > 1 elemen atau kosong */

/* *** Selektor *** */

```

```

infotypeLR FirstElmtListRek (ListRek L);
/* Mengirimkan elemen pertama sebuah listRek L yang tidak kosong */
ListRek TailListRek(ListRek L);
/* Mengirimkan listRek L tanpa elemen pertamanya, mungkin menjadi listRek
kosong */

/* *** Konstruktor *** */
ListRek KonsoListRek(infotypeLR e, ListRek L);
/* Mengirimkan listRek L dengan tambahan e sebagai elemen pertamanya.
e dialokasi terlebih dahulu. Jika alokasi gagal, mengirimkan L. */
ListRek KonsBListRek(ListRek L, infotypeLR e);
/* Mengirimkan listRek L dengan tambahan e sebagai elemen pertamanya */
/* e harus dialokasi terlebih dahulu */
/* Jika alokasi e gagal, mengirimkan L */

/* *** Operasi Lain *** */
ListRek CopyListRek (ListRek L);
/* Mengirimkan salinan listRek L (menjadi listRek baru) */
/* Jika ada alokasi gagal, mengirimkan L */
void MCopyListRek (ListRek Lin, ListRek *Lout);
/* I.S. Lin terdefinisi */
/* F.S. Lout berisi salinan dari Lin */
/* Proses : menyalin Lin ke Lout */
ListRek ConcatListRek (ListRek L1, ListRek L2);
/* Mengirimkan salinan hasil konkatenasi listRek L1 dan L2 (menjadi listRek
baru) */
/* Jika ada alokasi gagal, menghasilkan Nil */
void MConcatListRek (ListRek L1, ListRek L2, ListRek *LHsl);
/* I.S. L1, L2 terdefinisi */
/* F.S. LHsl adalah hasil melakukan konkatenasi L1 dan L2 dengan cara
"disalin" */
/* Proses : Menghasilkan salinan hasil konkatenasi listRek L1 dan L2 */
void PrintListRek (ListRek L);
/* I.S. L terdefinisi. */
/* F.S. Setiap elemen listRek dicetak. */
int NbElmtListRek (ListRek L);
/* Mengirimkan banyaknya elemen listRek L, Nol jika L kosong */
boolean SearchListRek (ListRek L, infotypeLR X);
/* Mengirim true jika X adalah anggota listRek, false jika tidak */

```

ADT listrek yang kami gunakan untuk memenuhi Tugas Besar 2 ini merupakan adaptasi ulang dari ADT listrek yang telah diberikan pada masa lampau untuk menangani sebuah list yang hanya memiliki address yang nantinya saat dioperasikan beraksi sebagai suatu list yang terus memanggil address selanjutnya secara rekursif. Pada Tugas Besar 2 ini struktur data yang menggunakan bentuk listrek ini kami gunakan sebagai salah satu alat yang digunakan untuk membuat multilist graf dan bintree. Pemilihan struktur data ini diperlukan sebagai salah satu cara untuk mengolah ADT, bagaimana cara pengolahan tree dan graf harus memanfaatkan ADT ini karena akan terus memanggil address lain.

3.8 Matriks

```

/* ***** DEFINISI PROTOTIPE PRIMITIF ***** */
/* *** Konstruktor membentuk MATRIKS *** */
void MakeMATRIKS (int NB, int NK, MATRIKS * M);

```

STEI- ITB	I	Halaman 15 dari 53 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.		

```

/* Membentuk sebuah MATRIKS "kosong" yang siap diisi berukuran NB x NK di
"ujung kiri" memori */
/* I.S. NB dan NK adalah valid untuk memori matriks yang dibuat */
/* F.S. Matriks M sesuai dengan definisi di atas terbentuk */

/* *** Selektor *** */
#define NBrsEff(M) (M).NBrsEff
#define NKolEff(M) (M).NKolEff
#define ElmtMat(M,i,j) (M).Mem[(i)][(j)]

/* *** Selektor "DUNIA MATRIKS" *** */
boolean IsIdxValidMat (int i, int j);
/* Mengirimkan true jika i, j adalah indeks yang valid untuk matriks apa pun
*/

/* *** Selektor: Untuk sebuah matriks M yang terdefinisi: *** */
indeks GetFirstIdxBrs (MATRIKS M);
/* Mengirimkan indeks baris terkecil M */
indeks GetFirstIdxKol (MATRIKS M);
/* Mengirimkan indeks kolom terkecil M */
indeks GetLastIdxBrs (MATRIKS M);
/* Mengirimkan indeks baris terbesar M */
indeks GetLastIdxKol (MATRIKS M);
/* Mengirimkan indeks kolom terbesar M */
boolean IsIdxEffMat (MATRIKS M, indeks i, indeks j);
/* Mengirimkan true jika i, j adalah indeks efektif bagi M */
/* ***** Assignment MATRIKS ***** */
void CopyMATRIKS (MATRIKS MIn, MATRIKS * MHsl);
/* Melakukan assignment MHsl MIn */

/* ***** KELOMPOK BACA/TULIS ***** */
void BacaMATRIKS (MATRIKS * M, int NB, int NK);
/* I.S. IsIdxValidMat(NB,NK) */
/* F.S. M terdefinisi nilai elemen efektifnya, berukuran NB x NK */
/* Proses: Melakukan MakeMATRIKS(M,NB,NK) dan mengisi nilai efektifnya */
/* Selanjutnya membaca nilai elemen per baris dan kolom */
/* Contoh: Jika NB = 3 dan NK = 3, maka contoh cara membaca isi matriks :
1 2 3
4 5 6
8 9 10
*/
void TulisMATRIKS (MATRIKS M);
/* I.S. M terdefinisi */
/* F.S. Nilai M(i,j) ditulis ke layar per baris per kolom, masing-masing
elemen per baris
dipisahkan sebuah spasi */
/* Proses: Menulis nilai setiap elemen M ke layar dengan traversal per baris
dan per kolom */
/* Contoh: menulis matriks 3x3 (ingat di akhir tiap baris, tidak ada spasi)
1 2 3
4 5 6
8 9 10
*/

/* ***** KELOMPOK OPERASI RELASIONAL TERHADAP MATRIKS ***** */

```



```

boolean EQ (MATRIKS M1, MATRIKS M2);
/* Mengirimkan true jika M1 = M2, yaitu NBelmtMat(M1) = NBelmtMat(M2) dan */
/* untuk setiap i,j yang merupakan indeks baris dan kolom M1(i,j) = M2(i,j) */
/* Juga merupakan strong EQ karena GetFirstIdxBrs(M1) = GetFirstIdxBrs(M2)
   dan GetLastIdxKol(M1) = GetLastIdxKol(M2) */
boolean NEQ (MATRIKS M1, MATRIKS M2);
/* Mengirimkan true jika M1 tidak sama dengan M2 */
boolean EQSize (MATRIKS M1, MATRIKS M2);
/* Mengirimkan true jika ukuran efektif matriks M1 sama dengan ukuran efektif
M2 */
/* yaitu GetBrsEff(M1) = GetNBrsEff (M2) dan GetNKolEff (M1) = GetNKolEff
(M2) */

/* ***** Operasi lain ***** */
int NBelmtMATRIKS (MATRIKS M);
/* Mengirimkan banyaknya elemen M */

/* ***** KELOMPOK TEST TERHADAP MATRIKS ***** */
boolean IsBujurSangkar (MATRIKS M);
/* Mengirimkan true jika M adalah matriks dg ukuran baris dan kolom sama */
boolean IsSimetri (MATRIKS M);
/* Mengirimkan true jika M adalah matriks simetri : IsBujurSangkar(M)
   dan untuk setiap elemen M, M(i,j)=M(j,i) */
void Transpose (MATRIKS * M);
/* I.S. M terdefinisi dan IsBujursangkar(M) */
/* F.S. M "di-transpose", yaitu setiap elemen M(i,j) ditukar nilainya dengan
elemen M(j,i) */

```

ADT matriks yang kami gunakan untuk memenuhi Tugas Besar 2 ini merupakan adaptasi ulang dari ADT matriks yang telah diberikan pada masa lampau untuk menangani sebuah matriks integer. Pada Tugas Besar 2 ini struktur data yang menggunakan bentuk matriks ini kami gunakan sebagai pengolah berbagai macam integer dalam program, dalam implementasinya kami telah mengurangi beberapa primitif yang menurut kami kurang relevan untuk berjalannya tugas. Pemilihan struktur data ini sangat diperlukan karena matriks yang mengolah integer digunakan di berbagai tempat dan dapat digunakan oleh ADT buatan untuk diimplementasikan.

3.9 Mesin Baris

```

void IgnoreBlank();
/* Mengabaikan satu atau beberapa BLANK
   I.S. : CC sembarang
   F.S. : CC ≠ BLANK atau CC = MARK */

void STARTBARIS(char filename[100]);
/* I.S. : CC sembarang
   F.S. : EndBaris = true, dan CC = MARK;
          atau EndBaris = false, CBaris adalah baris yang sudah diakuisisi,
          CC karakter pertama sesudah karakter terakhir baris */

void ADVBARIS();
/* I.S. : CC adalah karakter pertama baris yang akan diakuisisi
   F.S. : CBaris adalah baris terakhir yang sudah diakuisisi,
          CC adalah karakter pertama dari baris berikutnya, mungkin MARK

```

STEI- ITB	I	Halaman 17 dari 53 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.		

