

Tugas Besar Logika Komputasional – IF2121

I Got Scammed By My Client And Have to Restart My Life As A Farmer

“A Farmer’s Life is Not That Bad, I Think”

Versi: 8 November 2021

A. Topik

Membuat sebuah *farm simulation role-playing game* sebagai seorang programmer yang harus bertani untuk membayar hutangnya dengan menggunakan bahasa pemrograman deklaratif Prolog (gunakan GNU Prolog).



Gambar 1 Wajah *shock* Anda ketika ditipu oleh klien

Implementasi tugas besar harus mengandung materi:

1. Rekurens
2. *List*
3. *Cut*
4. *Fail*
5. *Loop*

B. Jadwal Penugasan/Penyetoran

No	Waktu	Kegiatan	Deliverables
1	Senin, 8 November 2021	Pembahasan Pra-Praktikum, Rilis Tugas Besar IF2121	-

2	Senin, 15 November 2021 (15:40)	Asistensi Tugas Besar, pengerjaan, dan pengumpulan <i>draft</i> kasar tugas besar	Gambaran kasar tentang cerita dan alur permainan, atau rencana fakta dan <i>rule</i> yang akan digunakan, baik yang menggunakan list, <i>control loop</i> , atau lainnya yang dapat menunjukkan <i>progress</i> pekerjaan kelompok
3	Senin, 22 November 2021 (15:40)	Asistensi Tugas Besar dan pengumpulan <i>progress</i> kedua dari tugas besar.	<i>Source code</i> program yang minimal berisi: <ul style="list-style-type: none"> • Fakta-fakta terkait peta dan pemain • Implementasi <i>rule-rule</i> kendali dasar (<i>start, help, quit, inventory, dll</i>)
4	Senin, 29 November 2021 (15:40)	Asistensi Tugas Besar, Pengumuman mekanisme pembuatan Demo Tugas Besar, dan pengumpulan <i>final deliverables</i> Tugas Besar Logika Komputasional.	Seluruh <i>deliverables</i> dari Tugas Besar ini
5	Minggu, 5 Desember 2021 (23:59)	<i>Deadline</i> pengumpulan Demo Tugas Besar	Video demo program yang sudah dibuat

C. Tujuan

Tujuan dari tugas besar ini adalah mengkombinasikan berbagai keterampilan dan teknik yang telah dipelajari dalam perkuliahan Logika Komputasional IF2121, pra-praktikum, dan eksplorasi mandiri mengenai Logika Komputasional dan Prolog.

D. Domain Permasalahan

Claire adalah seorang gadis yang baru saja ditipu oleh seorang kliennya yang hilang begitu saja secara *GAIB* tanpa membayar proyek yang ia bangun dengan susah payah. Selain menjadi korban penipuan, Claire juga terlilit hutang sebesar 20000 gold yang perlu dilunasi dalam jangka waktu 1 tahun. Oleh karena itu, Claire memutuskan untuk pulang ke kampung halamannya dan melanjutkan usaha tani milik kakeknya untuk bisa melanjutkan hidup dan membayar hutangnya.

Tugas Anda adalah mengimplementasikan sebuah *farm simulation role-playing game*. Buatlah permainan **sekreatif mungkin**. Kreativitas Anda akan menjadi bagian dari penilaian asisten. Spesifikasi yang diberikan merupakan **batas minimum** yang harus dikerjakan. Oleh

karena itu, sebisa mungkin kalian membuat *farm simulation role-playing game* yang lebih dari spesifikasi. Berikut beberapa spesifikasi yang harus Anda buat dalam program Anda:

1. Pemain

Terdapat seorang pemain yang baru saja datang ke suatu desa baru. Pemain akan memilih salah satu dari 3 *specialty* yang disediakan (Farmer, Fisherman, Rancher) yang nantinya akan memberikan *advantage* yang berbeda-beda untuk pemain. Setiap *specialty* akan mendapatkan keuntungan masing-masing ketika menyelesaikan aktivitas yang berkaitan dengan *specialty*-nya, yaitu tambahan *experience point* yang diberikan kepada pemain.

