

TUGAS BESAR
ENGI'S KITCHEN

IF1210 DASAR PEMROGRAMAN
KELAS 04

Oleh

Kelompok 11

Asisten: Bervianto Leo P. / 13514047

Mgs. Muhammad Riandi Ramadhan	16517368
Haris Salman Al-Ghifary	16517375
Muhammad Fatkhi	16517382
Christofel Rio Goenawan	16517389
Habli Yahdi	16517396
Nurhayati Marqoh Sirfefa	16517403



INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2018

DAFTAR ISI

Daftar Isi.....	
Daftar Tabel.....	
Daftar Gambar.....	
1. Daftar Pembagian Kerja Kelompok.....	
2. Deskripsi Persoalan.....	
3. Desain Kamus Data.....	
4. Desain <i>Command</i>	
5. Desain Dekomposisi Persoalan.....	
6. Spesifikasi Fungsi.....	
7. Data Hasil Pengujian.....	
8. Lampiran.....	

Daftar Tabel

Tabel 1. Tabel Pembagian Kerja Anggota Kelompok
Tabel 2. Tabel Checklist Hasil Rancangan, Implementasi, dan Testing.....
Tabel 3. Tabel Spesifikasi Tiap Fitur Dalam Unit
Tabel 4. Tabel Pengujian Program

Daftar Gambar

1. Responsi 1 Lembar 1.....	28
2. Responsi 1 Lembar 2.....	29
3. Responsi 2 Lembar 1.....	30
4. Responsi 2 Lembar 2.....	31

1. DAFTAR PEMBAGIAN KERJA KELOMPOK

Tabel 1. Tabel Pembagian Kerja Anggota Kelompok

Fitur	Implementasi	NIM Desainer	NIM Koder	NIM Tester
F1- Load	Procedure membaca semua file untuk melakukan simulasi dan menyimpannya ke array	16517396	16517396	16517396/16517368
F2-Exit	Procedure untuk menyimpan semua perubahan data array ke file	16517396	16517396	1651796/16517368
F3-Start Simulasi	Procedure untuk membaca nomor simulasi dan tanggal simulasi	16517396	16517396	16517396/16517389
F4- Stop Simulasi	Procedure TampilkanStatistik	16517389	16517389	16517389
F5- Beli Bahan	Procedure BeliBahan; function CekAdaGak; function NemulIndex; function NemulIndexInventaris;	16517389	16517389	16517389
F6- Olah Bahan	Procedure BeliOlah; function NemulIndexOlahan;	16517389	16517389	16517389

	function CekPersediaanInv entaris; function CekBahanOlah; procedure TulisBahanKuran g			
F7- Jual Olahan	Procedure jual olahan	16517375	16517375	16517375/16517 389
F8- Jual Resep	Procedure jual resep	16517375	16517375	16517375
F9- Makan	Procedure Makan	16517375	16517375	16517375/16517 368
F10- Istirahat	Procedure Istirahat	16517382	16517382	16517382/16517 389
F11- Tidur	Procedure Tidur	16517382	16517382	16517382/16517 368
F12- Lihat Statistik	Procedure TampilStat	16517382	16517382	16517382/16517 389
F13- Lihat Inventori	Procedure LihatInventori	16517368	16517368	16517368
F14- Lihat Resep	Procedure LihatResep	16517368	16517368	16517368
F15- Cari Resep	Procedure CariResep	16517368	16517368	16517368
F16- Tambah Resep	Procedure TambahResep	16517403	16517403	16517403
F17- Upgrade Inventori	Procedure UpgradelInventori	16517403	16517403	16517403
B4- Restock	procedure Restock; function NemuIndexInven taris	16517389	16517389	16517389

2. CHECKLIST HASIL RANCANGAN, IMPLEMENTASI, DAN TESTING

Tabel 2. Tabel Checklist Hasil Rancangan, Implementasi, dan Testing

Fitur	Desain	Implementasi	Testing
F1	v	v	v
F2	v	v	v
F3	v	v	v
F4	v	v	v
F5	v	v	v
F6	v	v	v
F7	v	v	v
F8	v	v	v
F9	v	v	v
F10	v	v	v
F11	v	v	v
F12	v	v	v
F13	v	v	v
F14	v	v	v
F15	v	v	v
F16	v	v	v
F17	v	v	v
B4	v	v	v

Ket:

V : Sudah selesai dikerjakan

X : Dikerjakan, tapi belum selesai

-: Tidak dikerjakan sama sekali

3. DESKRIPSI PERSOALAN

Penulis diminta untuk membuat suatu program yang dapat menjalankan semua proses yang dilakukan oleh Engi's Kitchen, sebuah restoran yang didirikan lulusan baru ITB, bernama Chef. Dengan program ini, Chef dapat melakukan apapun yang harus ia lakukan di restorannya. Chef akan mengadakan simulasi selama maksimal 10 hari. Simulasi dapat dilakukan lebih dari sekali, dan bisa saja simulasi dijalankan tidak secara urut.

