

Engi's Kitchen

Versi 1 April 2018

Deskripsi Persoalan

Chef baru saja lulus dari ITB. Dengan segala ilmu yang ia miliki, ia ingin membuka sebuah tempat makan. Karena ia seorang *engineer*, nama tempat makan yang akan ia buka adalah “Engi's Kitchen”. Sebelum membuka Engi's Kitchen, Chef ingin mensimulasikan terlebih dahulu bagaimana menjalankan restoran untuk beberapa hari. Untuk itu Chef meminta kalian untuk membuat sebuah program yang dapat mensimulasikan Engi's Kitchen selama maksimum 10 hari.

Saat pertama kali dibuka, Engi's Kitchen dibekali sejumlah modal tertentu sebagai pendapatan awal. Seperti restoran pada umumnya, modal tersebut digunakan untuk membeli bahan mentah dari supermarket. Untuk setiap bahan mentah dicatat harga satuan untuk tiap *piece*-nya dan durasi kadaluarsanya (dalam hari). Bahan mentah yang dibeli dari supermarket disimpan dalam inventori. Didata jumlah unit yang dibeli dan tanggal pembelian untuk setiap bahan mentah. Tanggal kadaluarsa suatu bahan mentah yang disimpan dalam inventori dihitung berdasarkan tanggal pembelian.

Bahan mentah dapat diolah untuk menghasilkan bahan olahan yang memiliki harga yang lebih tinggi. Bahan olahan terbuat dari 1 jenis bahan mentah atau lebih (maksimum dibuat dari 10 jenis bahan mentah). Diasumsikan dibutuhkan hanya 1 *piece* untuk tiap bahan mentah. Harga jual bahan olahan lebih tinggi atau minimum sama dengan total penjumlahan harga tiap bahan mentah yang menyusunnya. Kadaluarsa semua bahan olahan adalah sama, yaitu 3 hari. Tanggal kadaluarsa bahan olahan dihitung berdasarkan tanggal dibuatnya bahan olahan tersebut. Pembuatan bahan olahan mengurangi jumlah bahan mentah yang tersimpan di inventori. Bahan olahan ini juga dijual di restoran ini untuk menambah keuntungan.

Bahan mentah dan bahan olahan disimpan dalam 1 buah inventori. Inventori restoran memiliki batas maksimum bahan yang dapat disimpan. Jika Chef sudah merasa bahwa inventornya tidak cukup, Chef dapat melakukan *upgrade* inventori dengan harga tertentu untuk menambah ruang maksimum yang dapat disimpan inventori.

Makanan/minuman di restoran dibuat berdasarkan resep tertentu. Untuk setiap resep dicatat daftar bahan yang dibutuhkan untuk membuat 1 porsi hidangan dari resep tersebut (maksimum 20 bahan untuk menyusun resep). Bahan penyusun resep dapat berupa bahan mentah maupun bahan olahan. Diasumsikan dibutuhkan hanya 1 *piece* bahan mentah atau bahan olahan untuk membuat 1 porsi hidangan dari suatu resep. Untuk tiap resep, disimpan harga hidangan dari resep tersebut. Chef memutuskan setiap hidangan memiliki harga jual minimum 12,5% lebih tinggi dari total seluruh bahan penyusunnya. Setiap pembuatan masakan dari suatu resep, mengurangi jumlah bahan mentah dan bahan olahan yang tersimpan di inventori. Chef dapat menambahkan resep baru yang dibuat dari bahan yang ada.

Dalam simulasi ini, Chef memiliki sejumlah energi untuk melaksanakan aktivitasnya. Didefinisikan sejumlah aksi yang dapat mengurangi atau menambah energi Chef. Aksi-aksi yang dapat mengurangi energi Chef, masing-masing sebanyak 1 buah, adalah: 1) membeli bahan mentah ke supermarket; 2) membuat bahan olahan; 3) membuat masakan berdasarkan resep; 4) menjual bahan olahan (lihat keterangan masing-masing pada bagian Fitur).