Experience point seorang pemain dapat dibagi menjadi *experience point* setiap *specialty* dan *experience point* pemain secara keseluruhan, sama halnya dengan *level* yang dimiliki pemain. Sistem *level up* dan *growth rate* dari pemain dibebaskan implementasinya. Selain itu, pemain juga memiliki atribut *money* yang mencatat jumlah uang yang dimilikinya pada saat ini. Pemain mampu menambah jumlah uangnya dengan menjual komoditas yang dihasilkan dari aktivitas yang dilakukannya.

Contoh Program

```
|  ?- startGame.
```

Harvest Star!!!

Let's play and pay our debts together!

[illegible]

(47 ms) yes

```
| ?- start.  
Welcome to Harvest Star. Choose your job  
1. Fisherman  
2. Farmer  
3. Rancher  
> 1
```

```
You choose Fisherman, let's start farming
```

```
(108 ms) yes
```

```
| ?- status.
```

```
Your status:
```

```
Job: Fisherman
```

```
Level: 1
```

```
Level farming: 1
```

```
Exp farming: 56
```

```
Level fishing: 1
```

```
Exp fishing: 76
```

```
Level ranching: 1
```

```
Exp ranching: 56
```

```
Exp: 0/300
```

```
Gold: 1000
```

```
(20 ms) yes
```

2. Peta

Peta pada permainan minimal berukuran 10 untuk panjang dan lebarnya. Posisi pemain pada peta direpresentasikan dengan huruf P. Peta ini dapat ditampilkan ke layar dengan memanggil sebuah *command*. Pada pinggiran peta terdapat pagar yang direpresentasikan dengan karakter `#`. Terdapat lokasi-lokasi spesial pada peta yang akan dijelaskan pada bagian berikutnya, dan dapat direpresentasikan pada peta sebagai berikut:

- M : *Marketplace*
- R : *Ranch*
- H : *House*
- Q : Tempat pengambilan *quest*
- o : *Tile air*
- = : *Digged tile*

Tata letak masing-masing lokasi ini dibebaskan, asal tetap menarik, tidak terlalu berdekatan, dan dapat memberikan ruang gerak yang mudah bagi pemain. Implementasi lokasi dibebaskan kepada peserta, bisa berupa suatu titik pada peta maupun suatu area khusus yang perintah-perintah terkait hanya bisa digunakan di area tersebut saja.

Contoh Program

```
| ?- map.  
#####  
#-----#  
#-----#  
#-----Q-----#  
#-----#  
#-----R-----#  
#-----H-----#  
#-----P-----#  
#-----ooo-----#  
#---ooo---#
```

```
#----ooo-----#
#-----#
#-----M-----#
#-----#
#-----#
#-----#
#-----#
#-----#
#####
```

(62 ms) yes

3. *Inventory*

Inventory digunakan untuk menyimpan barang dan *equipments* tidak terpakai yang didapatkan oleh pemain dari semua aktivitas yang ada. Pemain hanya dapat menyimpan 100 buah item di dalamnya. Mekanisme menyimpan dan membuang barang yang ada di *inventory* dibebaskan. *Inventory* dapat diakses di manapun dengan memanggil menu *inventory*.

Contoh Program

```
| ?- inventory.
```

```
Your inventory (26 / 100)
```

```
1 Level 1 shovel
```

```
1 Level 1 fishing rod
```

```
3 Carrots
```

```
4 Corns
```

```
3 Eggs
```

```
5 Salmons
```

```
3 Tunas
```

(62 ms) yes

```
| ?- throwItem.
```

```
Your inventory
```

```
1 Level 1 shovel
```

```
1 Level 1 fishing rod
```

```
3 Carrot
```

```
4 Corn
```

```
3 Egg
```

```
5 Salmon
```

```
3 Tuna
```

```
What do you want to throw?
```

```
> tuna.
```

```
You have 3 tuna. How many do you want to throw?
```

```
> 3
```

```
You threw away 3 tuna.
```