Pertama, program harus dapat menerima sebuah file eksternal yang berisi data-data awal restoran, mulai dari modal awal, nomor simulasi, jumlah hari sudah hidup, kapasitas maksimum inventori, hingga total aksi yang telah dilakukan chef(berapa barang telah dibeli, dijual, dan berapa keuntungan keseluruhannya). Program juga harus dapat membaca sebuah file eksternal yang berisi list bahan-bahan mentah yang dapat dibeli di restoran, harga belinya, dan waktu kadaluarsanya. Setelah itu program juga harus bisa membaca sebuah file eksternal yang berisi daftar semua bahan olahan yang dapat dibuat dari bahan mentah, dengan maksimal 10 bahan mentah, dan tiap bahan mentah maksimal dipakai sekali. Setiap bahan olahan memiliki kadaluarsa 3 hari. Setelah itu, program juga harus dapat membaca sebuah file eksternal yang berisi daftar resep makanan yang dapat dibuat, dengan maksimal terdiri dari 20 bahan mentah maupun olahan, dan setiap bahan dipakai maksimal sekali. Harga jual resep harus diatas 12,5 persen dari total harga jual semua bahan pembentuknya. Selain itu, setelah simulasi dilakukan, program juga harus bisa menuliskan data terbaru pada sebuah file eksternal.

Bahan mentah dan bahan olahan bersama-sama ditaruh dalam sebuah inventori yang memiliki maksimal tampungan. Kapasitas maksimal ini dapat ditingkatkan sebesar 25 buah dengan cara membayar.

Chef memiliki energi, yang nilai maksimalnya 10. Energi chef akan berkurang sebanyak 1 tiap kali chef membeli bahan mentah, membuat bahan olahan, membuat makanan berdasar resep, dan menjual bahan olahan. Jika energi Chef sudah habis, Chef harus tidur. Jika Chef tidur, hari akan berganti, dan setiap makanan di inventori yang kadaluarsa di hari sebelumnya, harus dibuang. Setelah tidur, Chef memiliki energi kembali menjadi 10. Selain tidur, Chef juga dapat menambah energi dengan cara beristirahat. Tiap kali beristirahat, energi Chef akan bertambah sebanyak 1. Tapi dalam sehari, Chef maksimal beristirahat sebanyak 6 kali. Selain itu, Chef juga dapat makan, yang akan menambah energi Chef sebanyak 3 buah, tapi 1 hari Chef makan maksimal 3 kali.

Selain itu, program juga harus dapat menerima masukan resep baru, dan menampilkan resep jika diminta, baik secara terurut naik maupun turun(secara abjad). Program juga diminta untuk memiliki sistem *restock* otomatis, yaitu dalam beberapa hari sekali(lebih dari 1 hari), beberapa bahan umum seperti beras, minyak, dll. Akan ditambah(tentu saja beli) secara otomatis.

Program akan berhenti jika sudah mencapai hari ke-10. Simulasi dapat dimulai maupun diberhentikan kapan pun, sehingga program harus menyimpan semua file secara utuh.

4. DESAIN COMMAND

----- WELCOME TO ENGI`S KITCHEN -----

----- Sebuah Simulasi Restoran Kekinian -----

> Load

Loading.....

Success

> startSimulasi

Simulasi yang tersedia :

'1. Stop Simulasi

2. Beli Bahan

3. Olah Bahan

4. Jual Olahan

5. Jual Resep

6. Makan

7. Istirahat

8. Tidur

9. Lihat Statistik

10. Lihat Inventori

11. Lihat Resep

12. Cari Resep

13. Tambah Resep

14. Upgrade Inventori

Silahkan pilih nomor simulasi yang diinginkan

>>1

Tanggal Mulai :

Jumlah Energi :

Simulasi Hari ke :

Kapasitas Inventori :

Jumlah Uang :

Jumlah Makan :

Jumlah Istirahat :

Jumlah Bahan Mentah Dibeli :

Jumlah Bahan Olahan Dibuat :

Jumlah Bahan Olahan Dijual :

Jumlah Resep yang Terjual :

Total Pemasukan :

Total Pengeluaran :

Total Uang yang Didapatkan :

>exit

Saving.....