Chef juga dapat melakukan aksi yang dapat menambah energi, yaitu: 1) Makan: akan menambah energi sebanyak 3 buah, namun maksimum dalam 1 hari hanya boleh 3x makan; 2) Istirahat: akan menambah energi sebanyak 1 buah, maksimum dalam 1 hari hanya boleh istirahat sebanyak 6x; 3) Tidur: untuk memulihkan energi menjadi 10 buah. Jika energi habis, aksi yang dapat dilakukan oleh Chef hanya

tidur. Jika Chef tidur, terjadi pergantian hari. Jika terjadi pergantian hari, secara otomatis bahan-bahan mentah dan olahan yang kadaluarsa pada hari yang baru, dihapus dari inventori.

Simulasi berjalannya restoran dimulai dengan sejumlah modal dan keadaan inventori tertentu (dengan bahan mentah dan bahan olahan yang tersedia). Pengguna menjalankan simulasi dengan melakukan aksi-aksi yang dapat mengurangi/menambah energi, persediaan bahan di inventori, dan pendapatan. Simulasi dihentikan jika jumlah hari sudah mencapai 10 hari. Simulasi dapat dihentikan pada hari tertentu atau dimulai pada hari tertentu.

Informasi lebih detail dapat dilihat pada bagian Fitur.

Struktur Data File Eksternal

Program Engi's Kitchen perlu membaca beberapa data dari file eksternal untuk beroperasi dengan lancar.

1. File Bahan Mentah

Berisi daftar bahan mentah yang dibeli dari supermarket dan disimpan dalam inventori. Bahan mentah dapat digunakan dalam bahan olahan ataupun dalam resep. Berikut data yang disimpan:

Nama Bahan Mentah | Harga Satuan | Durasi Kadaluarsa

Keterangan:

- Nama Bahan Mentah: nama bahan mentah diasumsikan unik dan tidak ada yang sama dengan nama bahan olahan.
- Harga Satuan: harga per piece untuk tiap bahan.
- Durasi Kadaluarsa: durasi kadaluarsa dalam satuan hari untuk bahan mentah/

Contoh:

```
Daging Sapi | 5000 | 10
Kangkung | 1000 | 10
Lada | 200 | 100
Bawang Putih | 500 | 30
Air | 500 | 100
Es Krim | 2000 | 10
Kecap | 400 | 300
Beras | 500 | 100
```

2. File Bahan Olahan

Berisi daftar bahan olahan yang dapat dibuat berdasarkan bahan mentah dan dapat digunakan dalam resep hidangan. Sebagai catatan: semua bahan olahan kadaluarsa dalam 3 hari sejak dibuat. Berikut data yang disimpan:

Nama Bahan Olahan | Harga Jual | N | Bahan-1 | Bahan-2 | Bahan-3 | ... | Bahan-N

Keterangan:

- Nama Bahan Olahan: nama bahan olahan yang dihasilkan restoran. Diasumsikan nama bahan olahan unik dan tidak ada yang sama dengan nama bahan mentah.
- Harga Jual: harga jual per *piece* bahan olahan. Harga jual bahan olahan lebih tinggi atau minimum sama dengan total penjumlahan harga tiap bahan mentah yang menyusunnya.
- N: banyaknya bahan mentah yang menyusun bahan olahan (minimum 1 dan maksimum 10).
- Bahan-1, Bahan-2, ..., Bahan-N: daftar nama bahan mentah yang dibutuhkan untuk membuat bahan olahan. Diasumsikan hanya dibutuhkan 1 *piece* bahan mentah untuk menyusun 1 *piece* bahan olahan.