(43 ms) yes

```
| ?- throwItem.
```

```
Your inventory
1 Level 1 shovel
1 Level 1 fishing rod
3 Carrot
4 Corn
3 Egg
5 Salmon
3 Tuna

What do you want to throw?

> tunas.

You have 3 tunas. How many do you want to throw?

> 5.

You don't have enough tunas. Cancelling...

(43 ms) yes
```

4. *Items*

Items terdiri dari *equipments* yang bisa digunakan oleh pemain dalam setiap aktivitas yang dilakukan. Selain itu, *items* juga dapat berupa barang yang didapat dari setiap aktivitas yang dilakukan, maupun barang yang dibutuhkan untuk melakukan suatu aktivitas, misalnya aktivitas bertani membutuhkan *seed* yang akan ditanamkan, kemudian mendapatkan hasil panen berupa sayuran atau buah-buahan. Setiap komoditas diharapkan memiliki jenis atau varian yang berbeda dan ketika dijual di *marketplace* juga memberikan harga jual yang berbeda pula.

Setiap *equipments* memiliki *level* yang dapat ditingkatkan oleh pemain dan setiap *level* dapat memberikan keuntungan yang berbeda pada aktivitas yang dilakukan pemain. Semua *equipments* dapat digunakan pemain dengan spesialisasi apapun, tetapi bisa jadi tidak kompatibel dan menurunkan *stats* dari *equipment* tersebut. Silakan berkreasi sendiri dalam menentukan efek dari setiap *item* ketika dipakai. **Tidak ada jumlah minimal varian *item*, tetapi diharapkan tidak terlalu sedikit.**

5. *Quest*

Quest merupakan misi yang harus diselesaikan oleh pemain, dan diambil dari tempat pengambilan *Quest*. Sebuah quest terdiri atas sebuah tripel bilangan (x, y, z), yang berarti untuk menyelesaikan *quest* tersebut pemain harus mengumpulkan x hasil panen, y ikan, dan z telur. Apabila menyelesaikan sebuah misi, pemain menerima *reward* berupa tambahan EXP serta sejumlah gold. Dalam satu waktu, tidak boleh terdapat dua *quest* yang aktif, yang berarti pemain harus menyelesaikan sebuah misi terlebih dahulu sebelum dapat menerima misi yang baru. Implementasi penentuan misi serta hadiahnya dibebaskan, dan harus dijelaskan pada laporan.

Contoh Program

```
| ?- quest. /* command dijalankan ketika player berada di tile Q*/
You got a new quest!

You need to collect:
- 3 harvest item
- 5 fish
- 4 eggs

(25 ms) yes

| ?- quest.
You have an on-going quest!

(10 ms) yes
```

6. Exploration Mechanism

Pemain dapat berpindah sebanyak 1 *tile* pada peta dengan arah atas, bawah, kiri, dan kanan. Jika pemain berada pada lokasi yang unik (seperti *marketplace*, *ranch*, *farm*, atau *house*), pemain dapat melakukan interaksi sesuai dengan fungsi yang valid lokasi yang unik tersebut. Dikarenakan belum ada teknologi terapung, pemain tidak bisa bergerak atau menempati *tile* danau dan hanya bisa memancing dari pinggiran danau.

Berikut adalah hal-hal yang akan terjadi ketika Anda berinteraksi dengan lokasi unik:

- Ketika berinteraksi dengan *tile marketplace*, pemain akan masuk ke menu *marketplace* yang detail lebih lanjutnya dapat dilihat di bagian **marketplace**.
- Ketika berinteraksi dengan *tile house*, pemain akan masuk ke menu house yang detail lebih lanjutnya dapat dilihat di bagian **house**.
- Ketika berinteraksi dengan *tile ranch*, *farm*, atau danau, pemain bisa melakukan interaksi dengan fungsi yang sesuai untuk setiap lokasi tersebut.