Success

5. DESAIN KAMUS DATA

Dalam membuat program yang memenuhi persyaratan, kami membuat struktur data seperti di bawah ini:

.

1. Tipe Bentuk **Inventori**, digunakan untuk menyimpan data bahan-bahan di inventori. Tipe ini mempunyai beberapa sub-tipe:
 - i. **Nama** : Berbentuk array of string, yaitu berisi nama bahan makanan.
 - ii. **Tanggal**: Berbentuk array of date, yaitu berisi tanggal pembelian dengan format dd/yy/mm.
 - iii. **Banyak**: Berbentuk integer, menyatakan banyak bahan yang dibeli di suatu hari.
 - iv. **Neff**: Berbentuk integer, menyimpan banyak kolom efektif.
 - v. **Total**: Berbentuk integer, digunakan untuk menyimpan banyak total makanan di inventori(harus \leq NMax).
2. Tipe bentuk **Avatar**, digunakan untuk menyimpan data-data yang berhubungan dengan Chef. Terdiri dari:
 - i. **Energi**: Bertipe shortint, Berisi energi chef saat ini.
 - ii. **Hari**: Bertipe shortint, berisi berapa banyak hari telah dilalui Chef.

- lii. **Uang**: Bertipe longint, berisi banyak uang yang dimiliki Chef saat ini.
- iv. **BanyakMakan**: Bertipe shortint, berisi berapa kali Chef sudah makan hari ini.
- v. **BanyakIstirahat**: Bertipe shortint, berisi banyak kali Chef sudah istirahat hari ini.
- vi. **SudahTidur**: Bertipe boolean, berisi apakah Chef barusan tidur atau tidak(True jika barusan, dan No. Jika tidak)

3. Tipe bentukan **Bahan Mentah**, digunakan untuk menyimpan data bahan mentah yang dapat dibeli di supermarket. Terdiri dari:

- i. **Nama**: Bertipe array of string, yaitu menyimpan nama bahan mentah.
- ii. **Harga**: Bertipe array of longint, yaitu menyimpan harga.
- iii. **Kadaluarsa**: Bertipe array of shortint, yaitu menyimpan kadaluarsa.
- iv. **Neff**: Bertipe integer, yang menyimpan banyak bahan tercatat di tipe ini.

4. Tipe bentukan **Bahan Olahan**, digunakan untuk menyimpan data bahan olahan yang dapat dibuat. Terdiri dari:

- i. **Nama** : Berbentuk array of string, yaitu menyimpan nama bahan.
- ii. **JumlahBahan** : Berbentuk array of shortint, yaitu menyimpan banyak bahan yang diperlukan.
- iii. **Bahan**: Berebentuk matriks of string, yaitu menyimpan bahan apa saja yang diperlukan untuk membuat bahan olahan.
- iv. **Neff**: Bertipe shortint, digunakan untuk menyimpan banyak array terisi.

5. Tipe bentukan **BahanResep**, digunakan untuk menyimpan data bahan resep yang dapat dibuat. Terdiri dari:

- i. **Nama**: Berbentuk array of string, digunakan untuk menyimpan nama resep.
- ii. **BanyakBahan**: Berbentuk array of shortint, digunakan untuk menyimpan banyak bahan diperlukan untuk membuat.
- iii. **Bahan**: Berbentuk matriks of string, digunakan untuk menyimpan bahan yang diperlukan untuk membuat resep.

- iv. **Neff**: Berbentuk integer, digunakan untuk menyimpan banyak resep yang diketahui.
-
- 6. Tipe bentukan **TotalAksi**, digunakan untuk menyimpan total aksi yang dilakukan chef sejak hari pertama. Terdiri dari:
 - i. **MentahDibeli**: Bertipe integer, menyimpan banyak mentah telah dibeli.
 - ii. **OlahanDibuat**: Bertipe integer, menyimpan banyak olahan telah dibuat.
 - iii. **OlahanDijual**: Bertipe integer, menyimpan banyak olahan telah dijual.
 - iv. **ResepDijual**: Bertipe integer, menyimpan banyak resep telah dijual.
 - v. **Pengeluaran**: Bertipe longint, digunakan untuk menyimpan total pengeluaran.
 - vi. **Pendapatan**: Bertipe longint, digunakan untuk menyimpan total pendapatan(= pemasukan-pengeluaran).
 - vii. **Pemasukan**: Bertipe longint, digunakan untuk menyimpan total pemasukan.
-
- 7. Tipe bentukan **date**, untuk menyimpan suatu tanggal dalam format d/m/y. Terdiri dari:
 - i. **D**: Bertipe integer, hari (1-31).
 - ii. **M**: Bertipe integer, bulan (1-12).
 - iii. **Y**: Bertipe integer, tahun .
-
- 8. Tipe bentukan **FileSimulasi**, digunakan untuk menyimpan data simulasi(umu). Terdiri dari:
 - i. **No**: Bertipe shortint, berisi kode simulasi.
 - ii. **TanggalMulai**: Bertipe date, berisi tanggal simulasi dimulai.
 - iii. **KapasitasInv**: Bertipe integer, berisi maksimal bahan di inventori.
-
- 9. Tipe bentukan **invBahanMentah**, berisi inventori yang hanya menyimpan bahan mentah.
 - i. **Nama**: Bertipe array of string, berisi nama bahan.
 - ii. **Banyak**: Bertipe array of integer.

