

# Spesifikasi Tugas Besar

## IF2110 Algoritma dan Struktur Data

### Deskripsi Game

Pak Santo adalah seorang mahasiswa informatika yang sangat senang memprogram. Selama kuliah, beliau sangat menikmati berbagai aktivitas belajar dan sangat berniat mengerjakan tugas. Namun, seiring berjalannya waktu, beliau lelah terhadap berbagai tugas yang diberikan karena tidak sesuai dengan minat asli beliau. Oleh sebab itu, pada tahun terakhir kuliahnya, beliau memutuskan untuk mengerjakan skripsi dengan seadanya dan fokus untuk lulus secepat mungkin. Setelah Pak Santo lulus, beliau pun memutuskan untuk membuka toko komputer saja karena sudah terlanjur kehilangan minat terhadap dunia pemrograman.

Game Santo Tycoon adalah permainan strategy yang mensimulasikan cerita Pak Santo diatas. Game ini dimainkan dengan memasukkan perintah melalui *command line interface* dengan *command* yang akan dijelaskan lebih detail di bawah. Game ini dibuat dengan **bahasa C**.

### Game Mechanics

#### 1. Mekanisme Permainan - Overall Description

- a. Permainan fokus pada seputar keseharian seorang perakit komputer.
- b. Setiap hari, pemain akan memeriksa daftar order untuk melihat apakah ada pemesanan komputer baru dari pelanggan.
- c. Setelah mendapatkan order baru, pemain akan membuat komputer dengan spesifikasi yang telah diberikan, jika tidak mempunyai komponen yang cukup, pemain dapat membeli di shop.
- d. Saat komputer pesanan telah selesai dibuat, maka pemain harus mengantarkan sendiri komputer tersebut.
- e. Permainan berjalan dengan basis per hari.
- f. Dalam setiap harinya setiap pemain dapat melakukan aktivitas tanpa batas sebelum menyatakan hari selesai.
- g. Saat hari berganti, toko pemain akan mendapatkan orderan baru.
- h. Hari berganti saat pemain memberikan command `END_DAY`.

#### 2. Kondisi Awal Permainan

- a. Pada saat permainan dimulai, program akan membaca file konfigurasi melalui file eksternal yang dibuat. Format data dibebaskan asalkan dapat memenuhi requirement data yang dibutuhkan.

- b. Pada awal permainan, pemain akan mulai dengan inventory kosong dan diberikan modal awal sejumlah tertentu.
- c. Antrian pesanan dari pemain sudah memiliki pesanan yang dapat diproses.

### 3. Main Menu

- a. Pada awal permainan, pemain dihadapkan pada tampilan main menu.
- b. Pada main menu terdapat 1 buah tombol yaitu untuk memulai permainan.
- c. (Bonus) Selain memulai permainan baru, pemain dapat *load* game yang sudah disimpan.

### 4. Peta

- a. Pada permainan ini, terdapat peta wakanda sebagai representasi kota tempat tinggal Pak Santo. Ukuran peta minimal 10 x 10 dan maksimal 20 x 30 (dibebaskan pada mahasiswa). Titik (i, j) merupakan sebuah petak pada baris ke-i dan kolom ke-j.
- b. Berikut adalah ilustrasi peta 10 x 15 yang ditampilkan kepada pemain

```

*****
*1      4      *
*              *
*          6     *
*      3        *
*              *
* S            *
*          7     *
* 2           *
*          5     *
* P            *
*****

```

Pada peta terdapat simbol pemain (P), base(B), shop(S) dan pelanggan(angka). Batasan peta ditandai dengan karakter bintang (\*). Petak tanpa bangunan cukup dituliskan sebagai spasi saja. Saat pemain berpindah ke lokasi lain, simbol asli dari lokasi tersebut diganti dengan P. Pada kasus diatas, pemain sedang berada pada base sehingga simbol B tidak tampak.