```

        Jika CC = MARK, EndBaris = true.
Proses : Akuisisi baris menggunakan procedure SalinBaris */

```

```

void SalinBaris();
/* Mengakuisisi baris, menyimpan dalam baris
   I.S. : CC adalah karakter pertama dari baris
   F.S. : CBaris berisi baris yang sudah diakuisisi;
         CC = BLANK atau CC = MARK;
         CC adalah karakter sesudah karakter terakhir yang diakuisisi.

```

ADT mesinbaris yang kami gunakan untuk memenuhi Tugas Besar 2 ini merupakan adaptasi unik dari ADT mesinkata dan mesinkarakter yang telah diberikan pada masa lampau untuk menangani sebuah kata. Pada adaptasi kami dalam tugas ini mesinbaris adalah sebuah modifikasi dari mesinkata yang mengolah baris per baris, dengan primitif seperti STARTBARIS dan ADVBARIS yang dapat membaca string per baris jika belum mark. Pada Tugas Besar 2 ini struktur data yang menggunakan bentuk baris ini kami gunakan sebagai pembaca informasi-informasi dari file-file external agar kemudahan dalam membaca informasi yang lebih kompleks dari kalimat dapat dilakukan. Pemilihan struktur data ini diperlukan sebagai salah satu cara untuk mengolah berbagai string yang ada di file karena mayoritas data yang akan diolah adalah bentuk string dan mesinbaris bertindak sebagai salah satu pengganti primitif strings.

3.10Mesin Command

```

void IgnoreBlankInput();
/* Mengabaikan satu atau beberapa BLANK
   I.S. : CC sembarang
   F.S. : CC ≠ BLANK atau CC = MARK */

void IgnoreBlankNewlineInput();

void STARTCOMMAND();

void ADVCOMMAND();
/* I.S. : CC adalah karakter pertama kata yang akan diakuisisi
   F.S. : CKata adalah kata terakhir yang sudah diakuisisi,
         CC adalah karakter pertama dari kata berikutnya, mungkin MARK
         Jika CC = MARK, EndKata = true.
   Proses : Akuisisi kata menggunakan procedure SalinKata */

void SalinCommand();
/* Mengakuisisi kata, menyimpan dalam CKata
   I.S. : CC adalah karakter pertama dari kata
   F.S. : CKata berisi kata yang sudah diakuisisi;
         CC = BLANK atau CC = MARK;
         CC adalah karakter sesudah karakter terakhir yang diakuisisi.
         Jika panjang kata melebihi NMax, maka sisa kata "dipotong" */
boolean CommandIs(char s[]);

```

ADT mesin command yang kami gunakan untuk memenuhi Tugas Besar 2 ini merupakan adaptasi unik dari ADT mesinkata dan ADT mesinkar yang telah diberikan pada masa lampau untuk menangani sebuah kata dan karakter. Pada Tugas Besar 2 ini struktur data yang menggunakan bentuk kalimat ini kami gunakan sebagai bagaimana cara program untuk

STEI- ITB	I	Halaman 18 dari 53 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.		

meregister command dari user sebagai suatu command yang valid atau tidak, SalinCommand dan CommandIs berfungsi untuk memberikan informasi kepada program apa command yang telah diberikan oleh user. Pemilihan struktur data ini diperlukan sebagai salah satu cara untuk menjalankan program sebagai user dan bagaimana program ini akan berjalan secara utamanya.

3.11 Mesin Karakter

```
void START(char filename[100]);
/* Mesin siap dioperasikan. Pita disiapkan untuk dibaca.
   Karakter pertama yang ada pada pita posisinya adalah pada jendela.
   I.S. : sembarang
   F.S. : CC adalah karakter pertama pada pita
          Jika CC != MARK maka EOP akan padam (false)
          Jika CC = MARK maka EOP akan menyala (true) */

void ADV();
/* Pita dimajukan satu karakter.
   I.S. : Karakter pada jendela = CC, CC != MARK
   F.S. : CC adalah karakter berikutnya dari CC yang lama,
          CC mungkin = MARK
          Jika CC = MARK maka EOP akan menyala (true) */
```

ADT mesin karakter yang kami gunakan untuk memenuhi Tugas Besar 2 ini merupakan adaptasi ulang dari ADT mesinkar yang telah diberikan pada masa lampau untuk menangani sebuah pita karakter. Pada Tugas Besar 2 ini struktur data yang menggunakan bentuk pita karakter ini kami gunakan sebagai konstruktor yang dapat digunakan untuk ADT-ADT seperti MesinKarakterInput, mesincommand dan yang lainnya sebagai fondasi pembacaan suatu kata mulai dari karakter per karakter. Pemilihan struktur data ini diperlukan sebagai implementasi utama pengolahan suatu string yang dikirimkan oleh user atau akan diolah oleh program.

3.12 Mesin Karakter Input

```
void STARTINPUT();
/* Mesin siap dioperasikan. Pita disiapkan untuk dibaca.
   Karakter pertama yang ada pada pita posisinya adalah pada jendela.
   I.S. : sembarang
   F.S. : CC adalah karakter pertama pada pita. Jika CC != MARK maka EOP akan
   padam (false).
          Jika CC = MARK maka EOP akan menyala (true) */

void ADVINPUT();
/* Pita dimajukan satu karakter.
   I.S. : Karakter pada jendela = CC, CC != MARK
   F.S. : CC adalah karakter berikutnya dari CC yang lama,
          CC mungkin = MARK
          Jika CC = MARK maka EOP akan menyala (true) */
```

ADT mesin karakter input yang kami gunakan untuk memenuhi Tugas Besar 2 ini merupakan adaptasi unik dari ADT mesinkar yang telah diberikan pada masa lampau untuk menangani

STEI- ITB	I	Halaman 19 dari 53 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.		

sebuah pita karakter tetapi sebagai bentuk input dari user. Pada Tugas Besar 2 ini struktur data yang menggunakan bentuk pita karakter ini kami gunakan sebagai elemen utama dalam pengolahan command use. Pemilihan struktur data ini diperlukan sebagai implementasi utama pengolahan suatu string yang dikirimkan oleh user atau akan diolah oleh program dan bagaimana user berinteraksi dengan program.

3.13 Mesin Karakter Komparasi

```
void SETSTRINGA(char s[MaxLengthString]);
/* Load string s untuk dibaca oleh mesinkar */

void STARTA();
/* Mesin siap dioperasikan. Pita disiapkan untuk dibaca.
   Karakter pertama yang ada pada pita posisinya adalah pada jendela.
   I.S. : sembarang
   F.S. : CC adalah karakter pertama pada pita
          Jika CC != MARKKomp maka EOP akan padam (false)
          Jika CC = MARKKomp maka EOP akan menyala (true) */

void ADVA();
/* Pita dimajukan satu karakter.
   I.S. : Karakter pada jendela = CC, CC != MARKKomp
   F.S. : CC adalah karakter berikutnya dari CC yang lama,
          CC mungkin = MARKKomp
          Jika CC = MARKKomp maka EOP akan menyala (true) */

void SETSTRINGB(char s[MaxLengthString]);
/* Load string s untuk dibaca oleh mesinkar */

void STARTB();
/* Mesin siap dioperasikan. Pita disiapkan untuk dibaca.
   Karakter pertama yang ada pada pita posisinya adalah pada jendela.
   I.S. : sembarang
   F.S. : CC adalah karakter pertama pada pita
          Jika CC != MARKKomp maka EOP akan padam (false)
          Jika CC = MARKKomp maka EOP akan menyala (true) */

void ADVB();
/* Pita dimajukan satu karakter.
   I.S. : Karakter pada jendela = CC, CC != MARKKomp
   F.S. : CC adalah karakter berikutnya dari CC yang lama,
          CC mungkin = MARKKomp
          Jika CC = MARKKomp maka EOP akan menyala (true) */
```

ADT mesin karakter komparasi yang kami gunakan untuk memenuhi Tugas Besar 2 ini merupakan adaptasi unik dari ADT mesinkar yang telah diberikan pada masa lampau untuk menangani sebuah pita karakter tetapi sebagai fungsi-fungsi primitif untuk menjalankan perbandingan antar 2 string. Pada Tugas Besar 2 ini struktur data yang menggunakan bentuk pita karakter ini kami gunakan sebagai elemen utama dalam pencocokan, matching, dan bagaimana pengolahan string dapat dilakukan. Pemilihan struktur data ini diperlukan sebagai implementasi utama pengolahan suatu string yang dikirimkan oleh user atau akan diolah oleh program dan bagaimana data-data baru diterima oleh program.

3.14 Mesin Kata Komparasi

```
void IgnoreBlankA();
/* Mengabaikan satu atau beberapa BLANK
   I.S. : CC sembarang
   F.S. : CC ≠ BLANK atau CC = MARK */
void RESETKATAA();
void STARTKATAA(char s[MaxLengthString]);
/* I.S. : CC sembarang
   F.S. : EndKata = true, dan CC = MARK;
          atau EndKata = false, CKata adalah kata yang sudah diakuisisi,
          CC karakter pertama sesudah karakter terakhir kata */

void ADVKATAA();
/* I.S. : CC adalah karakter pertama kata yang akan diakuisisi
   F.S. : CKata adalah kata terakhir yang sudah diakuisisi,
          CC adalah karakter pertama dari kata berikutnya, mungkin MARK
          Jika CC = MARK, EndKata = true.
   Proses : Akuisisi kata menggunakan procedure SalinKata */

void SalinKataA();
/* Mengakuisisi kata, menyimpan dalam CKata
   I.S. : CC adalah karakter pertama dari kata
   F.S. : CKata berisi kata yang sudah diakuisisi;
          CC = BLANK atau CC = MARK;
          CC adalah karakter sesudah karakter terakhir yang diakuisisi.
          Jika panjang kata melebihi NMax, maka sisa kata "dipotong" */

void IgnoreBlankB();
/* Mengabaikan satu atau beberapa BLANK
   I.S. : CC sembarang
   F.S. : CC ≠ BLANK atau CC = MARK */
void RESETKATAB();
void STARTKATAB(char s[MaxLengthString]);
/* I.S. : CC sembarang
   F.S. : EndKata = true, dan CC = MARK;
          atau EndKata = false, CKata adalah kata yang sudah diakuisisi,
          CC karakter pertama sesudah karakter terakhir kata */