- iii. **Tgl**: Bertipe array of date, berisi tanggal pembelian bahan.
- iv. **Neff**: Bertipe integer, berisi indeks efektif(yang terpakai).

10. 10.Tipe bentukan **invBahanOlahan**, berisi inventori yang hanya menyimpan bahan olahan.

- i. **Nama**: Bertipe array of string, berisi nama bahan.
- li. **Banyak**: Bertipe array of integer.
- lii. **Tgl**: Bertipe array of date, berisi tanggal pembelian bahan.
- iv. . **Neff**: Bertipe integer, berisi indeks efektif(yang terpakai).

Dalam notasi algoritmik, dituliskan sebagai berikut:

Type inv:

< Nama : array [1..100] of string,

Tanggal: array[1..100] of date,

Banyak: array [1..10,1..100] of integer,

Neff: integer,

Total: integer >

Type avatar:

< Energi: integer,

Hari: integer,

Uang: integer,

CountMakan: integer ,

CountIstirahat: integer ,

SudahTidur: boolean>

Type BahanMentah:

< Nama: array [1..100] of string,

Harga: array [1..100] of integer,

Kadaluarsa: array [1..100] of integer,

Neff: integer >

Type BahanOlahan:

< Nama: array[1..100] of string,

JumlahBahan: array [1..100] of integer,

Bahan:array [1..100,1..10] of string,

Neff: integer >

Type BahanResep:

< Nama: array [1..100] of string,

HargaJual: array [1..100] of integer,

BanyakBahan: array[1..100] of integer.

Bahan: array[1..100,1..20] of string,

Neff: integer >

Type TotalAksi:

< MentahDibeli: integer.

OlahanDibuat: integer,

OlahanDijual:integer,

ResepDijual: integer,

Pengeluaran: Integer,

Pendapatan: integer >

Type date:

< D: integer,

M: integer,

Y: integer >

Type FileSimulasi:

< No : integer,

TanggalMulai: date,

KapasitasInv : integer >

Type invBahanMentah:

<Nama: array of string,

Banyak:array of integer,

Tanggal:array of date,

Neff:integer>

Type invBahanOlahan:

<Nama: array of string,

Banyak:array of integer,

Tanggal:array of date,

Neff:integer>

6. DESAIN DEKOMPOSISI PERSOALAN

Program Utama;

{Program yang menjalankan simulasi Engi's Kitchen}

Uses Tipe, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12

F13, F14, F15, F16, F17, B4;

{Kamus}

Resep: BahanResep {Menyimpan semua resep yang mungkin}

namaResep : string; {nama resep yang akan dicari dengan prosedur cariResep}

inv : inventori

mentah : bahanMentah {Menyimpan bahan mentah yang bisa dibeli}

olahan : bahanOlahan

invmentah : invbahanMentah {Menyimpan inventori mentah saja}

involahan : invbahanOlahan {Menyimpan inventori olahan saja}

ava : avatar {Menyimpan jumlah energi, dan atribut Chef lainnya}

command, simulasi : string {Menyimpan masukan user}

total : totalAksi {Menyimpan total aksi sejak awal}

stat : fileSimulasi

{Algoritma}

Output ('----- WELCOME TO ENGI'S KITCHEN -----')

Output('----- Sebuah Simulasi Restoran Kekinian -----')

Output('>'); Input(command); {Menerima masukan dari user apa yang akan dilakukan}

Repeat

Case command of

'load' : load(ava, inv, stat, mentah, olahan, invmentah, involahan, resep, total) {Memanggil prosedur load dari F1.}

