

**IF2121 - Logika Komputasional**  
**Tugas Besar**  
**CAMEL UP!**



Disusun Oleh:

13523041 Hanif Kalyana Aditya

13523051 Ferdinand Gabe Tua Sinaga

13523063 Syahrizal Bani Khairan

13523111 Muhammad Iqbal Haidar

**Program Studi Teknik Informatika**  
**Sekolah Teknik Elektro dan Informatika**  
**Institut Teknologi Bandung**  
**2024**

## Daftar Isi

Daftar Isi.....	2
BAB 1: Deskripsi Masalah.....	3
BAB 2: Penjelasan Command.....	4
2.1. Command startGame.....	4
2.2. Command cek_map.....	4
2.3. Command cek_info.....	4
2.4. Command investasi.....	5
2.5. Command papan_investasi.....	5
2.6. Command jalankanUnta.....	6
2.7. Command pasangTrap.....	7
2.8. Command nextTurn.....	7
2.9. Command godsHand.....	8
BAB 3: Hasil Eksekusi Program.....	9
Lampiran.....	12
Referensi.....	13

## BAB 1: Deskripsi Masalah

Program ini dibuat untuk mengimplementasikan sebuah permainan bernama “Camel Up!”. Permainan diinisiasi dengan *start game*. *Command* ini akan melakukan *setup* berupa pengacakan urutan pemain serta inisiasi kartu, poin, dan *trap* tiap pemain. Pada tiap giliran pemain, pemain dapat melakukan:

- pilih maksimal 1 aksi di antara: investasi, jalankan unta, pasang trap
- *display map* dengan jumlah tidak terbatas;
- cek info dengan jumlah tidak terbatas.

Pemain dapat mengakhiri gilirannya dengan menjalankan end turn.

Permainan akan berakhir ketika terdapat unta yang melewati garis *finish* (angka dadu untuk melewati garis *finish* tidak harus tepat dengan jumlah langkah yang dibutuhkan untuk sampai ke garis *finish*). Program secara otomatis akan menjalankan end game.

## **BAB 2: Penjelasan Command**

### **2.1. Command *startGame***

#### **2.1.1. Kegunaan Command**

Command ini digunakan untuk menginisialisasi program sebelum permainan dimulai. Pada command ini dilakukan pendataan pemain, pembuatan map serta inisialisasi dadu

#### **2.1.2. Skenario-Skenario Penggunaan**

- Skenario 1

Saat baru pertama kali menjalankan program, pengguna harus menjalankan perintah *startGame* agar bisa memainkan permainan.

- Skenario 2

Apabila dirasa konfigurasi yang disediakan secara random oleh sistem kurang memuaskan, dapat dilakukan *startGame* untuk mengocok kembali urutan pemain. Atau jika ada data yang ingin diubah/ditambah juga dapat dilakukan dengan *startGame* kembali

### **2.2. Command *cek\_map***

#### **2.2.1. Kegunaan Command**

Command ini digunakan untuk melihat kondisi umum permainan seperti posisi unta saat ini ada dimana saja dan apakah terdapat trap pada petak tertentu. Ini berguna untuk menentukan langkah strategis apa yang harus diambil pemain. Command ini dapat dijalankan sebanyak-banyaknya tanpa ada batasan.

#### **2.2.2. Skenario-Skenario Penggunaan**

- Skenario 1

Apabila pemain yang sedang mendapat giliran bermain dan bingung harus melakukan apa, ia bisa melihat map untuk mengetahui langkah terbaik apa yang bisa dilakukan olehnya

### **2.3. Command *cek\_info***

#### **2.2.1. Kegunaan Command**

Command ini digunakan untuk melihat informasi terkait pemain lain seperti jumlah poin dan banyaknya trap yang dimilikinya. Informasi ini tentu cukup krusial dalam penyusunan strategi bermain. Command ini juga bisa digunakan pada diri sendiri. Command ini dapat dijalankan sebanyak-banyaknya tanpa ada batasan.