### 5. Komponen & Build

- a. Setiap komponen terdiri dari nama, harga dan juga kategori. Kategori komponen dapat dikelompokkan menjadi CPU, CPU Cooler, GPU, Motherboard, Memory, Storage, Case, PSU.

- b. Pada saat melakukan build, komponen yang dipasang harus sesuai dengan urutan berikut:
  - i. Motherboard
  - ii. CPU
  - iii. Memory
  - iv. CPU Cooler
  - v. Case
  - vi. GPU
  - vii. Storage
  - viii. PSU
- c. Sebelum dapat menambahkan komponen, pemain harus memulai sebuah build. Sebuah build yang sudah berjalan tidak dapat diganti dengan build lain.

## 6. Order

- a. Sebuah pesanan komputer memiliki properti sebagai berikut:
  - i. Pemesan  
Berisi informasi pelanggan yang membuat pesanan komputer tersebut. Informasi ini dapat berupa nomor pelanggan yang juga digunakan dalam representasi pada peta.
  - ii. Daftar Komponen  
Berisi daftar komponen yang harus dipatuhi sebagai bagian dari pesanan.
  - iii. Nilai Pesanan  
Nilai yang akan dibayarkan saat pesanan selesai dan berhasil diantarkan.
- b. Pesanan dibangkitkan secara random setiap mulainya hari baru. Pemesan harus terdefinisi pada kota, daftar komponen juga harus ditentukan dari daftar komponen yang ada. Nilai pesanan dapat ditentukan secara random asalkan masih menghasilkan keuntungan saat pesanan tersebut diselesaikan.
- c. Saat dilakukan delivery ke rumah pelanggan, maka melakukan pencocokan antara pemesan dengan pesanan yang telah diselesaikan untuk menentukan pesanan mana yang akan diantar.
- d. Pemain hanya dapat melihat melihat satu orderan saja karena orderan yang masuk bersifat antrian.

## 7. Inventory

- a. Inventory digunakan untuk menyimpan komponen yang dibeli dan juga pesanan yang telah selesai dirakit.
- b. Pada permainan ini, inventory pemain tidak mempunyai batasan.

- c. Setiap item yang sama akan dikelompokkan menjadi satu, bukan terpisah dalam entitas yang berbeda-beda.

## 8. Commands

Setelah state awal permainan diinisiasi, maka game akan selalu menunggu masukan dari pemain. Berikut adalah masukan *command* yang dapat diberikan oleh pengguna.

### a. MOVE

Digunakan untuk melakukan perpindahan menuju lokasi lain.

Contoh skenario 1:

```
ENTER COMMAND: MOVE  
Kamu berada pada base.  
Daftar lokasi yang dapat dicapai:  
1. Pelanggan 2  
2. Pelanggan 5  
Nomor tujuan: 2  
Kamu telah mencapai lokasi Pelanggan 5!
```

Sesuai dengan graph yang ada diakhir spek, lokasi yang bisa dicapai hanya lokasi yang terhubung melalui graph yang terdefinisi.

### b. STATUS

Digunakan untuk melihat status pemain.

Contoh skenario 1:

```
ENTER COMMAND: STATUS  
Uang tersisa: $4000  
Build yang sedang dikerjakan: pesanan 23 untuk pelanggan 4.  
Lokasi: pemain sedang berada pada base.
```

### c. STARTBUILD

Digunakan untuk memulai sebuah build.

Contoh skenario 1:

```
ENTER COMMAND: STARTBUILD  
Kamu telah memulai pesanan 5 untuk pelanggan 3.
```

Sesuai dengan graph yang ada diakhir spek, lokasi yang bisa dicapai hanya lokasi yang terhubung melalui graph yang terdefinisi.

d. FINISHBUILD

Digunakan untuk menyelesaikan sebuah build.

Contoh skenario 1:

ENTER COMMAND: **FINISHBUILD**  
Pesanan 5 telah selesai. Silahkan antar ke pelanggan 3!