Output('>'); Input(command) {Terima apa yang akan dilakukan selanjutnya}

'exit' : keluar(ava, stat, mentah, olahan, invmentah, involahan, resep, total) {Memanggil prosedur keluar dari F2}

Command <- 'Done' {Memasang Command='Done', sehingga akan keluar dari loop}

'startSimulasi' :

NamaResep <- " {Inisialisasi untuk dicari}

Simulasi <-"

startSimulasi(command, simulasi, ava, stat, inv, invmentah, involahan, resep, namaResep, total)

{Memanggil prosedur StartSimulasi dari F3 yang akan melanjutkan simulasi}

Else : output('>Input salah') {Keluarkan pesan kesalahan}

Output('Command yang tersedia: load,StartSimulasi, dan Exit')

Output('>') ; Input(command) {Baca lagi command}

Until (command='done');

End.

7. SPESIFIKASI FUNGSI

Tabel 3. Tabel Spesifikasi Tiap Fitur Dalam Unit

Unit	Fitur/ Fungsi	Spesifikasi
TypeDasar	<ol style="list-style-type: none"> 1. <u>Function</u> TambahTanggal(n: <u>integer</u>, x: date) -> date 2. <u>Function</u> PembandingTanggal(T1, T2: date) -> <u>boolean</u> 	<p><u>1.Function</u> TambahTanggal(n: <u>integer</u>, x: date) -> date { Fungsi yang menerima masukan bilangan bulat n, dan tanggal x, dan menghasilkan tanggal n hari setelah x}</p> <p><u>2.Function</u> PembandingTanggal(T1, T2: date) -> <u>boolean</u> { Fungsi yang menerima 2 tanggal, akan menghasilkan True jika tanggal pertama lebih besar dari tanggal kedua, dan False jika tanggal sama atau lebih kecil.}</p>

F1	<u>procedure</u> load (<u>output</u> tinv : inv; <u>output</u> tavatar:avatar; <u>output</u> tBahanMentah:BahanMentah; <u>output</u> tBahanOlahan : BahanOlahan; <u>output</u> tBahanResep:BahanResep; <u>output</u> tTotalAksi: TotalAksi; <u>output</u> tFileSimulasi :FileSimulasi);	<u>procedure</u> load (<u>output</u> tinv : inv; <u>output</u> tavatar:avatar; <u>output</u> tBahanMentah:BahanMentah; <u>output</u> tBahanOlahan : BahanOlahan; <u>output</u> tBahanResep:BahanResep; <u>output</u> tTotalAksi: TotalAksi; <u>output</u> tFileSimulasi :FileSimulasi); { Prosedur yang membaca semua file eksternal(resep, inventori,dsb) dan menyimpannya dalam sebuah array dalam program}
F2	<u>procedure</u> keluar (input tinv : inv;input tavatar:avatar;input tBahanMentah:BahanMentah;input tBahanOlahan : BahanOlahan;input tBahanResep:BahanResep;input tTotalAksi: TotalAksi;input tFileSimulasi :FileSimulasi);	<u>procedure</u> keluar (input tinv : inv;input tavatar:avatar;input tBahanMentah:BahanMentah;input tBahanOlahan : BahanOlahan;input tBahanResep:BahanResep;input tTotalAksi: TotalAksi;input tFileSimulasi :FileSimulasi); {Prosedur yang menuliskan semua data simulasi ke file eksternal.}
F3		
ProsedurF4F5F6	1. <u>function</u> CekAdaGak(<u>input</u> x: <u>string</u> ; <u>input</u> T: BahanMentah)-> <u>boolean</u> ; 2. <u>function</u> NemulIndex(<u>input</u> x: <u>string</u> ; <u>input</u> T: BahanMentah): <u>integer</u> ; 3. <u>function</u> NemulIndexInventaris(<u>input</u> x: <u>string</u> ; <u>input</u> T: avatar)-> <u>integer</u> ;	