Contoh:

```
Nasi Putih | 1200 | 2 | Beras | Air
```

Nasi Goreng | 2000 | 5 | Beras | Air | Bawang Putih | Lada | Kecap

3. File Inventori Bahan Mentah

Berisi data bahan mentah yang tersimpan dalam inventori. Berikut data yang disimpan:

Nama Bahan Mentah | Tanggal Beli | Jumlah

Keterangan:

- Nama Bahan Mentah: nama bahan mentah. Diasumsikan nama bahan mentah sesuai dengan nama bahan mentah pada daftar bahan mentah.
- Tanggal Beli: tanggal pembelian bahan mentah dalam format DD/MM/YYYY (hari/bulan/tahun).
- Jumlah: banyaknya bahan mentah dalam inventori.

Contoh:

Beras | 12/1/2018 | 23
 Beras | 23/1/2018 | 10
 Bawang Putih | 4/12/2017 | 100
 Air | 10/1/2018 | 34

4. File Inventori Bahan Olahan

Berisi data bahan mentah yang tersimpan dalam inventori. Berikut data yang disimpan:

Nama Bahan Olahan | Tanggal Buat | Jumlah

Keterangan:

- Nama Bahan Mentah: nama bahan olahan. Diasumsikan nama bahan olahan sesuai dengan nama bahan olahan pada daftar bahan olahan.
- Tanggal Buat: tanggal pembuatan bahan olahan dalam format DD/MM/YYYY (hari/bulan/tahun).
- Jumlah: banyaknya bahan olahan dalam inventori.

Contoh:

Sosis | 24/2/2018 | 100
 Nasi Goreng | 10/2/2018 | 34

5. File Resep

Berisi daftar resep masakan yang ditawarkan oleh restoran. Berikut data yang disimpan:

Nama Resep | Harga Jual | N | Bahan-1 | Bahan-2 | Bahan-3 | ... | Bahan-N

Catatan:

- Nama Resep: nama resep masakan. Diasumsikan nama resep unik.
- Harga Jual: harga jual per porsi hidangan berdasarkan resep. Harga jual diasumsikan minimum 12,5% lebih tinggi dari semua bahan penyusunnya.
- N: banyaknya bahan yang digunakan untuk menyusun resep (minimum 2 dan maksimum 20).
- Bahan-1, Bahan-2, ..., Bahan-N: daftar nama bahan mentah dan olahan yang dibutuhkan untuk membuat masakan berdasarkan resep. Diasumsikan hanya dibutuhkan 1 *piece* bahan mentah atau bahan olahan untuk menyusun 1 porsi hidangan.

Contoh:

Nasi Goreng Sapi | 15000 | 2 | Nasi Goreng | Daging Sapi
 Soda Float | 10000 | 2 | Es Krim | Soda
 Nasi Sate Ayam | 15000 | 3 | Nasi Putih | Daging Sapi | Kecap
 Roti Lapis | 10000 | 6 | Roti | Selada | Tomat | Telur | Garam | Lada

6. File Simulasi

Berisi daftar simulasi yang dapat dijalankan dalam program berdasarkan data bahan mentah, bahan olahan, resep, dan inventori. Berikut data yang disimpan:

Nomor Simulasi | Jumlah Hari Hidup | Jumlah Energi | Kapasitas Maksimum Inventori |
 Total Bahan Mentah Dibeli | Total Bahan Olahan Dibuat | Total Bahan Olahan Dijual |
 Total Resep Dijual | Total Pemasukan | Total Pengeluaran | Total Pendapatan

Keterangan:

- Nomor Simulasi: nomor simulasi yang sedang dijalankan.
- Tanggal: tanggal awal simulasi dalam format DD/MM/YYYY.
- Jumlah Hari Hidup: jumlah hari yang sudah dilewati dalam simulasi. Nilai minimum adalah 0 (simulasi belum dimulai sama sekali) dan maksimum 10 (simulasi berakhir).
- Jumlah Energi: jumlah energi milik pengguna yang tersisa.
- Kapasitas Maksimum Inventori: jumlah bahan yang mungkin disimpan pada inventori pengguna dalam waktu bersamaan.
- Total Bahan Mentah Dibeli: banyaknya total bahan mentah dibeli selama simulasi (dari jenis apa pun).
- Total Bahan Olahan Dibuat: banyaknya total bahan olahan dibuat selama simulasi (dari jenis apa pun)
- Total Bahan Olahan Dijual: banyaknya total bahan olahan dijual selama simulasi (dari jenis apa pun)
- Total Resep Dijual: banyaknya total bahan olahan dijual selama simulasi (dari jenis apa pun).
- Total Pemasukan: jumlah pemasukan yang diterima oleh pengguna dari modal, penjualan hidangan dari resep dan penjualan bahan olahan.
- Total Pengeluaran: jumlah pengeluaran yang dihabiskan oleh pengguna dari pembelian bahan mentah dan upgrade inventori.
- Total Uang: jumlah total uang yang dimiliki oleh restoran, yaitu pemasukan dikurangi semua pengeluaran.

Contoh:

1 | 12/2/2018 | 2 | 9 | 20 | 4 | 5 | 3 | 5 | 20000 | 13500 | 6500

Fitur

Buatlah sebuah program dengan bahasa Pascal yang mengelola simulasi Engi's Kitchen dengan fitur-fitur sebagai berikut:

1. **F1-load:** membaca semua data dari file dan load ke dalam struktur data internal (array).
2. **F2-exit:** keluar dari program, dan menyimpan semua perubahan data ke file eksternal.
3. **F3-startSimulasi:** memulai suatu simulasi nomor tertentu dari daftar simulasi yang ada. Hari simulasi dimulai pada tanggal yang bersangkutan.

Perintah-perintah berikut ini hanya dapat dijalankan jika F3-startSimulasi sudah dijalankan dan berkaitan dengan nomor simulasi yang sedang berjalan.

4. **F4-stopSimulasi:** menghentikan simulasi tertentu yang sedang berjalan dan menampilkan statistik simulasi (minimum menampilkan apa yang ada dalam data simulasi yang bersangkutan, silakan menambahkan statistik lain – bonus).
5. **F5-beliBahan:** melakukan pembelian bahan mentah dari supermarket. Pembelian bahan mentah menambah inventori bahan mentah dan mengurangi energi Chef sebanyak 1 buah. Tidak bisa melakukan pembelian bahan apabila sudah mencapai maksimum kapasitas inventori dan jika jumlah uang yang dimiliki restoran tidak mencukupi untuk membeli bahan.
6. **F6-olahBahan:** melakukan aktivitas pengolahan bahan mentah menjadi bahan olahan. Pembuatan bahan olahan mengurangi inventori bahan mentah, menambah inventori bahan olahan, dan mengurangi energi Chef sebanyak 1 buah. Tidak bisa membuat bahan olahan jika bahan mentah

yang tersedia tidak mencukupi dan jika penambahan bahan olahan mencapai maksimum kapasitas inventori.