Komponen yang digunakan telah sesuai dengan pesanan yang dibuat oleh pelanggan.  
Build dinyatakan selesai.

Contoh skenario 2:

ENTER COMMAND: **FINISHBUILD**  
Komponen yang dipasangkan belum sesuai dengan pesanan, build belum dapat diselesaikan.

Dalam skenario kedua mungkin terjadi kita komponen yang dipasang masih sebagian saja, urutannya salah ataupun komponen yang digunakan tidak sesuai pesanan.

e. ADDCOMPONENT

Digunakan untuk memasang komponen pada komputer.

ENTER COMMAND: **ADDCOMPONENT**  
Komponen yang telah terpasang:  
1. Gigabyte B550m Aorus Pro  
2. Ryzen 5 5950x  
Komponen yang tersedia:  
1. NVIDIA RTX 3099Ti Super  
2. MJOLNIR 9.7 Liter SFF PC Case  
3. Vengeance DDR4 16GB  
Komponen yang ingin dipasang: **3**  
Komponen berhasil dipasang!

f. REMOVECOMPONENT

Digunakan untuk mencopot komponen terakhir yang telah dipasang.

ENTER COMMAND: **REMOVECOMPONENT**  
Komponen Ryzen 5 5950x berhasil dicopot!

g. SHOP

Digunakan untuk membeli komponen pada saat berada di shop:

Contoh skenario 1:

```
ENTER COMMAND: SHOP  
Komponen yang tersedia:  
1. NVIDIA RTX 3099Ti Super - $1200  
2. MJOLNIR 9.7 Liter SFF PC Case - $250  
Komponen yang ingin dibeli: 2  
Masukkan jumlah yang ingin dibeli: 1  
Komponen berhasil dibeli!
```

Pada skenario di atas, uang cukup untuk membeli komponen yang diinginkan.

Contoh skenario 2:

```
ENTER COMMAND: SHOP  
Komponen yang tersedia:  
1. NVIDIA RTX 3099Ti Super - $1200  
2. MJOLNIR 9.7 Liter SFF PC Case - $250  
Komponen yang ingin dibeli: 2  
Masukkan jumlah yang ingin dibeli: 3  
Uang tidak cukup!
```

Pada skenario di atas, uang tidak cukup untuk membeli komponen.

h. DELIVER

Digunakan untuk mengantarkan pesanan komputer yang sudah selesai.

```
ENTER COMMAND: DELIVER  
Pesanan #23 berhasil diantarkan kepada pelanggan 3!
```

i. END\_DAY

Digunakan untuk mengakhiri hari menuju keesokan hari.

```
ENTER COMMAND: END_DAY
```

j. SAVE (bonus)

Digunakan untuk menyimpan status permainan sekarang ke dalam file eksternal.

```
ENTER COMMAND: SAVE  
Lokasi save file: C:\User\Documents\save_file.txt  
Game berhasil di save!
```

k. MAP

Digunakan untuk menampilkan peta dan juga posisi pemain saat ini.

Contoh :

```

ENTER COMMAND: MAP
*****
*1          4      *
*              *
*              6    *
*      3          *
*              *
*  S              *
*      7          *
* 2              *
*              5    *
*  P              *
*****

```

#### I. EXIT

Digunakan untuk keluar dari permainan/program.

```

ENTER COMMAND: EXIT

```

## Konfigurasi Permainan

File konfigurasi digunakan untuk membuat permainan dapat diatur sesuai dengan isi file. Terdapat beberapa file konfigurasi yang akan digunakan pada permainan ini diantaranya konfigurasi map, dan item yang ada dalam permainan.