void ADVKATAB();
/* I.S. : CC adalah karakter pertama kata yang akan diakuisisi
   F.S. : CKata adalah kata terakhir yang sudah diakuisisi,
          CC adalah karakter pertama dari kata berikutnya, mungkin MARK
          Jika CC = MARK, EndKata = true.
   Proses : Akuisisi kata menggunakan procedure SalinKata */

void SalinKataB();
/* Mengakuisisi kata, menyimpan dalam CKata
   I.S. : CC adalah karakter pertama dari kata
   F.S. : CKata berisi kata yang sudah diakuisisi;
          CC = BLANK atau CC = MARK;
          CC adalah karakter sesudah karakter terakhir yang diakuisisi.
```

Jika panjang kata melebihi NMax, maka sisa kata "dipotong" */

ADT mesin kata komparasi yang kami gunakan untuk memenuhi Tugas Besar 2 ini merupakan adaptasi unik dari ADT mesinkata yang telah diberikan pada masa lampau untuk menangani sebuah kata tetapi sebagai fungsi-fungsi primitif untuk menjalankan perbandingan antar 2 kata. Pada Tugas Besar 2 ini struktur data yang menggunakan bentuk kata ini kami gunakan sebagai elemen utama dalam pencocokan, matching, dan bagaimana pengolahan kata dapat dilakukan yang nantinya dapat digunakan untuk mengolah suatu string yang lebih panjang. Pemilihan struktur data ini diperlukan sebagai implementasi utama pengolahan suatu string.

3.15 Movement

```
void GoUp(MATRIKS *M, POINT *P, JAM *J);
```

```
void GoDown(MATRIKS *M, POINT *P, JAM *J);
```

```
void GoLeft(MATRIKS *M, POINT *P, JAM *J);
```

```
void GoRight(MATRIKS *M, POINT *P, JAM *J);
```

ADT movement yang kami gunakan untuk memenuhi Tugas Besar 2 ini merupakan implementasi unik dari jam, point dan matriks yang dapat digunakan untuk melakukan perubahan-perubahan terhadap koordinat player dengan 4 arah up, down, left dan right. Pada Tugas Besar 2 ini struktur data yang menggunakan bentuk primitif untuk menggerakkan ini kami gunakan sebagai mekanik utama dalam penggerakan karakter yang dapat mengurangi tick. Pemilihan struktur data ini diperlukan sebagai cara untuk mempermudah mengolah perubahan koordinat player yang berbentuk Point dan bagaimana cara agar mekanisme waktu terus berjalan.

3.16 Pelanggan

```
{
    int noRuangan;
    POINT posisi;
    int banyak;
    JAM waktuCabut;
    Kata orderan;
    boolean isStarPelanggan;
    boolean sudahOrder;
} PELANGGAN ;
```

ADT pelanggan yang kami gunakan untuk memenuhi Tugas Besar 2 ini merupakan ADT baru yang diperlukan untuk mengolah setiap pelanggan yang ada di permainan. Pada Tugas Besar 2 ini struktur data yang menggunakan bentuk primitif pelanggan untuk mengetahui apa pesannya, berapa banyak jumlahnya dan berapa kesabarannya juga lokasinya. Pemilihan struktur data ini diperlukan sebagai cara untuk mempermudah mengolah data pelanggan-pelanggan yang baru datang dan bagaimana pelanggan-pelanggan yang sudah lampau dikelola.

3.17 Point

```
/* *** DEFINISI PROTOTIPE PRIMITIF *** */
/* *** Konstruktor membentuk POINT *** */
```

STEI- ITB	I	Halaman 22 dari 53 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.		

```

POINT MakePOINT(int X, int Y);
/* Membentuk sebuah POINT dari komponen-komponennya */

/* *** Kelompok operasi relasional terhadap POINT *** */
boolean PointEQ(POINT P1, POINT P2);
/* Mengirimkan true jika P1 = P2 : absis dan ordinatnya sama */

/* *** KELOMPOK OPERASI LAIN TERHADAP TYPE *** */
POINT NextX(POINT P);
/* Mengirim salinan P dengan absis ditambah satu */
POINT NextY(POINT P);
/* Mengirim salinan P dengan ordinat ditambah satu */
POINT PrevX(POINT P);
/* Mengirim salinan P dengan absis dikurang satu */
POINT PrevY(POINT P);
/* Mengirim salinan P dengan ordinat dikurang satu */

```

ADT point yang kami gunakan untuk memenuhi Tugas Besar 2 ini merupakan adaptasi ulang dari ADT point yang telah diberikan pada masa lampau untuk menangani 2 buah integer yang akan diolah menjadi titik. Pada Tugas Besar 2 ini, struktur data yang menggunakan bentuk point ini kami gunakan sebagai penentu koordinat player dan pelanggan dan juga meja di layar. Pemilihan struktur data ini diperlukan sebagai salah satu cara untuk mengolah mekanika utama permainan yaitu bagaimana penentuan koordinat dan lokasi setiap karakter di permainan.

3.18 Queue

```

/* ***** Prototype ***** */
boolean IsEmptyQueue (Queue Q);
/* Mengirim true jika Q kosong: lihat definisi di atas */
boolean IsFullQueue (Queue Q);
/* Mengirim true jika tabel penampung elemen Q sudah penuh */
/* yaitu mengandung elemen sebanyak MaxElQueue */
int NBelmt (Queue Q);
/* Mengirimkan banyaknya elemen queue. Mengirimkan 0 jika Q kosong. */

/* *** Kreator *** */
void CreateEmptyQueue (Queue * Q, int Max);
/* I.S. sembarang */
/* F.S. Sebuah Q kosong terbentuk dan salah satu kondisi sbb: */
/* Jika alokasi berhasil, Tabel memori dialokasi berukuran Max+1 */
/* atau : jika alokasi gagal, Q kosong dg MaxElQueue=0 */
/* Proses : Melakukan alokasi, membuat sebuah Q kosong */

/* *** Destruktor *** */
void DeAlokasi(Queue * Q);
/* Proses: Mengembalikan memori Q */
/* I.S. Q pernah dialokasi */
/* F.S. Q menjadi tidak terdefinisi lagi, MaxElQueue(Q) diset 0 */

/* *** Primitif Add/Delete *** */
void Add (Queue * Q, infotypeQueue X);
/* Proses: Menambahkan X pada Q dengan aturan FIFO */
/* I.S. Q mungkin kosong, tabel penampung elemen Q TIDAK penuh */

```

```

/* F.S. X menjadi TAIL yang baru, TAIL "maju" dengan mekanisme circular
buffer */
void Del (Queue * Q, infotypeQueue * X);
/* Proses: Menghapus X pada Q dengan aturan FIFO */
/* I.S. Q tidak mungkin kosong */
/* F.S. X = nilQueueai elemen HEAD pd I.S., HEAD "maju" dengan mekanisme
circular buffer;
    Q mungkin kosong */

```

ADT queue yang kami gunakan untuk memenuhi Tugas Besar 2 ini merupakan adaptasi ulang dari ADT queue yang telah diberikan pada masa lampau untuk menangani suatu antrian integer yang dapat dilakukan operasi primitif padanya. Pada Tugas Besar 2 ini, struktur data yang menggunakan bentuk queue ini kami gunakan sebagai penentu urutan pelayanan pelanggan sebagai bentuk antrian. Pemilihan struktur data ini diperlukan sebagai salah satu cara untuk mempermudah pengelolaan pelanggan yang tiba satu per satu di tempat.

3.19 Stack

```

/* ***** Prototype ***** */
/* *** Konstruktor/Kreator *** */
void CreateEmptyStack (Stack *S);
/* I.S. sembarang; */
/* F.S. Membuat sebuah stack S yang kosong berkapasitas MaxElStack */
/* jadi indeksnya antara 1.. MaxElStack+1 karena 0 tidak dipakai */
/* Ciri stack kosong : TOP bernilStackai NilStack */

/* ***** Predikat Untuk test keadaan KOLEKSI ***** */
boolean IsEmptyStack (Stack S);
/* Mengirim true jika Stack kosong: lihat definisi di atas */
boolean IsFullStack (Stack S);
/* Mengirim true jika tabel penampung nilStackai elemen stack penuh */

/* ***** Menambahkan sebuah elemen ke Stack ***** */
void Push (Stack * S, infotypeStack X);
/* Menambahkan X sebagai elemen Stack S. */
/* I.S. S mungkin kosong, tabel penampung elemen stack TIDAK penuh */
/* F.S. X menjadi TOP yang baru, TOP bertambah 1 */