Contoh Program

```
| ?- map.  
#####  
#-----#  
#-----#  
#-----Q-----#  
#-----F-----#  
#-----R-----#  
#-----H-----#  
#-----P-----#  
#-----ooo-----#  
#-----oooo-----#  
#-----ooo-----#  
#-----#  
#-----M-----#  
#-----#  
#-----#  
#-----#  
#-----#  
#-----#  
#####
```

(62 ms) yes

```
| ?- a.  
You moved west.
```

(22 ms) yes

```
| ?- map.  
#####  
#-----#  
#-----#  
#-----Q-----#  
#-----F-----#  
#-----R-----#  
#-----H-----#  
#-----P-----#  
#-----ooo-----#  
#-----oooo-----#  
#-----ooo-----#  
#-----#  
#-----M-----#  
#-----#  
#-----#  
#-----#  
#-----#  
#-----#  
#####
```

(62 ms) yes

```
| ?- s.  
You can't get into water!
```

(22 ms) yes

7. *Farming*

Aktivitas *farming* adalah aktivitas menanam dan memanen tanaman. Pemain dapat melakukan aktivitas *farming* di *tile* manapun yang bukan merupakan *special tile*. Untuk menghasilkan suatu hasil panen, pemain pertama-tama perlu untuk menggali *tile*, menanam *seed*, menunggu hingga tanaman siap dipanen (silakan definisikan sendiri lama waktu menunggu), kemudian memanen tanaman jika sudah siap dipanen. Perhatikan bahwa setelah menggali, pemain berada di atas *tile* yang baru saja selesai digali. Selain itu, apabila pemain mencoba untuk memanen tanaman yang belum siap, program akan memberikan pesan bahwa tanaman belum bisa dipanen.

Contoh Program

```
| ?- map.
#####
#-----#
#-----#
#-----Q-----#
#-----#
#-----R-----#
#-----H-----#
#-----#
#----ooo-----#
#---ooooo-----#
#---ooo-----#
#-----#
#-----M-----#
#-----#
#-----#
#-----P-----#
#-----#
#-----#
#####

(62 ms) yes

| ?- dig.
You digged the tile.

(12 ms) yes

/* Tampilan peta akan menjadi seperti ini:
#####
#-----#
#-----#
#-----Q-----#
#-----#
#-----R-----#
#-----H-----#
#-----#
#----ooo-----#
#---ooooo-----#
#---ooo-----#
#-----#
#-----M-----#
#-----#
#-----P-----#
```

```

#-----#
#-----#
#-----#
#####
(P dipindahkan satu tile ke atas agar kondisi akhir tile terlihat)
*/

| ?- plant.
You have:
- 1 corn seed
- 2 tomato seed
What do you want to plant?
corn

You planted a corn seed.
(56 ms) yes

/* Tampilan peta akan menjadi seperti ini:
#####
#-----#
#-----#
#-----Q-----#
#-----#
#-----R-----#
#-----H-----#
#-----#
#-----ooo-----#
#---ooooo-----#
#---ooo-----#
#-----#
#-----M-----#
#-----#
#-----P-----#
#-----C-----#
#-----#
#-----#
#####
(P dipindahkan satu tile ke atas agar kondisi akhir tile terlihat)
*/

| ?- harvest.
You harvested corn.

(22 ms) yes

```

8. Fishing

Aktivitas *fishing* adalah aktivitas memancing untuk mendapatkan ikan. Aktivitas *fishing* dapat dilakukan ketika pemain sedang berada di sekitar *tile* air. Ketika melakukan aktivitas *fishing*, pemain akan bisa mendapatkan suatu ikan secara acak atau tidak mendapatkan ikan apapun sama sekali. Setelah selesai melakukan aktivitas, pemain akan mendapatkan *exp fishing* (*exp overall* juga akan otomatis bertambah), baik ketika berhasil mendapatkan ikan ataupun tidak. Semakin tinggi level *fishing* seorang player, semakin besar kemungkinan mendapatkan ikan dan jenis-jenis ikan yang bisa didapatkan akan lebih banyak dan harga jualnya lebih mahal.