7. **F7-jualOlahan:** menjual bahan olahan. Hanya bahan olahan yang belum kadaluarsa yang dapat dijual. Kadaluarsa dihitung berdasarkan tanggal pada inventori dan tanggal simulasi. Setiap bahan olahan kadaluarsa 3 hari setelah dibuat. Penjualan bahan olahan mengurangi inventori, menambah pemasukan, dan mengurangi energi Chef sebanyak 1 buah.
8. **F8-jualResep:** mengolah dan menjual hidangan berdasarkan resep. Hidangan hanya bisa dibuat jika bahan mentah dan olahan penyusun resep tersedia di inventori. Pembuatan hidangan mengurangi inventori, menambah pemasukan, dan mengurangi energi Chef sebanyak 1 buah.
9. **F9-makan:** Chef makan. Aksi ini menambah energi Chef sebanyak 3 buah. Dalam 1 hari, Chef hanya dapat makan maksimum 3 kali. Energi chef bertambah maksimum hanya 10.
10. **F10-istirahat:** Chef istirahat. Aksi ini menambah energi Chef sebanyak 1 buah. Dalam 1 hari, Chef hanya dapat istirahat maksimum 6 kali. Energi chef bertambah maksimum hanya 10.
11. **F11-tidur:** Chef tidur. Aksi ini menyebabkan energi Chef bertambah sampai menjadi 10 dan hari berganti. Setelah melakukan tidur, pemain tidak dapat langsung melakukan tidur kembali. Tidur dapat dilakukan setelah melakukan minimal satu aksi lainnya. Pada saat pergantian hari, dilakukan **penghapusan otomatis bahan-bahan mentah dan olahan yang sudah kadaluarsa** pada inventori. Penghapusan bahan mentah/olahan ini menyebabkan berkurangnya inventori.
12. **F12-lihatStatistik:** menampilkan statistik simulasi (minimum menampilkan apa yang ada dalam data simulasi yang bersangkutan, silakan menambahkan statistik lain – bonus).

Perintah-perintah berikut ini dapat dijalankan kapan saja, termasuk jika simulasi sedang dijalankan. Aksi tidak mempengaruhi simulasi dan tidak mengurangi energi Chef.

13. **F13-lihatInventori:** menampilkan data daftar bahan mentah dan bahan olahan yang tersedia di inventori saat ini.
14. **F14-lihatResep:** melihat daftar resep yang tersedia, termasuk semua daftar bahan penyusunnya.
15. **F15-cariResep:** melakukan pencarian resep berdasarkan nama resep yang tersedia. Jika masukan nama resep ditemukan, maka program akan menampilkan nama resep dan nama bahan-bahan yang diperlukan dan harga jual resep tersebut. Jika tidak sesuai, maka program akan menampilkan pesan kesalahan.
16. **F16-tambahResep:** melakukan penambahan resep. Setiap resep dibuat dari minimum 2 buah bahan mentah/olahan yang ada (tidak boleh dari bahan yang tidak tersedia). Harga jual minimum adalah 12.5% lebih tinggi dari total harga bahan mentah dan olahan yang menyusunnya. Apabila terdapat desimal pada harga jual, dilakukan pembulatan ke atas (contoh: 1999.1 menjadi 2000).
17. **F17-upgradeInventori:** melakukan perluasan kapasitas inventori. Jumlah perluasan setiap kali *upgrade* dilakukan adalah 25. Sebagai contoh, jumlah inventori awal adalah 25 sehingga pada saat dilakukan *upgrade* kapasitas bertambah menjadi 50.

Catatan:

1. Pembagian fitur/fungsionalitas di atas tidak merepresentasikan dekomposisi modul, fungsi, dan prosedur yang sesungguhnya.
2. Semua masukan program, kecuali disebutkan secara khusus, tidak perlu dilakukan validasi.

Antarmuka dan Cara Memberikan *Command*

Program dibuat dengan antarmuka berbasis teks. Di awal, program menyediakan *prompt* kepada *user* dalam bentuk karakter ">", dan *user* memberikan perintah (*command*) kepada program dalam bentuk teks perintah tertentu. Saat simulasi, *prompt user* adalah ">>", untuk membedakan saat program beroperasi di luar simulai.

Berikut adalah beberapa contoh perintah dan hasil eksekusi dari program. Teks yang dengan karakter tebal adalah teks perintah atau *input* yang dimasukkan oleh *user*. Semua perintah dan *input* diasumsikan benar sehingga tidak perlu ada pengecekan. Anda harus mendefinisikan sendiri *command* yang contohnya tidak diberikan dalam dokumen spesifikasi ini. Tuliskan spesifikasinya dalam laporan Anda.