	<p><u>4.function</u></p> <p>NemulIndexOlahan(input x: <u>string</u>; <u>input</u> T: BahanOlahan)-> <u>integer</u>;</p> <p><u>5.function</u></p> <p>CekPersediaanInventaris(<u>inp</u> <u>ut</u> s: <u>string</u>; <u>input</u> X: avatar)- > <u>integer</u>;</p> <p><u>6.function</u> CekBahanOlah(input s: <u>string</u>; input T: BahanOlahan; <u>input</u> X: avatar;<u>input</u> day: <u>integer</u>)-> <u>boolean</u>;</p> <p><u>7.procedure</u></p> <p>TulisBahanKurang(input s: <u>string</u>; <u>input</u> T: BahanOlahan; <u>input</u> X: avatar; <u>input</u> day:<u>integer</u>);</p>	
F4	<p><u>procedure stopSimulasi(ava</u> <u>: avatar; total : totalAksi ;</u> <u>stat : FileSimulasi);</u></p>	<p><u>procedure stopSimulasi(ava</u> <u>: avatar; total : totalAksi ;</u> <u>stat : FileSimulasi);</u> {<u>procedure yang digunakan</u> <u>untuk menyetop simulasi</u> <u>yang sedang berlangsung</u>}</p>
F5	<p><u>procedure</u></p> <p>BeliBahan(<u>input/output</u> X: BahanMentah; <u>input/output</u> Money: avatar; <u>output</u> Total: TotalAksi);</p>	<p><u>procedure</u></p> <p>BeliBahan(<u>input/output</u> X: BahanMentah; <u>input/output</u> Money: avatar; <u>output</u> Total: TotalAksi); {<u>procedure untuk membeli</u> <u>bahan yang diinginkan dari</u> <u>input dan mengecek apakah</u> <u>bahannya ada</u>}</p>

F6	<u>procedure</u> BeliOlah(<u>input/output</u> X: BahanOlahan; <u>input/output</u> chef: avatar; <u>input</u> n: shortint; <u>input/output</u> Total: TotalAksi);	<u>procedure</u> BeliOlah(<u>input/output</u> X: BahanOlahan; <u>input/output</u> chef: avatar; <u>input</u> n: shortint; <u>input/output</u> Total: TotalAksi); {procedure untuk membeli bahan olahan dan mengecek ada tidaknya bahan}
F7	procedure JualOlahan(var A:inv; var B:avatar;var C:TotalAksi);;	procedure JualOlahan(var A:inv; var B:avatar;var C:TotalAksi); {procedure untuk menjual bahan olahan, dengan cara mengecek apakah bahan penyusunnya tersedia atau tidak serta procedure ini mengurangi energi sebanyak satu, menambah pendapatan, dan mengurangi inventori}
F8	procedure jualResep(A:BahanResep; var B:inv;var Total: TotalAksi; var C:avatar);	procedure jualResep(A:BahanResep; var B:inv;var Total: TotalAksi; var C:avatar); {Procedure untuk menjual dan mengolah makanan dari resep. Program mengecek apakah bahan penyusunnya tersedia atau tidak serta program ini mengurangi energi sebanyak satu, menambah pendapatan, dan mengurangi inventori}
F9	procedure makan (var ava : avatar);	procedure makan (var ava : avatar); {Procedure untuk menambah energi chef sebanyak 3 buah dengan batas 3 kali dalam sehari atau energi chef mencapai 10}
F10	<u>procedure</u> Istirahat (input/output a : avatar)	<u>procedure</u> Istirahat (input/output a : avatar)

		{Prosedur yang menambah satu jumlah energi avatar sampai dengan kondisi tertentu yaitu sudah istirahat sebanyak 6 kali atau jumlah energi telah mencapai 10}
F11	procedur Tidur (input/output a:avatar)	procedur Tidur (input/output a:avatar) {Prosedur untuk mengembalikan energi chef hingga maksimum (10) dan mengganti hari serta mengosongkan inventori untuk bahan-bahan yang kadaluarsa di hari yang baru}
F12	procedure TampilStat(output a:avatar; output b:TotalAksi)	procedure TampilStat(output a:avatar; output b:TotalAksi) {Prosedur untuk menampilkan statistik dari permainan yang dijalankan}
F13	procedure lihatInventori(var ava : avatar; var inv : inventori; var invmentah : invbahanMentah; var involahan : invbahanOlahan; var stat : fileSimulasi);	procedure lihatInventori(var ava : avatar; var inv : inventori; var invmentah : invbahanMentah; var involahan : invbahanOlahan; var stat : fileSimulasi); {Procedure untuk menampilkan inventori saat simulasi sedang berlangsung}
F14	procedure lihatResep(var ava : avatar; var resep : bahanResep; var stat : fileSimulasi);	procedure lihatResep(var ava : avatar; var resep : bahanResep; var stat : fileSimulasi); {procedure untuk menampilkan resep dengan baik}
F15	procedure cariResep(var ava : avatar; namaResep : string; var resep : bahanResep; var stat : fileSimulasi);	procedure cariResep(var ava : avatar; namaResep : string; var resep : bahanResep; var stat : fileSimulasi); {procedure untuk mencari resep yang diinginkan}