/* ***** Menghapus sebuah elemen Stack ***** */
void Pop (Stack * S, infotypeStack* X);
/* Menghapus X dari Stack S. */
/* I.S. S tidak mungkin kosong */
/* F.S. X adalah nilStackai elemen TOP yang lama, TOP berkurang 1 */

```

ADT stack yang kami gunakan untuk memenuhi Tugas Besar 2 ini merupakan adaptasi ulang dari ADT stack yang telah diberikan pada masa lampau untuk menangani suatu stack integer yang dapat dilakukan operasi primitif padanya, pada program ini stack dapat mengolah berbagai hal. Pada Tugas Besar 2 ini, struktur data yang menggunakan bentuk stack ini kami gunakan sebagai konstruktor food stack dari karakter utama. Pemilihan struktur data ini diperlukan sebagai salah satu cara untuk mempermudah makanan yang telah dibawa oleh pemain.

4 Program Utama

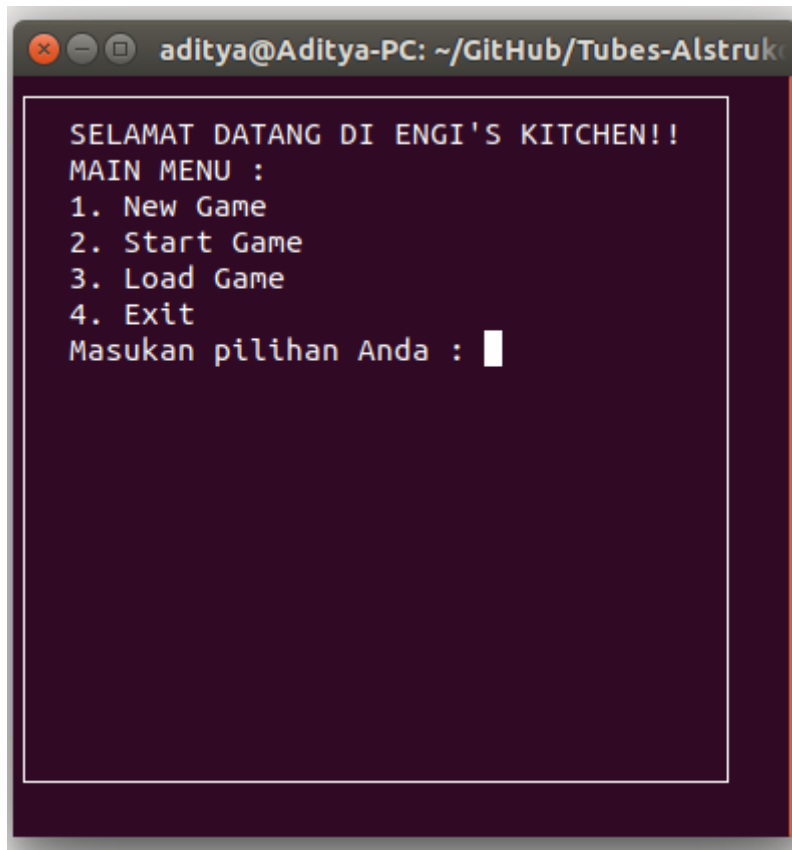
Permainan dimulai dengan memunculkan menu utama, sebelum masuk ke Start Game, player terlebih dahulu harus memilih menu New Game atau Load Game. Setelah permainan dimulai, semua variabel akan terisi dengan nilai default sesuai dengan file default.txt di folder Default Save, kecuali player memilih opsi Load Game. Kemudian permainan akan terus berjalan hingga player kalah atau memilih keluar dari permainan. Setiap aksi yang dilakukan saat permainan, gambar di layar dan data akan selalu terbaharui (update), serta waktu bertambah sebanyak 1 tick, dan program akan melakukan validasi kesabaran pelanggan.

5 Algoritma-Algoritma Menarik

5.1 *Recursive Descent*

Parser rekursif yang merupakan salah satu jenis parser top-down, yang juga saat ini dipelajari di matakuliah Tata Bahasa Formal dan Otomata (TBFO). Implementasi dari recursive decent pada tubes kali ini adalah untuk membaca tree dari file eksternal.

6 Data Test

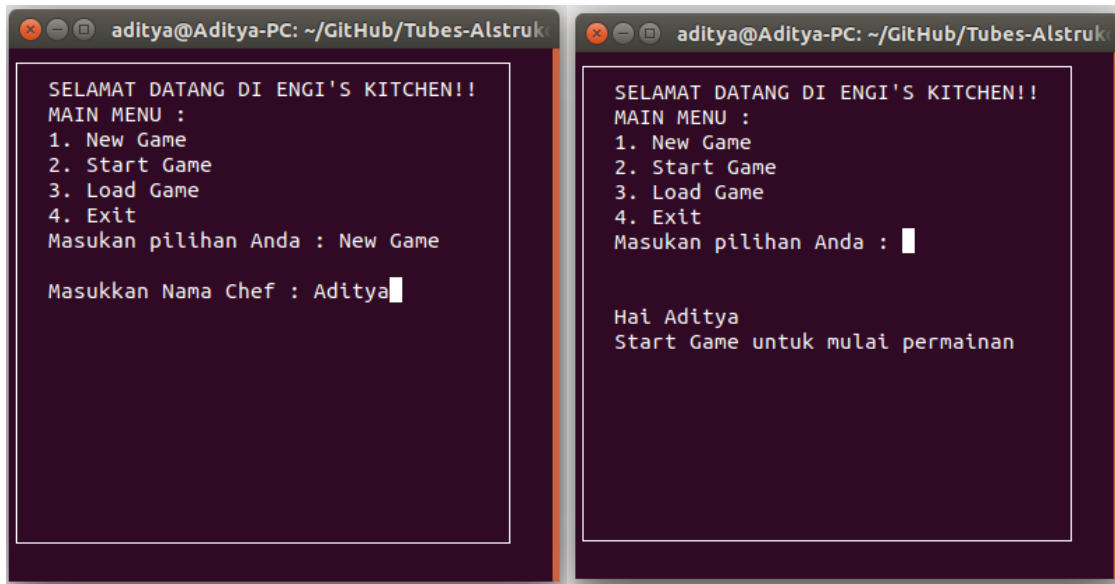