F1-loadFile

```
> load  
Loading file sukses.
```

F2-startSimulasi dan F2-beliBahan

```
> start 1  
Mulai simulasi 1  
>> beliBahan  
Nama bahan: Beras  
Kuantitas: 200  
Total harga: 100000  
Pembelian barang gagal, silahkan coba lagi.  
>> beliBahan  
Nama bahan: Beras  
Kuantitas : 2  
Total harga : 1000  
Pembelian sukses.
```

Bonus

Fitur berikut tidak harus diimplementasikan dalam program Anda, tetapi apabila diimplementasikan akan menambah nilai tugas anda maksimum 10% dari skala penilaian normal.

1. **B1-cekKesalahanLoading:** mengecek jika ada format file eksternal yang salah pada saat *loading*, atau pemberian perintah yang salah pada antarmuka dan memberikan *error message* (buatlah spesifikasinya secara lebih persis).
2. **B2-validasi:** menambahkan fitur untuk melakukan validasi terhadap berbagai hal yang membutuhkan pengecekan (buatlah spesifikasinya secara lebih persis).
3. **B3-unit:** program dibuat dengan memanfaatkan unit dalam Pascal. Struktur modul/unit yang dipakai harus dideskripsikan dalam laporan.
4. **B4-restock:** melakukan *restock* bahan mentah secara berkala setiap beberapa hari. Periode *restock* dibebaskan, tetapi harus lebih lama dari satu hari (buatlah spesifikasinya secara lebih persis).

Deliverables

1. *Source code* program harus dibuat sesuai standar yang diajarkan di kuliah, bersih (hanya mengandung bagian-bagian yang diperlukan), dan *well-commented*.
2. Laporan Tugas Besar dibuat dengan *template* sebagai berikut:
 - a. Halaman *Cover*, berisi minimum Kode dan Nama Kuliah, Nama Tugas, Nomor Kelas, Nomor Kelompok, NIM dan Nama Anggota Kelompok, Nama Sekolah (STEL), dan Perguruan Tinggi (ITB), dan Tahun (2018).
 - b. Daftar Isi.
 - c. Daftar Tabel (jika ada).
 - d. Daftar Gambar (jika ada).
 - e. Daftar pembagian kerja anggota kelompok, berdasarkan fitur yang dibuat, meliputi desain, implementasi/koding, dan testing.

Contoh:

Fitur	Implementasi *)	NIM Desainer **)	NIM Koder **)	NIM Tester **)
F1-load	Procedure Load	16517501/16517511	16517511	16517524
F2-exit	Procedure Exit; Procedure WriteFile;	16517501	16517524	16517511
F3-startSimulasi	Procedure StartSimulasi	16517511	16517501	16517501
....

*) bisa menjadi procedure/fungsi/bagian dari program utama dan bisa lebih dari 1

**) Bisa dikerjakan oleh lebih dari 1 orang

- f. Checklist hasil rancangan, implementasi dan testing setiap primitif.

Contoh Checklist:

Fitur	Desain	Implementasi	Testing
F1-load	V	V	X
F2-exit	V	V	V
F3-startSimulasi	V	X	-
....

Keterangan: V: sudah selesai dikerjakan, X: dikerjakan, tapi belum selesai, -: tidak dikerjakan sama sekali.

- g. Deskripsi persoalan: berisi penjelasan kembali (dengan “bahasa sendiri”) tentang persoalan yang akan diselesaikan.
- h. Desain *command* untuk setiap primitif (berisi: nama *command*, masukan, dan keluaran).
- i. Desain kamus data (dalam notasi algoritmik).
- j. Desain dekomposisi algoritmik dan fungsional program.
- k. Spesifikasi untuk tiap modul/prosedur/fungsi yang dibuat (dalam notasi algoritmik).
- l. Pengujian (testing), berisi: (1) data awal pengujian, (2) pengujian masing-masing fitur dan hasilnya, (3) evaluasi/diskusi hasil pengujian.
- m. Lampiran: Hasil scan form asistensi.