#### 2.2.2. Skenario-Skenario Penggunaan

- Skenario 1

Apabila pemain ingin bermain secara kompetitif bisa menggunakan command ini, sebab dengan mengetahui informasi musuh, pemain dapat menggunakan hal tersebut untuk dijadikan leverage bagi dirinya

### 2.4. Command *investasi*

#### 2.2.1. Kegunaan Command

Command ini digunakan para pemain sehingga pemain dapat memilih unta yang diharapkan dapat mendapatkan juara. Fitur ini dapat memberikan poin tambahan di akhir permainan berdasarkan urutan finish unta yang diinvestasikan.

#### 2.2.2. Skenario-Skenario Penggunaan

- Skenario 1

Apabila terdapat satu pemain yang berinvestasi pada suatu unta sedangkan tidak ada pemain lain yang melakukannya, pemain tersebut mendapatkan poin tambahan di akhir permainan berdasarkan urutan finish unta tersebut.

- Skenario 2

Apabila terdapat lebih dari satu pemain yang berinvestasi pada suatu unta, poin tambahan di akhir permainan diberikan berdasarkan urutan finish unta tersebut dan urutan investasi pemain. Dengan kata lain, pemain pertama yang berinvestasi mendapatkan poin tambahan yang lebih banyak dari pemain lain yang berinvestasi setelahnya.

### 2.5. Command *papan\_investasi*

#### 2.2.1. Kegunaan Command

Command ini digunakan untuk melihat data urutan investasi dari setiap unta. Dengan menjalankan program ini, pemain dapat melihat berapa banyak jumlah orang yang berinvestasi pada unta tertentu.

#### 2.2.2. Skenario-Skenario Penggunaan

- Skenario 1

Jika unta belum diinvestasikan oleh siapapun maka petanya akan menampilkan pesan “Belum ada investasi” dan jika sudah maka peta akan menampilkan list investor dalam bentuk tabel.

### 2.6. Command *jalankanUnta*

#### 2.2.1. Kegunaan Command

Command ini digunakan oleh pemain sebagai salah satu opsi *action* ketika mendapatkan giliran. Dengan command ini, pemain akan melempar dua buah dadu yaitu dadu warna dan dadu angka. Warna dari dadu yang terlempar akan menentukan unta mana yang bergerak dan angka dari dadu juga akan menentukan seberapa jauh unta tersebut akan bergerak

#### 2.2.2. Skenario-Skenario Penggunaan

- Skenario 1

Diasumsikan semua unta masih berada di garis *start*. Lalu misalkan, dadu dikocok dan didapatkan dadu warna merah dan dadu angka 3, maka unta berwarna merah akan bergerak menuju ke petak 3.

- Skenario 2

Misalkan dadu merah berada di petak 3 dan dadu dikocok sehingga didapatkan dadu warna merah dengan dadu angka 4. Sedangkan di petak 7 sudah ada unta hijau, maka unta merah akan berada di atas unta hijau.

- Skenario 3

Misalkan terdapat dadu hijau di bawah dadu merah di petak 7. Lalu, dadu dikocok dan didapatkan dadu warna hijau dengan dadu angka 2. Maka unta hijau akan bergerak menuju petak 9 dengan membawa unta merah di atasnya.

- Skenario 4

Misalkan terdapat dadu biru, dadu hijau, dan dadu merah secara berurutan dari bawah ke atas di petak 9. Lalu, dadu dikocok dan didapatkan dadu warna hijau dengan dadu angka 2. Maka unta hijau akan bergerak menuju petak 11 dengan membawa unta merah di atasnya sedangkan unta biru tetap di petak 9.

## **2.7. Command *pasangTrap***

### **2.2.1. Kegunaan Command**

Command ini digunakan apabila pemain yang sedang mendapat giliran ingin memasang trap pada petak tertentu sesuai dengan ketentuan yang berlaku. Dengan harapan trap tersebut akan mengenai seseorang dan sang pembuat trap mendapat tambahan 10 poin lalu unta yang terkena trap akan bergerak maju/mundur sesuai dengan jenis trap. Command ini termasuk kedalam jenis yang hanya dapat dijalankan sekali setiap giliran.