Contoh Program

```
| ?- map.  
#####  
#-----#  
#-----#  
#-----Q-----#  
#-----#  
#-----R-----#  
#-----H-----#  
#-----P-----#  
#----ooo-----#  
#---ooooo-----#  
#----ooo-----#  
#-----#  
#-----M-----#  
#-----#  
#-----#  
#-----#  
#-----#  
#-----#  
#####  
  
(62 ms) yes  
  
| ?- fish.  
You got tuna!  
You gained 10 fishing exp!  
  
      (12 ms) yes  
  
| ?- fish.  
You didn't get anything!  
You gained 5 fishing exp!  
  
(12 ms) yes
```

9. *Ranching*

Aktivitas *ranching* dapat dilakukan ketika pemain sedang berada di atas *tile ranch*. Ketika melakukan aktivitas *ranching*, pemain dapat melihat list hewan ternak beserta jumlahnya yang ia miliki di *ranch*. Saat di dalam *ranch*, pemain dapat memilih salah satu hewan ternak untuk diambil hasilnya setelah beberapa waktu berlalu. Pemain akan mendapatkan *exp ranching* (*exp overall* juga akan otomatis bertambah) ketika mengambil produk hasil dari hewan ternak. Semakin tinggi level *ranching* seorang, semakin cepat pemain dapat mengambil hasil dari hewan ternak yang ia miliki.

Contoh Program

```
| ?- map. /* P menimpa R */
#####
#-----#
#-----#
#-----Q-----#
#-----#
#-----P-----#
#-----H-----#
#-----#
#-----ooo-----#
#---ooooo-----#
#---ooo-----#
#-----#
#-----M-----#
#-----#
#-----#
#-----#
#-----#
#####

(62 ms) yes

| ?- ranch.
Welcome to the ranch! You have:
2 chicken
1 sheep
1 cow

What do you want to do?

(32 ms) yes

| ?- chicken.
Your chicken lays 2 eggs.
You got 2 eggs!
You gained 6 ranching exp!

(12 ms) yes

| ?- sheep.
Your sheep hasn't produced any wool.
Please check again later.

(12 ms) yes
```

10. Marketplace

Marketplace merupakan tempat untuk membeli barang dan menjual barang. Ketika pemain berinteraksi dengan objek *marketplace*, pemain kemudian akan diberikan opsi untuk melakukan pembelian atau penjualan. Di *marketplace*, pemain dapat membeli barang, *upgrade equipments*, *seeds*, dan hewan, serta menjual barang-barang yang didapatkan dari aktivitas-aktivitas yang telah dilakukan.

Contoh Program

```
| ?- market. /* command dilakukan ketika berada di tile M */
```

```
What do you want to do?
```

1. Buy
2. Sell

```
(10 ms) yes
```

```
| ?- buy.
```

```
What do you want to buy?
```

1. Carrot seed (50 golds)
2. Corn seed (50 golds)
3. Tomato seed (50 golds)
4. Potato seed (50 golds)
5. Chicken (500 golds)
6. Sheep (1000 golds)
7. Cow (1500 golds)
8. Level 2 shovel (300 golds)
9. Level 2 fishing rod (500 golds)

```
(30 ms) yes
```

```
| ?- 2.
```

```
How many do you want to buy?
```

```
(10 ms) yes
```

```
| ?- 5.
```

```
You have bought 5 corn seeds.
```

```
You are charged 250 golds.
```

```
What do you want to buy?
```

1. Carrot seed (50 golds)
2. Corn seed (50 golds)
3. Tomato seed (50 golds)
4. Potato seed (50 golds)
5. Chicken (500 golds)
6. Sheep (1000 golds)
7. Cow (1500 golds)
8. Level 2 shovel (300 golds)
9. Level 2 fishing rod (500 golds)

```
(30 ms) yes
```

| ?- exitShop.
Thanks for coming.

(5 ms) yes

| ?- market. /* command dilakukan ketika berada di tile M */
What do you want to do?
1. Buy
2. Sell

(10 ms) yes

| ?- sell.
Here are the items in your inventory
- 5 carrots
- 3 corns
- 5 eggs
What do you want to sell?

(30 ms) yes

| ?- egg.
How many do you want to sell?