```
aditya@Aditya-PC: ~/GitHub/Tubes-Alstruko
SELAMAT DATANG DI ENGI'S KITCHEN!!
MAIN MENU :
1. New Game
2. Start Game
3. Load Game
4. Exit
Masukan pilihan Anda : 
```

Tampilan Awal

6.1 Data Test 1

Fitur New Game, sesuai dengan harapan.



6.2 Data Test 2

Fitur Place, menaruh pelanggan ke meja makan, sesuai harapan.

Nama : Aditya	Uang : 0	Nyawa :5	Waktu :9
Antrean(Banyak,Waktu pergi) 2* 26 4* 27	Player :3, 1 <div style="border: 1px solid black; width: 100%; height: 100%; position: relative; margin-top: 10px;"> <!-- Empty cells --> <div style="position: absolute; top: 5px; left: 5px;"> </div> <div style="position: absolute; top: 5px; left: 150px;"> </div> <div style="position: absolute; top: 5px; left: 250px;"> </div> <div style="position: absolute; top: 5px; left: 350px;"> </div> <div style="position: absolute; top: 5px; left: 450px;"> </div> <div style="position: absolute; top: 5px; left: 550px;"> </div> <div style="position: absolute; top: 5px; left: 650px;"> </div> <!-- Row 2 --> <div style="position: absolute; top: 30px; left: 5px;">X</div> <div style="position: absolute; top: 30px; left: 100px;">1</div> <div style="position: absolute; top: 30px; left: 150px;">X</div> <div style="position: absolute; top: 30px; left: 250px;">P</div> <div style="position: absolute; top: 30px; left: 350px;"> </div> <div style="position: absolute; top: 30px; left: 450px;">X</div> <div style="position: absolute; top: 30px; left: 550px;">2</div> <div style="position: absolute; top: 30px; left: 650px;">X</div> <!-- Row 3 --> <div style="position: absolute; top: 55px; left: 5px;"> </div> <div style="position: absolute; top: 55px; left: 100px;">X</div> <div style="position: absolute; top: 55px; left: 150px;"> </div> <div style="position: absolute; top: 55px; left: 250px;"> </div> <div style="position: absolute; top: 55px; left: 350px;"> </div> <div style="position: absolute; top: 55px; left: 450px;"> </div> <div style="position: absolute; top: 55px; left: 550px;"> </div> <div style="position: absolute; top: 55px; left: 650px;"> </div> <!-- Row 4 --> <div style="position: absolute; top: 80px; left: 5px;"> </div> <div style="position: absolute; top: 80px; left: 100px;"> </div> <div style="position: absolute; top: 80px; left: 150px;"> </div> <div style="position: absolute; top: 80px; left: 250px;"> </div> <div style="position: absolute; top: 80px; left: 350px;"> </div> <div style="position: absolute; top: 80px; left: 450px;"> </div> <div style="position: absolute; top: 80px; left: 550px;"> </div> <div style="position: absolute; top: 80px; left: 650px;"> </div> <!-- Row 5 --> <div style="position: absolute; top: 105px; left: 5px;"> </div> <div style="position: absolute; top: 105px; left: 100px;">X</div> <div style="position: absolute; top: 105px; left: 150px;"> </div> <div style="position: absolute; top: 105px; left: 250px;"> </div> <div style="position: absolute; top: 105px; left: 350px;"> </div> <div style="position: absolute; top: 105px; left: 450px;"> </div> <div style="position: absolute; top: 105px; left: 550px;"> </div> <div style="position: absolute; top: 105px; left: 650px;"> </div> <!-- Row 6 --> <div style="position: absolute; top: 130px; left: 5px;">X</div> <div style="position: absolute; top: 130px; left: 100px;">3</div> <div style="position: absolute; top: 130px; left: 150px;">X</div> <div style="position: absolute; top: 130px; left: 250px;"> </div> <div style="position: absolute; top: 130px; left: 350px;"> </div> <div style="position: absolute; top: 130px; left: 450px;">X</div> <div style="position: absolute; top: 130px; left: 550px;">4</div> <div style="position: absolute; top: 130px; left: 650px;">X</div> <!-- Row 7 --> <div style="position: absolute; top: 155px; left: 5px;"> </div> <div style="position: absolute; top: 155px; left: 100px;">X</div> <div style="position: absolute; top: 155px; left: 150px;"> </div> <div style="position: absolute; top: 155px; left: 250px;"> </div> <div style="position: absolute; top: 155px; left: 350px;"> </div> <div style="position: absolute; top: 155px; left: 450px;"> </div> <div style="position: absolute; top: 155px; left: 550px;"> </div> <div style="position: absolute; top: 155px; left: 650px;"> </div> </div>		Food Stack <div style="border: 1px solid black; width: 100%; height: 100%; margin-top: 10px;"></div>
Order			Hand Stack
Masukan pilihan Anda : PLACE <div style="border: 1px solid black; width: 50px; height: 20px; display: inline-block; margin-left: 5px;"></div>			

<div style="background-color: #f0f0f0; padding: 2px; border: 1px solid #ccc; margin-bottom: 5px;">Search your computer</div>	Uang : 0	Nyawa :5	Waktu :10																																											
<p>Antrean(Banyak,Waktu pergi) 4* 27</p>	<p>Player :3, 1</p>		<p>Food Stack</p>																																											
<p>Order</p>	<table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td> x </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td>c 1 c p </td><td> x 2 </td><td>x</td></tr> <tr><td> x </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> x </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td>x 3 x </td><td> </td><td> x 4 </td><td>x</td></tr> <tr><td> x </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </table>		x						c 1 c p	x 2	x	x																		x						x 3 x		x 4	x	x						<p>Hand Stack</p>
x																																														
c 1 c p	x 2	x																																												
x																																														
x																																														
x 3 x		x 4	x																																											
x																																														
<p>Masukan pilihan Anda : <input style="width: 50px;" type="text"/></p>																																														

6.3 Data Test 3

Fitur Order, menambahkan orderan, sesuai harapan.

Nama : Aditya	Uang : 0	Nyawa :5	Waktu :10																																																
Antrean(Banyak,Waktu pergi) 4* 27	Player :3, 1	Food Stack																																																	
Order	<table border="1"><tr><td> X </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>C 1 </td><td>C P </td><td> X </td><td>2 </td><td>X</td><td></td></tr><tr><td> X </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> X </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>X 3 </td><td>X </td><td> </td><td> X </td><td>4 </td><td>X</td></tr><tr><td> X </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr></table>			X						C 1	C P	X	2	X		X																		X						X 3	X		X	4	X	X					
	X																																																		
C 1	C P	X	2	X																																															
X																																																			
X																																																			
X 3	X		X	4	X																																														
X																																																			
		Hand Stack																																																	
Masukan pilihan Anda : ORDER																																																			

Nama : Aditya	Uang : 0	Nyawa :5	Waktu :11																																																								
Antrean(Banyak,Waktu pergi) 4* 27 2, 41	Player :3, 1 <div style="border: 1px solid black; width: 100%; height: 100%; position: relative;"> <div style="position: absolute; top: 0; left: 0; right: 0; bottom: 0; display: flex; flex-direction: column; align-items: center; justify-content: center;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td> </td><td>X</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td>C</td><td>1</td><td>C</td><td>P</td><td> </td><td>X</td><td>2</td></tr> <tr><td> </td><td>X</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td>X</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td>X</td><td>3</td><td>X</td><td> </td><td> </td><td>X</td><td>4</td></tr> <tr><td> </td><td>X</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </table> </div> </div>			X						C	1	C	P		X	2		X																					X						X	3	X			X	4		X						Food Stack
	X																																																										
C	1	C	P		X	2																																																					
	X																																																										
	X																																																										
X	3	X			X	4																																																					
	X																																																										
Order Nasi_Ayam_Goreng, 2	<div style="border: 1px solid black; width: 100%; height: 100%; position: relative;"> <div style="position: absolute; top: 0; left: 0; right: 0; bottom: 0; display: flex; flex-direction: column; align-items: center; justify-content: center;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td> </td><td>X</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td>X</td><td>3</td><td>X</td><td> </td><td> </td><td>X</td><td>4</td></tr> <tr><td> </td><td>X</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </table> </div> </div>			X						X	3	X			X	4		X						Hand Stack																																			
	X																																																										
X	3	X			X	4																																																					
	X																																																										
Masukan pilihan Anda : <input style="width: 50px;" type="text"/>																																																											

6.4 Data Test 4

Fitur Movement, memindahkan posisi player, sesuai harapan.

Nama : Aditya	Uang : 0	Nyawa :5	Waktu :5																																																
Antrean(Banyak,Waktu pergi) 2* 26 4* 27	Player :3, 3	Food Stack																																																	
Order	<table border="1"><tr><td> x </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>x 1 x </td><td> </td><td> </td><td>x 2 x</td><td></td><td></td></tr><tr><td> x </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> p </td><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> x </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>x 3 x </td><td> </td><td> </td><td>x 4 x</td><td></td><td></td></tr><tr><td> x </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr></table>			x						x 1 x			x 2 x			x								p										x						x 3 x			x 4 x			x					
	x																																																		
x 1 x			x 2 x																																																
x																																																			
		p																																																	
x																																																			
x 3 x			x 4 x																																																
x																																																			
Hand Stack																																																			
Masukan pilihan Anda : GD																																																			

Nama : Aditya	Uang : 0	Nyawa :5	Waktu :6																																																								
Antrean(Banyak,Waktu pergi) 2* 26 4* 27	Player :3, 4	Food Stack																																																									