### **2.2.2. Skenario-Skenario Penggunaan**

- Skenario 1

Jika pemain memasukkan trap pada petak yang selangnya  $\leq 1$  ataupun memasukkan jenis trap yang tidak valid (bukan maju/mundur) maka program tidak akan mengizinkan untuk memasang trap di petak tersebut.

- Skenario 2

Jika pemain tidak memiliki trap namun memaksakan untuk memasang trap, maka program juga tidak akan mengizinkan pemain untuk memasang trap tersebut.

## **2.8. Command *nextTurn***

### **2.2.1. Kegunaan Command**

Command ini dijalankan oleh setiap pemain yang sedang mendapatkan giliran setelah melakukan aksinya. Command ini berfungsi sebagai konfirmasi bahwa kesempatan aksi akan berpindah ke pemain selanjutnya sesuai dengan urutan pemain diawal.

### **2.2.2. Skenario-Skenario Penggunaan**

- Skenario 1

Jika pemain sudah melakukan aksi seperti investasi, pasang trap, atau jalankan unta maka hanya cek\_map, cek\_info, dan perintah ini yang dapat dijalankan. Apabila command ini dijalankan maka turn akan berpindah ke pemain selanjutnya.

- Skenario 2

Pemain juga bisa langsung menjalankan perintah ini sebelum menjalankan aksi utama, nantinya pemain ini seperti tidak melakukan aksi pada kesempatannya tersebut.

## **2.9. Command *godsHand***

### **2.2.1. Kegunaan Command**

Command ini tidak dipanggil oleh pemain manapun melainkan akan terpicu sendiri secara random pada setiap turn. Dengan menjalankan perintah ini maka program akan mengambil secara random sebuah unta dan memindahkannya ke petak lain.

### **2.2.2. Kegunaan Command**

- Skenario 1

Apabila dalam suatu petak terdapat unta yang bertumpuk, maka unta yang dipindahkan adalah yang paling atas dari tumpukan.

- Skenario 2

Apabila dalam suatu petak tidak terdapat sama sekali unta, maka tidak ada unta yang dipindahkan sehingga semua unta tetap berada pada posisi saat ini.



# BAB 3: Hasil Eksekusi Program

## 3.1. startGame

```
| ?- startGame.  
Masukkan jumlah pemain: 3.  
  
Masukkan nama pemain 1: agus.  
Masukkan nama pemain 2: alfonso.  
Masukkan nama pemain 3: junaedi.  
  
Urutan pemain: junaedi - agus - alfonso  
  
Setiap pemain mendapatkan 4 kartu, 30 poin dan 1 trap.  
  
Kartu junaedi: merah, kuning, hijau, biru.  
Kartu alfonso: merah, kuning, hijau, biru.  
Kartu agus: merah, kuning, hijau, biru.  
  
Poin junaedi: 30.  
Poin alfonso: 30.  
Poin agus: 30.  
  
Trap junaedi: 1.  
Trap alfonso: 1.  
Trap agus: 1.
```

Gambar 3.1.1 Tampilan dari command startGame

```
Masukkan jumlah pemain: 10.  
Mohon masukkan angka antara 2 - 4.  
Masukkan jumlah pemain:
```

Gambar 3.1.2 Kasus Jumlah Pemain Tidak Valid

```
Masukkan nama pemain 1: iqbal.  
Masukkan nama pemain 2: iqbal.  
Nama sudah terpakai.
```