F16	<pre> procedure tambahResep(var ava : avatar; var mentah : bahanMentah; var olahan : bahanOlahan; var resep : bahanResep; var stat : fileSimulasi); </pre>	<pre> procedure tambahResep(var ava : avatar; var mentah : bahanMentah; var olahan : bahanOlahan; var resep : bahanResep; var stat : fileSimulasi); {procedure yang digunakan untuk menambah resep sesuai input} </pre>
F17	<pre> procedure upgradeInventori(var ava: avatar; var stat: fileSimulasi; var total: totalAksi); </pre>	<pre> procedure upgradeInventori(var ava: avatar; var stat: fileSimulasi; var total: totalAksi); {procedure yang dapat mengupgrade inventori sesuai masukkan si pengguna} </pre>
Pencarian	<pre> function cariResepKuy(A:BahanResep ; buat:string):integer; function nemulIndexBahan(A:inv;S:st ring):integer; procedure kurangStock(S:string;var A:inv); function cariBahan(S:string; A:inv; B:BahanResep):boolean; </pre>	
B4	<pre> 1.<u>function</u> NemulIndexInventaris(x: <u>string</u>;T: avatar)-> <u>integer</u>; 2.<u>procedure</u> Restock(<u>input/output</u> chef: avatar; <u>input/output</u> Total: TotalAksi); </pre>	

8. DATA HASIL PENGUJIAN

Tabel 4. Tabel Pengujian Program






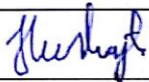
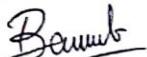
Fitur	Data Awal	Hasil Pengujian	Evaluasi
F1- load	File eksternal simulasi, resep, inventori mentah, dan inventori olahan.	Semua file terbaca	Semua file terbaca dengan baik
F2- exit	File eksternal belum berubah.	Semua informasi simulasi ditulis di file external.	Chef dapat keluar dari program tanpa kendala
F3- startSimulasi	Masih pada layar pertama menu program, hanya untuk load dan startSimulasi.	Semua pilihan program dapat ditampilkan.	Dapat memulai simulasi tanpa ada kendala
F4- stopSimulasi	Simulasi masih berjalan di menu pilihan utama.	Simulasi dapat dihentikan dengan menampilkan statistik terakhir(<i>recent</i>).	Dapat memberhentikan simulasi dengan baik
F5- beliBahan	Inventori mentah dan biasa.	User dapat membeli bahan dengan validasi yang sesuai.	Dapat membeli bahan dengan baik
F6- olahBahan	Inventori olahan dan biasa.	User dapat membuat olahan dengan validasi yang sesuai.	Dapat mengolah bahan dengan benar
F7- jualOlahan	Inventori Olahan, biasa, dan uang avatar.	User dapat menjual olahan dengan validasi yang sesuai.	Kegiatan menjual olahan dapat bekerja dengan baik
F8- jualResep	Inventori biasa, dan uang.	Resep terjual secara tepat dan inventori tepat.	Dapat menjual makanan sesuai resep dengan baik
F9- makan	Energi avatar dan BanyakMakan avatar.	Makan dapat meningkatkan energi,	Aktivitas makan dapat berjalan dengan baik

		dan maksimal dilakukan 3 kali sehari.	
F10- istirahat	Energi avatara dan BanyakTidur avatar.	Istirahat daoat meningkatkan energi, dan maksimal dilakukan 6 kali sehari.	Aktivitas istirahat dapat bekerja dengan baik
F11- tidur	Energi avatar, inventori, hari hidup, dan sudah tidur(boolean).	Tidur dapat mengganti hari, menghapus yang kadaluarsa, dan memenuhi energi.User tidak dapat tidur 2 kali berturut-turut.	Aktivitas tidur berjalan dengan baik
F12- lihatStatistik	File simulasi.	Semua statistik yang ingin ditunjukkan terlihat.	Statistik ditunjukkan dengan baik.
F13- lihatInventori	Inventori biasa.	Inventori dapat ditampilkan urut dari A-Z atau sebaliknya.	Dapat menampilkan inventori dengan baik
F14- lihatResep	Bahan Resep.	Resep dapat dilihat beserta harga dan bahannya.	Dapat melihat resep dengan baik
F15- cariResep	Bahan Resep.	Resep dapat dicari dengan tepat.	Dapat mencari resep dengan baik
F16- tambahResep	Bahan resep.	Resep ditambahkan dengan benar.	Kegiatan menambah resep berjalan tanpa kendala
F17- upgradeInventori	File Simulasi KapasitasInv.	Inventori ditambahkan dengan benar dan uang berkurang secara tepat.	Dapat mengupgrade inventori
B4- restock	Inventori umum, mentah.	Restock dilakukan 3 kali sehari untuk bahan mentah beras dan air, maksimal masing-masing 10 buah(beras lebih diutamakan).	Dapat merestock ulang bahan yang baru