Order	<table><tr><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td>1</td><td>X</td><td></td><td></td><td>X</td><td>2</td></tr><tr><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td>P</td><td></td><td></td><td></td></tr><tr><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td>3</td><td>X</td><td></td><td></td><td>X</td><td>4</td></tr><tr><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr></table>			X						X	1	X			X	2		X																P					X						X	3	X			X	4		X						Hand Stack
	X																																																										
X	1	X			X	2																																																					
	X																																																										
			P																																																								
	X																																																										
X	3	X			X	4																																																					
	X																																																										
Masukan pilihan Anda : █																																																											

Nama : Aditya	Uang : 0	Nyawa :5	Waktu :4																																																								
Antrean(Banyak,Waktu pergi) 2* 26 4* 27	Player :4, 3	Food Stack																																																									
Order	<table><tr><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td>1</td><td>X</td><td></td><td></td><td>X</td><td>2</td></tr><tr><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td>P</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td>3</td><td>X</td><td></td><td></td><td>X</td><td>4</td></tr><tr><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr></table>			X						X	1	X			X	2		X									P												X						X	3	X			X	4		X						Hand Stack
	X																																																										
X	1	X			X	2																																																					
	X																																																										
			P																																																								
	X																																																										
X	3	X			X	4																																																					
	X																																																										
Masukan pilihan Anda : GL█																																																											

Nama : Aditya	Uang : 0	Nyawa :5	Waktu :5																																																								
Antrean(Banyak,Waktu pergi) 2* 26 4* 27	Player :3, 3	Food Stack																																																									
Order	<table><tr><td> </td><td>X</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>X</td><td> </td><td>1</td><td> </td><td>X</td><td> </td><td> </td></tr><tr><td> </td><td>X</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td><td>P</td><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> </td><td>X</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>X</td><td> </td><td>3</td><td> </td><td>X</td><td> </td><td> </td></tr><tr><td> </td><td>X</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr></table>			X						X		1		X				X									P												X						X		3		X				X						Hand Stack
	X																																																										
X		1		X																																																							
	X																																																										
			P																																																								
	X																																																										
X		3		X																																																							
	X																																																										
Masukan pilihan Anda : █																																																											

Nama : Aditya	Uang : 0	Nyawa :5	Waktu :3																																																								
Antrean(Banyak,Waktu pergi) 2* 26 4* 27	Player :3, 3	Food Stack																																																									
Order	<table><tr><td> </td><td>X</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>X</td><td> </td><td>1</td><td> </td><td>X</td><td> </td><td> </td></tr><tr><td> </td><td>X</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td><td>P</td><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> </td><td>X</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>X</td><td> </td><td>3</td><td> </td><td>X</td><td> </td><td> </td></tr><tr><td> </td><td>X</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr></table>			X						X		1		X				X									P												X						X		3		X				X						Hand Stack
	X																																																										
X		1		X																																																							
	X																																																										
			P																																																								
	X																																																										
X		3		X																																																							
	X																																																										
Masukan pilihan Anda : GR█																																																											

Nama : Aditya	Uang : 0	Nyawa :5	Waktu :4																																																								
Antrean(Banyak,Waktu pergi) 2* 26 4* 27	Player :4, 3		Food Stack																																																								
Order	<table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td> </td><td>X</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td>X</td><td>1</td><td>X</td><td> </td><td> </td><td>X</td><td>2</td></tr> <tr><td> </td><td>X</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td>P</td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td>X</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td>X</td><td>3</td><td>X</td><td> </td><td> </td><td>X</td><td>4</td></tr> <tr><td> </td><td>X</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </table>			X						X	1	X			X	2		X										P											X						X	3	X			X	4		X						Hand Stack
	X																																																										
X	1	X			X	2																																																					
	X																																																										
				P																																																							
	X																																																										
X	3	X			X	4																																																					
	X																																																										
Masukan pilihan Anda : <input style="width: 50px;" type="text"/>																																																											

6.5 Data Test 5

Fitur Recipe, menampilkan resep (pohon makanan), sesuai harapan.

Nana : Aditya	Uang : 2500	Nyawa :5	Waktu :11																																																								
Antrean(Banyak,Waktu pergi) 2, 38	Player :4, 1	Food Stack																																																									
Order	<table border="1"><tr><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td>1</td><td>X</td><td></td><td>P</td><td>X</td><td>2</td></tr><tr><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>X</td><td>3</td><td>X</td><td></td><td></td><td>X</td><td>4</td></tr><tr><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr></table>			X						X	1	X		P	X	2		X																					X						X	3	X			X	4		X						Hand Stack
	X																																																										
X	1	X		P	X	2																																																					
	X																																																										
	X																																																										
X	3	X			X	4																																																					
	X																																																										
Masukan pilihan Anda : <input type="text" value="RECIPE"/>																																																											

Piring

- Sendok
 - EsKrim
 - Pisang
 - Banana_Split
 - Stroberi
 - Sundae
 - Nasi
 - Telur
 - Nasi_Telur_Dadar
 - Ayam_Goreng
 - Nasi_Ayam_Goreng
 - Garpu
 - Rotti
 - Patty
 - Burger
 - Sosis
 - Hot_Dog
 - Spaghetti
 - Bolognese
 - Keju
 - Spaghetti_Bolognese
 - Carbonara
 - Sphagetti_Carbonara

6.6 Data Test 6

Fitur Take, mengambil objek, sesuai harapan.

Nama : Aditya	Uang : 0	Nyawa :2	Waktu :52																																																																
Antrean(Banyak,Waktu pergi) 4* 66 4, 54 4, 69 4, 76	Player :1, 2	Food Stack																																																																	
Order	<table border="1"><tr><td>M</td><td></td><td></td><td></td><td></td><td></td><td></td><td>T</td></tr><tr><td>M</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>M</td><td>P</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>M</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>M</td><td></td><td></td><td>M</td><td>M</td><td>M</td><td></td><td></td></tr><tr><td>M</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>M</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>M</td><td></td><td></td><td>M</td><td>M</td><td>M</td><td>M</td><td>M</td></tr></table>		M							T	M								M	P							M								M			M	M	M			M								M								M			M	M	M	M	M	Hand Stack
M							T																																																												
M																																																																			
M	P																																																																		
M																																																																			
M			M	M	M																																																														
M																																																																			
M																																																																			
M			M	M	M	M	M																																																												
Masukan pilihan Anda : TAKE																																																																			

Nama : Aditya

Uang : 0

Nyawa :2

Waktu :53

Antrean(Banyak,Waktu pergi)
4* 66
4, 54
4, 69
4, 76

Player :1, 2

M										T
M										
M		P								
M										
M						M		M		M
M										
M										
M						M		M		M

Food Stack

Order

Hand Stack
EsKrim

Masukan pilihan Anda :

6.7 Data Test 7

Fitur Put, meletakkan objek, sesuai harapan.

Nama : Aditya	Uang : 0	Nyawa : 4	Waktu : 6																																																																																							
Antrean(Banyak,Waktu pergi)	Player : 6, 0	Food Stack																																																																																								
	<table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr><td>M</td><td> </td><td></td><td> </td><td></td><td> </td><td></td><td> </td><td>P</td><td> </td><td>T</td></tr> <tr><td>M</td><td> </td><td></td><td> </td><td></td><td> </td><td></td><td> </td><td></td><td> </td><td></td></tr> <tr><td>M</td><td> </td><td></td><td> </td><td></td><td> </td><td></td><td> </td><td></td><td> </td><td></td></tr> <tr><td>M</td><td> </td><td></td><td> </td><td></td><td> </td><td></td><td> </td><td></td><td> </td><td></td></tr> <tr><td>M</td><td> </td><td></td><td> </td><td>M</td><td> </td><td>M</td><td> </td><td>M</td><td> </td><td></td></tr> <tr><td>M</td><td> </td><td></td><td> </td><td></td><td> </td><td></td><td> </td><td></td><td> </td><td></td></tr> <tr><td>M</td><td> </td><td></td><td> </td><td></td><td> </td><td></td><td> </td><td></td><td> </td><td></td></tr> <tr><td>M</td><td> </td><td></td><td> </td><td>M</td><td> </td><td>M</td><td> </td><td>M</td><td> </td><td>M</td></tr> </table>	M								P		T	M											M											M											M				M		M		M			M											M											M				M		M		M		M	