Format konfigurasi map adalah sebagai berikut:

1. Pada baris pertama terdapat ukuran peta yaitu jumlah baris dan kolom
2. Pada baris kedua terdapat  $n$  jumlah bangunan yang terdefinisi pada peta
3. Pada  $n$  baris berikutnya terdapat lokasi bangunan, baris pertama untuk base, baris kedua shop lalu sisanya menjadi pelanggan mulai dari 1 sampai seterusnya.
4. Pada bagian selanjutnya terdapat matriks yang menggambarkan hubungan jalur antara bangunan  $i$  dengan bangunan  $j$ . Jika pada baris  $i$  dan baris  $j$  bernilai 1, maka bangunan  $i$  dan bangunan  $j$  terhubung dengan jalan. Bangunan dinomori 1 mulai dari base, 2 untuk shop, lalu 3 untuk pelanggan nomor 1 dst.

Contoh file konfigurasi map:

```

10 15
9
B 10 3
S 6 3
C 1 1

```

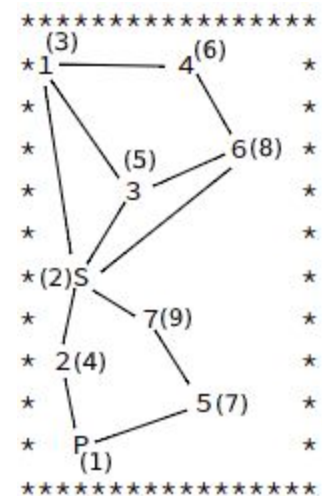
C	8	2							
C	4	6							
C	1	9							
C	9	10							
C	3	12							
C	7	7							
0	0	0	1	0	0	1	0	0	
0	0	1	1	1	0	0	1	1	
0	1	0	0	1	1	0	0	0	
1	1	0	0	0	0	0	0	0	
0	1	1	0	0	0	0	1	0	
0	0	1	0	0	0	0	1	0	
1	0	0	0	0	0	0	0	1	
0	1	0	0	1	1	0	0	0	
0	1	0	0	0	0	1	0	0	

File konfigurasi di atas bersesuaian dengan tampilan peta berikut:

```

*****
*1      4      *
*                               *
*                               *
*          6      *
*      3          *
*                               *
*      S          *
*          7      *
*  2              *
*          5      *
*      P          *
*****

```



File konfigurasi untuk daftar item dan save permainan (bonus) dibebaskan kepada mahasiswa.



## Daftar ADT yang Digunakan

Anda diwajibkan menggunakan ADT di bawah ini. Selain itu, Anda dapat pula menggunakan ADT lain, namun cantumkan analisis alasan kenapa menggunakan ADT tersebut pada laporan.

1. **ADT Point**

ADT ini digunakan untuk menyimpan posisi dari berbagai objek seperti pemain, pelanggan dan shop.

2. **ADT Array Dinamis**

ADT ini digunakan untuk merepresentasikan inventory dari pemain.

3. **ADT Matriks**

ADT ini digunakan sebagai representasi tampilan peta.

4. **ADT Mesin Karakter dan Mesin Kata**

ADT ini digunakan untuk:

- a. Membaca informasi peta dari file eksternal,
- b. Membaca *command* dari interaksi user terhadap program, dan
- c. Membaca *state* dari *game* yang sudah pernah disimpan

5. **ADT Queue**

ADT ini digunakan sebagai representasi dari antrian pesanan pelanggan.

6. **ADT Stack**

ADT ini digunakan sebagai representasi dari model perakitan komputer. Komponen yang dapat diakses hanya komponen yang berada pada TOP dari stack.

7. **ADT Graph (variasi multilist)**

ADT ini digunakan untuk merepresentasikan jalur pada tempat-tempat yang terdefinisi pada peta.

8. **ADT Lain**

ADT ini dapat dibuat dan digunakan untuk abstraksi berbagai hal lain sesuai kebutuhan masing-masing. Mahasiswa dipersilakan untuk mendefinisikan sendiri.

## Kelompok

Pembagian kelompok dapat diakses pada link ini:

<https://drive.google.com/file/d/1wMPuVbWCR57BJ1mq9g3aURUweHA4RNx7/view?usp=sharing>

## **Deadline**

Batas pengumpulan akhir tugas besar ini adalah **5 Desember 2020 23:59**.