Gambar 3.1.3 Kasus Nama Pemain Tidak Unik

## 3.2. cek\_map

```
| ?- cek_map.  
Beware of the traps!  


|                   |            |                 |          |          |
|-------------------|------------|-----------------|----------|----------|
| S/F<br>[ ]<br>>>> | A<br>[ ]   | B<br>[ ]<br><<< | C<br>[ ] | D<br>[ ] |
| O<br>[ ]          |            |                 |          | E<br>[ ] |
| N<br>[ ]          | CAMEL POP! |                 |          | F<br>[ ] |
| M<br>[ ]          |            |                 |          | G<br>[ ] |
| L<br>[ ]<br>>>>   | K<br>[ ]   | J<br>[ ]<br><<< | I<br>[ ] | H<br>[ ] |

  
(16 ms) yes
```

Gambar 3.2.1 Tampilan dari command cek\_map

### 3.3. cek\_info

```
| ?- cek_info.  
Masukkan nama pemain: alfonso.  
Pemain alfonso  
Poin : 30  
Kartu : merah, kuning, hijau, biru  
Trap : 1
```

**Gambar 3.3.1 Tampilan dari command cek\_info**

### 3.4. investasi

```
| ?- investasi.  
Pilih warna unta untuk investasi (merah, kuning, hijau, biru): ungu.  
Gagal! ungu bukan warna unta yang valid.  
Silakan pilih warna yang valid (merah, kuning, hijau, biru).
```

**Gambar 3.4.1 Kasus Warna Unta Tidak Valid**

```
| ?- investasi  
Pilih warna unta untuk investasi (merah, kuning, hijau, biru): merah.  
Investasi berhasil! ferdinand telah berinvestasi pada unta merah.  
+-----+  
| INVESTASI UNTA |  
+-----+  
1. ferdinand  
  
true ?
```

**Gambar 3.4.2 Kasus Masukan Valid**

### 3.5. Papan\_investasi

```
Papan investasi pada unta merah  
+-----+  
| INVESTASI UNTA |  
+-----+  
Belum ada Investasi.  
  
Papan investasi pada unta kuning  
+-----+  
| INVESTASI UNTA |  
+-----+  
Belum ada Investasi.  
  
Papan investasi pada unta hijau  
+-----+  
| INVESTASI UNTA |  
+-----+  
1. hatta  
  
Papan investasi pada unta biru  
+-----+  
| INVESTASI UNTA |  
+-----+  
Belum ada Investasi.
```

**Gambar 3.5.1 Papan Investasi**

### 3.6. jalankanUnta

### 3.7.pasangTrap

```
| ?- pasangTrap.  
Masukkan kode petak (KAPITAL dengan kutip satu): 'A'.  
Masukan jenis trap (maju/mundur): maju.  
Trap berhasil dipasang. Unta yang mendarat pada petak tersebut akan maju 1 petak.  
iqbal akan mendapatkan 10 poin ketika ada unta yang mendarat di petak tersebut.  
Sisa trap iqbal: 0.  
true ?
```

***Gambar 3.7.1 Kasus Pasang Trap Berhasil***

```
| ?- pasangTrap.  
Masukkan kode petak (KAPITAL dengan kutip satu): 'Z'.  
Masukan jenis trap (maju/mundur): mandi.  
Trap gagal dipasang.  
Pastikan mempunyai trap, lokasi trap valid, dan jenis trap valid.  
Sisa trap iqbal: 0.  
true ?
```

***Gambar 3.7.2 Kasus Pasang Trap Gagal***

### 3.8.nextTurn

```
| ?- nextTurn.  
no
```

***Gambar 3.8.1 Apabila Memasukkan Perintah nextTurn***

### 3.9.godsHand

## Lampiran

NIM	Persentase Kerja
13523041	<i>Persentase kerja NIM 1</i>
13523051	<i>Persentase kerja NIM 2</i>
13523063	<i>Persentase kerja NIM 3</i>
13523111	<i>Persentase kerja NIM 4</i>

## Referensi

Diaz, D. (n.d.). The GNU Prolog web site. <http://www.gprolog.org/>

Tutorials Point. (2020). Prolog tutorial. Retrieved December 19, 2024, from [https://www.tutorialspoint.com/prolog/prolog\\_tutorial.pdf](https://www.tutorialspoint.com/prolog/prolog_tutorial.pdf)