M								P		T																																																																																
M																																																																																										
M																																																																																										
M																																																																																										
M				M		M		M																																																																																		
M																																																																																										
M																																																																																										
M				M		M		M		M																																																																																
Order		Hand Stack Pisang EsKrim Sendok Piring																																																																																								
Masukan pilihan Anda : PUT																																																																																										

Nama : Aditya		Uang : 0		Nyawa :4		Waktu :7																																																																			
Antrean(Banyak,Waktu pergi)		Player :6, 0				Food Stack Banana_Split																																																																			
Order		<table><tr><td>M</td><td> </td><td> </td><td> </td><td> </td><td> </td><td>P</td><td>T</td></tr><tr><td>M</td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>M</td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>M</td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>M</td><td> </td><td> </td><td>M</td><td>M</td><td>M</td><td> </td><td> </td></tr><tr><td>M</td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>M</td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>M</td><td> </td><td> </td><td>M</td><td>M</td><td>M</td><td>M</td><td>M</td></tr></table>						M						P	T	M								M								M								M			M	M	M			M								M								M			M	M	M	M	M	Hand Stack	
		M						P	T																																																																
		M																																																																							
		M																																																																							
		M																																																																							
		M			M	M	M																																																																		
		M																																																																							
		M																																																																							
		M			M	M	M	M	M																																																																
Masukan pilihan Anda : <input type="text"/>																																																																									

6.8 Data Test 8

Fitur Save, menyimpan permainan, sesuai harapan.

Nama : Aditya	Uang : 0	Nyawa :5	Waktu :11																																																
Antrean(Banyak,Waktu pergi) 4* 27 2, 41	Player :3, 1	Food Stack																																																	
<table border="1"><tr><td> X </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>C 1 </td><td>C P </td><td> X </td><td>2 </td><td>X</td><td></td></tr><tr><td> X </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> X </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>X 3 </td><td>X </td><td> </td><td> X </td><td>4 </td><td>X</td></tr><tr><td> X </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr></table>				X						C 1	C P	X	2	X		X																		X						X 3	X		X	4	X	X					
X																																																			
C 1	C P	X	2	X																																															
X																																																			
X																																																			
X 3	X		X	4	X																																														
X																																																			
Order Nasi_Ayam_Goreng, 2	Hand Stack																																																		
Masukan pilihan Anda : SAVE																																																			

Nama : Aditya

Uang : 0

Nyawa :5

Waktu :11

Antrean(Banyak,Waktu pergl)

4* 27

2, 41

Player :3, 1

	x					
c	1	c	p		x	2
	x					
	x					
x	3	x			x	4
	x					

Food Stack

Hand Stack

Order

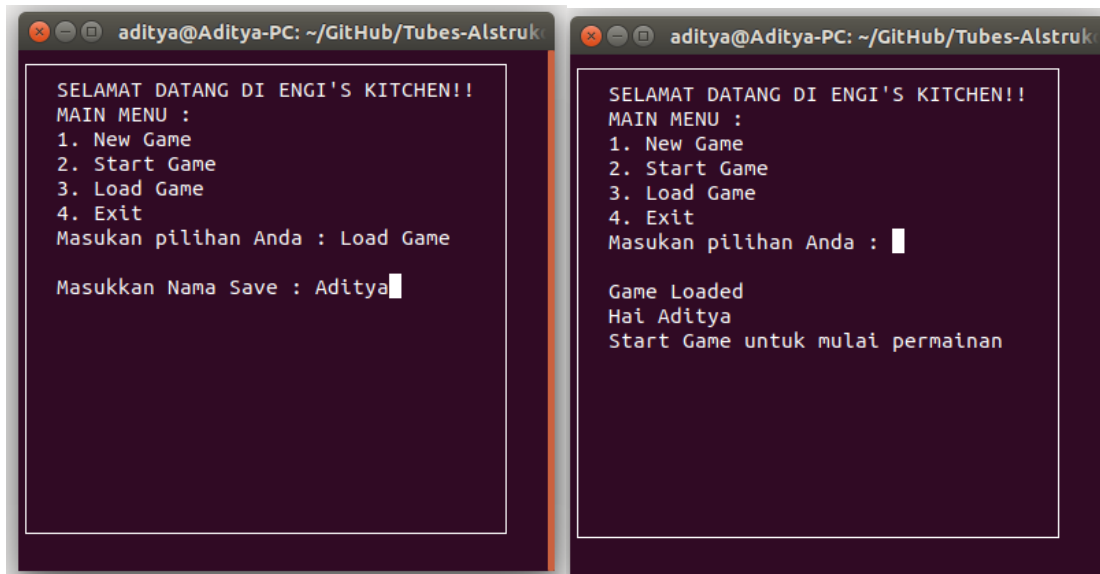
Nasi_Ayam_Goreng, 2

Masukan pilihan Anda :

Save berhasil

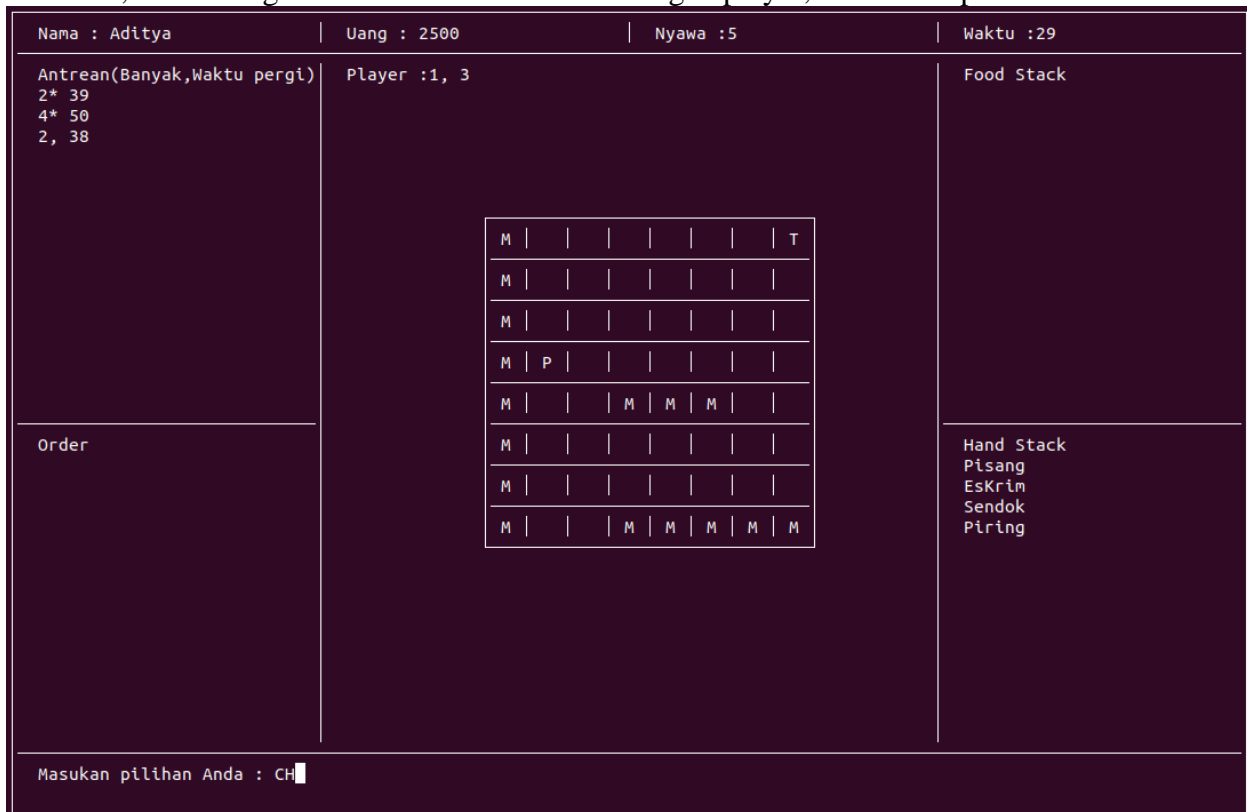
6.9 Data Test 9

Fitur Load, memuat permainan yang sudah di simpan, sesuai harapan.



6.10 Data Test 10

Fitur CH, membuang seluruh bahan makanan di tangan player, sesuai harapan.



Nama : Aditya	Uang : 2500	Nyawa :5	Waktu :30																																																																
Antrean(Banyak,Waktu pergi) 2* 39 4* 50 2, 38	Player :1, 3		Food Stack																																																																
Order	<table border="1" style="margin: auto; border-collapse: collapse; text-align: center;"> <tr><td>M</td><td></td><td></td><td></td><td></td><td></td><td></td><td>T</td></tr> <tr><td>M</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>M</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>M</td><td>P</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>M</td><td></td><td></td><td>M</td><td>M</td><td>M</td><td></td><td></td></tr> <tr><td>M</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>M</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>M</td><td></td><td></td><td>M</td><td>M</td><td>M</td><td>M</td><td>M</td></tr> </table>		M							T	M								M								M	P							M			M	M	M			M								M								M			M	M	M	M	M	Hand Stack
M							T																																																												
M																																																																			
M																																																																			
M	P																																																																		
M			M	M	M																																																														
M																																																																			
M																																																																			
M			M	M	M	M	M																																																												
Masukan pilihan Anda : <input style="width: 50px;" type="text"/>																																																																			

6.11 Data Test 11

Fitur CT, membuang seluruh bahan makanan di nampan, sesuai harapan.

Nama : Aditya	Uang : 0	Nyawa :5	Waktu :10																																																
Antrean(Banyak,Waktu pergi) 2, 38	Player :4, 1	Food Stack Nasi_Telur_Dadar																																																	
<table border="1"><tr><td> x </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>x 1 x </td><td> </td><td>p c 2 c</td><td colspan="2"></td><td></td></tr><tr><td> x </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> x </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>x 3 x </td><td> </td><td> </td><td>x 4 x</td><td colspan="2"></td></tr><tr><td> x </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr></table>				x						x 1 x		p c 2 c				x																		x						x 3 x			x 4 x			x					
x																																																			
x 1 x		p c 2 c																																																	
x																																																			
x																																																			
x 3 x			x 4 x																																																
x																																																			
Order Nasi_Telur_Dadar, 2	Hand Stack																																																		
Masukan pilihan Anda : CT																																																			
Game Loaded																																																			

Nama : Aditya	Uang : 0	Nyawa :5	Waktu :11
Antrean(Banyak,Waktu pergi) 2, 38	Player :4, 1 <div style="border: 1px solid black; width: 100%; height: 100%; position: relative; margin-top: 10px;"> <!-- Empty cells --> <div style="position: absolute; top: 0; left: 0; width: 100%; height: 100%; background-color: #f0f0f0;"></div> <!-- Player 1 --> <div style="position: absolute; top: 10%; left: 10%; width: 10%; height: 10%; background-color: #000080; color: white; text-align: center; line-height: 10px;">1</div> <!-- Player 2 --> <div style="position: absolute; top: 10%; left: 30%; width: 10%; height: 10%; background-color: #000080; color: white; text-align: center; line-height: 10px;">2</div> <!-- Player 3 --> <div style="position: absolute; top: 10%; left: 50%; width: 10%; height: 10%; background-color: #000080; color: white; text-align: center; line-height: 10px;">3</div> <!-- Player 4 --> <div style="position: absolute; top: 10%; left: 70%; width: 10%; height: 10%; background-color: #000080; color: white; text-align: center; line-height: 10px;">4</div> <!-- Empty cells --> <div style="position: absolute; top: 10%; left: 80%; width: 10%; height: 10%; background-color: #f0f0f0;"></div> <!-- Empty cells --> <div style="position: absolute; top: 30%; left: 10%; width: 10%; height: 10%; background-color: #f0f0f0;"></div> <!-- Empty cells --> <div style="position: absolute; top: 30%; left: 30%; width: 10%; height: 10%; background-color: #f0f0f0;"></div> <!-- Empty cells --> <div style="position: absolute; top: 30%; left: 50%; width: 10%; height: 10%; background-color: #f0f0f0;"></div> <!-- Empty cells --> <div style="position: absolute; top: 30%; left: 70%; width: 10%; height: 10%; background-color: #f0f0f0;"></div> <!-- Empty cells --> <div style="position: absolute; top: 30%; left: 80%; width: 10%; height: 10%; background-color: #f0f0f0;"></div> <!-- Empty cells --> <div style="position: absolute; top: 50%; left: 10%; width: 10%; height: 10%; background-color: #f0f0f0;"></div> <!-- Empty cells --> <div style="position: absolute; top: 50%; left: 30%; width: 10%; height: 10%; background-color: #f0f0f0;"></div> <!-- Empty cells --> <div style="position: absolute; top: 50%; left: 50%; width: 10%; height: 10%; background-color: #f0f0f0;"></div> <!-- Empty cells --> <div style="position: absolute; top: 50%; left: 70%; width: 10%; height: 10%; background-color: #f0f0f0;"></div> <!-- Empty cells --> <div style="position: absolute; top: 50%; left: 80%; width: 10%; height: 10%; background-color: #f0f0f0;"></div> <!-- Empty cells --> <div style="position: absolute; top: 70%; left: 10%; width: 10%; height: 10%; background-color: #f0f0f0;"></div> <!-- Empty cells --> <div style="position: absolute; top: 70%; left: 30%; width: 10%; height: 10%; background-color: #f0f0f0;"></div> <!-- Empty cells --> <div style="position: absolute; top: 70%; left: 50%; width: 10%; height: 10%; background-color: #f0f0f0;"></div> <!-- Empty cells --> <div style="position: absolute; top: 70%; left: 70%; width: 10%; height: 10%; background-color: #f0f0f0;"></div> <!-- Empty cells --> <div style="position: absolute; top: 70%; left: 80%; width: 10%; height: 10%; background-color: #f0f0f0;"></div> </div>		Food Stack
Order Nasi_Telur_Dadar, 2	Hand Stack		Hand Stack
Masukan pilihan Anda : <div style="border: 1px solid black; width: 100px; height: 20px; display: inline-block;"></div>			

6.12 Data Test 12

Fitur Give, memberikan makanan ke pelanggan, sesuai harapan.