9. LAMPIRAN

**Form Asistensi Tugas Besar
IF1210/Dasar Pemrograman
Sem. 2 2017/2018**

Nomor Asistensi : 1
No. Kelompok/Kelas : 11 / K-04
Tanggal : Rabu, 11 April 2018

Anggota kelompok	NIM / Nama	Tanda Tangan Kehadiran
1	16517368 / Mgs.Muhammad Riandi Ramadhan	1 
2	16517375 / Haris Salman Al-Ghifary	2 
3	16517382 / Muhammad Fatkhi	3 
4	16517389 / Christofel Rio Goenawan	4 
5	16517396 / Habli Yahdi	5 
6	16517403 / Nurhayati Marqoh Sirfeffa	6 
Asisten pembimbing	NIM / Nama	Tanda Tangan Kehadiran
	13514047 / Bervianto Leo P.	

Catatan Asistensi:

- Input file eksternal kemungkinan disediakan sendiri
- Pemisahan data eksternal boleh tidak memfari garis
- Perlu fitur sorting untuk mengurutkan nama ~~dan kecil ke atas atau~~ menurun atau membesar secara alfabet pada fitur lihat file dan lihat Inventori
- Perlu dibuat suatu fungsi untuk memproses huruf input pada fitur pencarian dengan file eksternal sehingga tidak perlu mengkhawatirkan besar kecil huruf
- Format tanggal diperbolehkan tidak memfari ' / ', ~~ke~~
- Setiap fitur input dari user, sebaiknya ada validasi
- Untuk fitur reset, bahan ditentukan sendiri

Foto 1

Form Asistensi Tugas Besar
IF1210/Dasar Pemrograman
Sem. 2 2017/2018

Tindak Lanjut




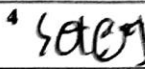



- File eksternal dibuat sendiri, namun jenis dan detail dari file eksternal menjadi sama
- Penempatan data yang menjadi garis diganti dengan spasi, sehingga tipe karakter terdapat masalah
- Setelah diteliti prosedur First Recp atau First Inventory, user akan ditanyakan apakah data atau format tersebut akan mengedit
- Pada titik pencarian, dibuat sebuah fungsi untuk mengubah input nama menjadi lower case atau uppercase atau ~~lower~~ proporsional agar sesuai dengan nama di file eksternal
- Penempatan tanggal, bulan, dan tahun menggunakan spasi tanpa ada '/'
- Pada titik input, dibuat validasi untuk mengecek input benar atau salah
- Bahan-bahan yang akan track, dituliskan sendiri

Foto 2

Form Asistensi Tugas Besar
IF1210/Dasar Pemrograman
Sem. 2 2017/2018

Nomor Asistensi : 1
 No. Kelompok/Kelas : 11 / K-04
 Tanggal : 19 April 2018

Anggota kelompok

	NIM / Nama	Tanda Tangan Kehadiran
1	16517368 / Mgs. Muhammad Riandi Ramadhan	1 
2	16517375 / Haris Salman Al-Ghifary	2 
3	16517382 / Muhammad Fatkhi	3 
4	16517389 / Christofel Rio Goenawan	4 
5	16517396 / Habli Yahdi	5 
6	16517403 / Nurhayati Marqoh Sirfefa	6 
Asisten pembimbing	NIM / Nama 13514047 / Bervianto Leo P.	Tanda Tangan Kehadiran 

Catatan Asistensi:

Rangkuman Diskusi

- Untuk laporan bagian prosedur hanya ditulis spek sama header (I.S and F.S)
- Boleh menambah fungsi / procedure dalam tugas utama
- Dekomposisi program, memisah-misah program dalam unit
- Jangan telat mengumpulkan di olympia, kalau telat kemungkinan diberi nilai 0.
- Fitur B-1 pada bagian bonus dapat dilakukan di F1

Foto 3

Form Asistensi Tugas Besar
IF1210/Dasar Pemrograman
Sem. 2 2017/2018

Tindak Lanjut

- Laporan segera diselesaikan secepatnya
- jangan merunda-runda untuk menggabungkan program,
gak usah nunggu semua jadi

Foto 4