(30 ms) yes

| ?- 3.
You sold 3 eggs.
You received 300 golds.

Here are the items in your inventory
- 5 carrots
- 3 corns
- 5 eggs
What do you want to sell?

(30 ms) yes

11. House

Ketika pemain masuk ke menu *house*, pemain akan diberikan opsi untuk tidur atau kembali ke peta. Apabila pemain memilih untuk tidur, maka waktu di *game* akan berpindah ke hari selanjutnya. Apabila Anda mengerjakan spek bonus *diary*, opsi untuk menulis *diary* dan membaca *diary* akan muncul di menu *house*.

Contoh Program

```
| ?- house. /* command dilakukan ketika pemain berada di menu house */
What do you want to do?
- sleep
- writeDiary
- readDiary
- exit

(30 ms) yes

| ?- sleep.
You went to sleep

Day 2
Season: Spring /* apabila mengerjakan spek bonus pergantian musim dan cuaca */
Weather: Sunny /* apabila mengerjakan spek bonus pergantian musim dan cuaca */

(50 ms) yes

| ?- writeDiary.
Write your diary for Day 1

(50 ms) yes

| ?- Today is my first day as a farmer blablabla.
Day 1 entry saved

(40 ms) yes

| ?- readDiary.
Here are the list of your entries:
- Day 1
- Day 3
- Day 4

Which entry do you want to read?

(50 ms) yes

| ?- 1.
Here is your entry for day 1:
Today is my first day as a farmer blablabla.

(40 ms) yes
```

12. Goal State

Pemain akan dinyatakan menang apabila pemain berhasil mengumpulkan uang sebesar 20000 gold dalam jangka waktu 1 tahun. Silahkan definisikan sendiri mekanisme perhitungan dan pergantian waktu, misalnya dapat dihitung per jumlah *move*. Apabila Anda mengerjakan tugas bonus sistem pergantian musim, maka 1 tahun adalah waktu dari musim semi ke musim semi lagi.

Contoh Program

```
| ?- 3.  
You sold 3 eggs.  
You received 300 golds.  
  
/* Di titik ini, pemain berhasil mengumpulkan 20000 golds*/  
  
Congratulations! You have finally collected 20000 golds!  
  
(30 ms) yes
```

13. Fail State

Pemain dinyatakan kalah apabila pemain gagal mengumpulkan uang sebesar 20000 gold dalam jangka waktu 1 tahun.

Contoh Program

```
| ?- /* Aktivitas apapun, telah melewati batas waktu yang ditentukan */  
  
You have worked hard, but in the end result is all that matters.  
May God bless you in the future with kind people!  
  
(62 ms) yes
```

14. Bonus

- Anda mendengar kabar bahwa seorang *Alchemist* legendaris baru saja tiba di desa Anda. *Alchemist* ini menyediakan *specialties potion* yang bisa meningkatkan keterampilan Anda, tetapi harganya sangat mahal dan jumlahnya juga terbatas karena *Alchemist* ini tidak ingin ada eksploitasi *item* tersebut. Anda dapat membeli dan memakai *potion* ini.
- Anda sering bertemu dengan Peri Tidur di dalam mimpi Anda. Jika Anda beruntung, Anda bisa mendapatkan tawaran untuk pergi ke posisi mana saja yang Anda inginkan dan Anda bisa tempati ketika Anda terbangun sehingga Anda tidak perlu melakukan perjalanan yang melelahkan itu.
- Terdapat sistem pergantian cuaca dan musim. Permainan diawali dengan musim semi kemudian akan berganti dengan urutan musim semi → musim panas → musim gugur → musim dingin dengan durasi tiap musim dibebaskan. Cuaca ditentukan secara acak

- tetapi disesuaikan dengan musimnya (contoh. ketika musim panas kemungkinan cerah akan lebih tinggi, salju hanya bisa turun di musim dingin). *Marketplace* menjual varian seeds yang berbeda di setiap musimnya.
- d. Sewaktu luang Anda, Anda ingin menuliskan apa yang telah Anda alami pada *diary* yang Anda miliki. Di masa depan, anak cucu Anda dapat membuka dan membaca *diary* Anda, lalu seolah-olah kembali ke masa lalu Anda dan melakukan rekoleksi terhadap kehidupan Anda yang penuh tantangan.
 - e. Kreativitas tambahan dalam implementasi *game* akan selalu diapresiasi (contoh: alur cerita, animasi dalam *activity mechanism*, fitur-fitur tambahan lainnya).

E. Kelompok

Pembagian kelompok ditentukan sendiri oleh mahasiswa dengan mengisi [sheet kelompok](#) berikut ini dengan 1 kelompok terdiri dari 4 mahasiswa yang berasal dari kelas yang sama. Silakan lihat instruksi lebih lanjut di *sheet* kelompok di atas.

F. FAQ

Pertanyaan dapat ditanyakan pada link [FAQ](#) berikut. Pastikan pertanyaan yang ditanyakan tidak berulang.

G. Aturan

Terdapat beberapa hal yang harus diperhatikan dalam pengerjaan tugas ini, yakni:

1. *Query* dan keluaran tidak harus persis dengan contoh program yang diberikan. Hal-hal yang tidak dispesifikkan dapat dianggap sebagai hal yang bebas diimplementasikan seperti apa, silakan berkreativitas sendiri.
2. Semua perubahan yang mungkin Anda lakukan diimplementasi Anda, ataupun fitur-fitur lain yang Anda ubah atau tambahkan, dapat dituliskan di laporan agar dapat dinilai oleh asisten dengan lengkap.
3. Apabila Anda mencari dan mencontoh kode dari Internet, harap cantumkan sumbernya dalam bentuk komentar dalam program.
4. Jika terdapat hal yang tidak dimengerti, silahkan ajukan pertanyaan kepada asisten melalui link FAQ yang telah diberikan di atas. Pertanyaan yang diajukan secara personal ke asisten tidak akan dijawab untuk menghindari perbedaan informasi yang didapatkan oleh peserta praktikum.
5. Dilarang melakukan **plagiarisme**. Pelanggaran pada poin ini akan menyebabkan pemberian **nilai E** pada setiap anggota kelompok yang melakukan maupun memberi.

H. Deliverables

File yang dikumpulkan berupa hasil zip semua file yang telah dikerjakan dengan format penamaan **IF2121_KXX_GYY.zip** dengan XX adalah nomor kelas dan YY adalah nomor kelompok. File zip terdiri dari:

1. *Source code* dari program yang telah kelompok Anda buat.
2. File README yang berisi bagaimana cara melakukan eksekusi program dan spesifikasi lingkungan untuk menjalankan program.
3. File-file pendukung lainnya (jika ada).
4. Laporan hasil kerja dengan format penamaan **Laporan_KXX_GYY.pdf** dengan XX adalah nomor kelas dan YY adalah nomor kelompok yang terdiri atas:
 - a. Halaman *cover* yang memuat judul tugas, kode dan nama mata kuliah, dan identitas anggota kelompok
 - b. Penjelasan setiap *command* yang kelompok Anda telah buat, termasuk command bonus (jika ada), penjelasan meliputi:
 - i. Kegunaan command tersebut
 - ii. Skenario-skenario penggunaannya (beserta contoh)
 - iii. Tidak perlu menjelaskan cara kerja *command*
 - c. Hasil eksekusi program berupa jalannya alur permainan (dalam bentuk *screenshot*)
 - d. Pembagian dan persentase kerja masing-masing anggota kelompok
5. Pengumpulan akan dilakukan pada *Google Form* yang dapat diakses pada [pranala](#) berikut. Pastikan jenis *deliverable* yang dikumpulkan telah sesuai dengan *progress* yang diminta pada saat pengumpulan.
 - a. *Deliverables 1: Deadline* 15 November 2021 pukul 15.40
 - b. *Deliverables 2: Deadline* 22 November 2021 pukul 15.40
 - c. *Final Deliverables: Deadline* 29 November 2021 pukul 15.40