Nama : Aditya	Uang : 0	Nyawa :5	Waktu :10																																																
Antrean(Banyak,Waktu pergi) 2, 38	Player :4, 1	Food Stack Nasi_Telur_Dadar																																																	
<table><tr><td> x </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>x 1 x </td><td> </td><td>p c 2 c</td><td colspan="3"></td></tr><tr><td> x </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> x </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>x 3 x </td><td> </td><td> </td><td>x 4 x</td><td colspan="2"></td></tr><tr><td> x </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr></table>				x						x 1 x		p c 2 c				x																		x						x 3 x			x 4 x			x					
x																																																			
x 1 x		p c 2 c																																																	
x																																																			
x																																																			
x 3 x			x 4 x																																																
x																																																			
Order Nasi_Telur_Dadar, 2	Hand Stack																																																		
Masukan pilihan Anda : GIVE																																																			

Nama : Aditya		Uang : 2500		Nyawa : 5		Waktu : 11																																																																	
Antrean(Banyak,Waktu pergi) 2, 38		Player : 4, 1				Food Stack																																																																	
Order		<table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td> </td><td>x</td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td>x</td><td>1</td><td>x</td><td> </td><td>P</td><td>x</td><td>2</td><td>x</td></tr> <tr><td> </td><td>x</td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td>x</td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td>x</td><td>3</td><td>x</td><td> </td><td> </td><td>x</td><td>4</td><td>x</td></tr> <tr><td> </td><td>x</td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </table>					x							x	1	x		P	x	2	x		x																								x							x	3	x			x	4	x		x							Hand Stack	
	x																																																																						
x	1	x		P	x	2	x																																																																
	x																																																																						
	x																																																																						
x	3	x			x	4	x																																																																
	x																																																																						
Masukan pilihan Anda : <input style="width: 50px;" type="text"/>																																																																							

7 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	New Game	Mengetahui keberjalanan fitur.	Pada menu utama, ketik New Game, lalu player diminta memasukkan nama.	Data Test 1	Program meminta input nama.	Sesuai harapan.
2	Place	Mengetahui keberjalanan fitur.	Saat permainan berlangsung, ketik PLACE.	Data Test 2	Menaruh pelanggan di meja.	Sesuai harapan.
3	Order	Mengetahui keberjalanan fitur.	Saat permainan berlangsung, ketik ORDER.	Data Test 3	Menambahkan orderan.	Sesuai harapan.
4	Movement (GU, GL, GR, GD)	Mengetahui keberjalanan fitur.	Saat permainan berlangsung, ketik GU, GL, GR, atau GD.	Data Test 4	Player berpindah sesuai perintah.	Sesuai harapan.
5	Recipe	Mengetahui keberjalanan fitur.	Saat permainan berlangsung, ketik RECIPE.	Data Test 5	Menampilkan pohon makanan.	Sesuai harapan.
6	Take	Mengetahui keberjalanan fitur.	Saat permainan berlangsung, ketik TAKE.	Data Test 7	Mengambil objek.	Sesuai harapan.

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
7	Put	Mengetahui keberjalanan fitur.	Saat permainan berlangsung, ketik PUT.	Data Test 8	Meletakkan objek.	Sesuai harapan.
8	Save	Mengetahui keberjalanan fitur.	Saat permainan berlangsung, ketik SAVE.	Data Test 9	Menyimpan permainan.	Sesuai harapan.
9	Load	Mengetahui keberjalanan fitur.	Pada menu utama, ketik Load Game.	Data Test 10	Memuat permainan yang sudah di simpan.	Sesuai harapan.
10	CH	Mengetahui keberjalanan fitur.	Saat permainan berlangsung, ketik CH.	Data Test 11	Membuang seluruh bahan makanan di tangan player.	Sesuai harapan.
11	CT	Mengetahui keberjalanan fitur.	Saat permainan berlangsung, ketik CT.	Data Test 12	Membuang seluruh bahan makanan di nampan.	Sesuai harapan.
12	Give	Mengetahui keberjalanan fitur.	Saat permainan berlangsung, ketik GIVE.	Data Test 13	Memberikan makanan ke pelanggan.	Sesuai harapan.

8 Pembagian Kerja dalam Kelompok

NIM	Nama	Tugas
13517013	Aditya Putra Santoso	Mengurus ADT, Main Program, Implementasi ADT, Debugging
13517040	Ariel Ansa Razumardi	Mengurus ADT, Main Program, Implementasi ADT, Debugging
13517073	Rayza Mahendra Guntara Harsono	Mengurus ADT, Debugging
13517145	Muhammad Al Terra	Mengurus ADT, Implementasi ADT
13517151	Rakhmad Budiono	Mengurus Main Program, Implementasi ADT

9 Lampiran

9.1 Deskripsi Tugas Besar 2

Engi's Kitchen adalah restoran yang dirintis tahun lalu oleh Chef. Hanya dalam jangka waktu satu tahun, Engi's Kitchen sudah tergolong sangat sukses dan berjaya. Oleh karena itu, Chef sebagai seorang engineer selalu ingin berkembang dan berusaha untuk mengembangkan bisnisnya lebih lanjut. Pada tahun ini, Chef berani mengembangkan restorannya menjadi lebih besar. Sekarang pengunjung dapat menikmati makanannya di dalam restoran. Karena Chef ingin menjaga kualitas restorannya, makanan yang disajikan harus segar sehingga pembuatan makan harus dilakukan on-the-spot. Sebelum ia mengambil resiko untuk merenovasi dan mengembangkan bisnisnya, Chef meminta Anda untuk membuat sebuah program simulasi berjalannya restoran ini. Restoran Engi's Kitchen memiliki 3 ruang yang digunakan untuk tamu

STEI- ITB	I	Halaman 48 dari 53 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.		

dan 1 ruang yang digunakan sebagai dapur. Ruangan-ruangan ini terhubung dengan pintu. Pada setiap ruangan terdapat meja-meja yang disusun sedemikian rupa. Meja tersebut beragam kapasitasnya. Ada meja yang hanya bisa untuk berdua, dan ada juga yang mampu menampung 4 pengunjung. Pengunjung yang mendatangi restoran akan mengantre terlebih dahulu. Interval waktu kedatangan pelanggan dibebaskan. Chef lalu akan mengarahkan pengunjung yang berada pada antrean paling depan untuk duduk di meja yang cukup kapasitasnya. Setelah pengunjung duduk di meja, mereka akan langsung melakukan pemesanan makanan. Pengunjung yang sedang mengantre diharapkan menunggu maksimal selama 30 satuan waktu, jika pengunjung tersebut menunggu lebih lama maka pengunjung akan pulang dan kredibilitas restoran berkurang. Tugas Besar IF2110 Algoritma dan Struktur Data. Meskipun terdapat beberapa varietas makanan yang dijual di Engi's Kitchen. Dipastikan bahwa 1 meja hanya memesan 1 jenis makanan. Pesanan makanan tersebut akan disimpan pada suatu daftar, dan setiap pengunjung yang duduk juga memiliki batas waktu kesabaran seperti pada antrean. Namun, batas waktu kesabaran ini bervariasi setiap pengunjung. Pembuatan makanan dilakukan dengan menambahkan bahan makanan kepada piring. Pencampuran bahan makanan harus mengikuti urutan tertentu berdasarkan pohon makanan. Makanan yang telah terbentuk harus diletakkan di nampan terlebih dahulu sebelum dibawa ke pengunjung. Nampan mampu menampung 5 tumpuk makanan untuk sekali jalan. Makanan pada nampan berbentuk tumpukan. Setiap aksi yang dilakukan oleh Chef memakan 1 tick waktu, pada setiap tick mungkin terjadi penambahan pengunjung baru. Setiap 1 tick, kesabaran pengunjung berkurang 1 satuan. Chef sangat berharap dengan adanya program ini ia dapat mengatur strategi agar Engi's Kitchen semakin sukses dan terus berkembang kedepannya.

9.2 Notulen Rapat

2 November 2018

Pelanggan datang secara random merupakan sebuah bonus, diperbolehkan memodifikasi ADT. Satu meja diisi dengan satu pesanan. Food stack sesuai dengan tree.

12 November 2018

Pengolahan string harus menggunakan mesin kata. ADT Graph merupakan modifikasi dari ADT List. ADT Tree akan dirilis di Olympia.

23 November 2018

Variasi Graph cukup pakai List atau Matriks List saja. Boleh menggunakan ncurses, tidak boleh menggunakan string.h, pintu dibuat dari file eksternal.

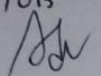
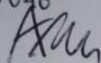
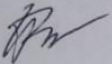

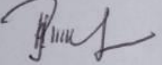

9.3 Log Activity Anggota Kelompok

	Aditya Putra Santoso	Ariel Ansa Razumardi	Rayza Mahendra Guntara Harsono	Muhammad Al Terra	Rakhmad Budiono
30-10-2018	Breakdown Masalah	Breakdown Masalah	Breakdown Masalah	Breakdown Masalah	Breakdown Masalah
3-11-2018	Desain	Mengumpulkan			

STEI- ITB	I	Halaman 49 dari 53 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.		

s.d. 6-11-2018	Timeline Pengerjaan	ADT Pra- Praktikum			
7-11-2018 s.d. 8-11-2018			Pembuatan ADT Baru	Pembuatan ADT Baru	Membuat Main Program Versi Pertama
9-11-2018 s.d. 12-11-2018	Integrasi Semua Pengerjaan	Membuat ADT Baru			
13-11-2018 s.d. 20-11-2018	Debugging	Debugging	Debugging	Debugging	Penyesuaian Main Program dengan ADT yang Ada
21-11-2018	Migrasi Planning, Mulai Menggunakan ncurses	Integrasi Semua Pengerjaan			
22-11-2018 s.d. 25-11-2018	Debugging, Finishing	Debugging, Finishing	Debugging	Membuat Laporan	Membuat Laporan

9.4 Form Asistensi

Asistensi I	
Tanggal : 2 November 2018	Catatan Asistensi:
Tempat : Labpro I	1. pelanggan datang secara random (bunyi)
Kehadiran Anggota Kelompok:	2. " " tidak random (bunyi)
No	3. boleh modif ADT.
NIM	4. 1 meja & pesanan
Tanda tangan	5. Meja & tameng & counter bisa namanya 1 set & pelanggan / 1 set & 2 pelanggan.
1	6. Food Stock sesuai tree agar bisa dibuat (can boleh kencana).
13517013	7. M. kelas isinya
	8. Per user 1 Save game
2	
13517040	
	
3	
13517073	
	
4	
13517045	
	
5	
13517151	
	
6	
13517151	
	Tanda Tangan Asisten: 

Asistensi II

Tanggal : 12 November 2018	Catatan Asistensi:
Tempat : Lab Pemrograman	
Kehadiran Anggota Kelompok:	<ul style="list-style-type: none"> - match string harus menggunakan mesin kata - graf modifikasi dari ADT list - tree ADT ganti dikeluarkan - Graf ambil dari kuliah itb.ac.id - tree sang
No	
NIM	
Tanda tangan	
1	
13517013	
2	
13517040	
3	
13517073	
4	
13517145	
5	
13517151	
6	
Tanda Tangan Asisten: Hdg	

Asistensi III

Tanggal : 23 November 2018

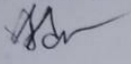
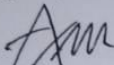
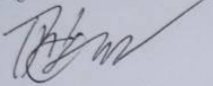
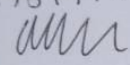
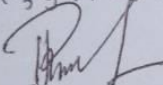
Tempat : LabPro I

Kehadiran Anggota Kelompok:

No

NIM

Tanda tangan

- | | |
|---|---|
| 1 | 13517013
 |
| 2 | 13517040
 |
| 3 | 13517073
 |
| 4 | 13517145
 |
| 5 | 13517151
 |
| 6 | |

Catatan Asistensi:

1. Apakah Variasi graf boleh
macam-macam, cek pakai
matriks?
- no. Pakai list aja atau
matriks list aja
2. Apakah boleh menggunakan array
- tidak apa-apa menggunakan
array untuk menyimpan
data
- asal tidak pakai string.h
3. boleh menghapus -hapus
dari .h
4. buat driven. x
5. they array of integer
6. Pinta dibuat dari file
externy tidak dihard code

Tanda Tangan Asisten:

