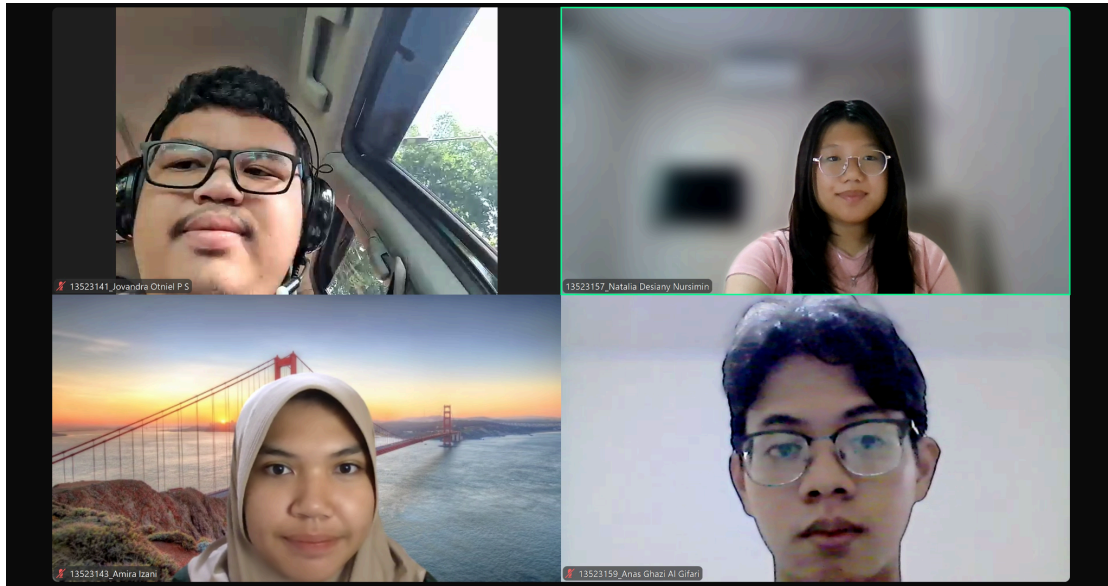


**IF2121 - Logika Komputasional**  
**Tugas Besar**  
**CAMEL UP!**



Disusun Oleh:

*13523141 Jovandra Otniel P.S.*

*13523143 Amira Izani*

*13523157 Natalia Desiany Nursimin*

*13523159 Anas Ghazi Al Gifari*

**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
BAB 3: Hasil Eksekusi Program.....	5
Lampiran.....	6
Referensi.....	7

## BAB 1: Deskripsi Masalah

### 1.1 Topik

Dalam tugas besar ini, kami membuat sebuah board game dengan menggunakan bahasa pemrograman deklaratif Prolog (GNU Prolog). Implementasi tugas besar mengandung materi:

- Rekurens
- List
- Cut
- Fail
- Loop

### 1.2 Tujuan

Tujuan dari tugas besar ini adalah mengkombinasikan berbagai keterampilan dan teknik yang telah dipelajari dalam perkuliahan IF1221 Logika Komputasional, pra-praktikum, dan eksplorasi mandiri mengenai Logika Komputasional dan Prolog.

### 1.3 Timeline

No.	Waktu	Kegiatan
1.	Jumat, 29 November 2024	Pembahasan Pra-Praktikum, Rilis Tugas Besar IF2121.
2.	Rabu, 4 Desember 2024	Batas pengisian daftar kelompok
3.	Jumat, 6 Desember 2024 pukul 12.21	Milestone 1: Asistensi Tugas Besar, pengerjaan, dan pengumpulan progress pertama tugas besar.
4.	Jumat, 13 Desember 2024 pukul 12.21	Milestone 2: Asistensi Tugas Besar, pengerjaan, dan pengumpulan progress kedua dari tugas besar.
5.	Jumat, 20 Desember 2024 pukul 12.21	Milestone 3: Pengumpulan final deliverables Tugas Besar Logika Komputasional.
6.	[menyusul]	Batas pengisian jadwal demo tugas besar.
7.	[menyusul]	Pelaksanaan demo tugas besar.

### 1.4 Alur Permainan

1. Permainan diinisiasi dengan **start game**. *Command* ini akan melakukan *setup* berupa pengacakan urutan pemain serta inisiasi kartu, poin, dan *trap* tiap pemain.

2. Pada tiap giliran pemain, pemain dapat melakukan:

- pilih maksimal 1 aksi di antara: investasi, jalankan unta, pasang trap (juga sembunyikan unta, restart unta, tukar unta, atau God's Hand jika mengerjakan bonus);
- display map dengan jumlah tidak terbatas;
- cek info dengan jumlah tidak terbatas.

Pemain dapat mengakhiri gilirannya dengan menjalankan **end turn**.

3. 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 memulai permainan dan menginisialisasikan pemain beserta kebutuhan mereka.

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

- Permainan belum dimulai

Game mulai berjalan dengan meminta input jumlah pemain yang akan berpartisipasi. Jumlah pemain yang valid adalah antara 2 hingga 4. Jika input tidak valid, program akan meminta input ulang. Selanjutnya, program akan meminta nama-nama pemain sesuai dengan jumlah yang dimasukkan. Jika terdapat duplikat nama atau terdapat huruf pada nama, program akan meminta input ulang. Setiap pemain kemudian akan diberikan empat kartu berwarna merah, kuning, hijau, dan biru, serta satu trap dan 30 poin awal.

- Permainan sudah dimulai

Masukan command tidak valid karena permainan sudah dijalankan sebelumnya.

### **2.2. Command investasi**

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

Perintah ini digunakan untuk melakukan investasi pada unta yang dipilih. Pada tahap akhir permainan (endgame), pemain akan menerima imbalan investasi berdasarkan urutan unta yang telah dipilih.

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

- Permainan sudah berjalan

Program akan meminta pemain pada giliran sekarang untuk memasukkan warna unta yang ingin diinvestasikan.

- Permainan sudah berjalan, masukan warna unta tidak valid

Program akan memberikan pemberitahuan bahwa input warna unta salah, dan meminta pemain untuk memasukkan warna yang valid.

- Permainan sudah berjalan, pemain sudah pernah menginvestasi unta berwarna tersebut

Program akan memberikan pemberitahuan bahwa pemain telah berhasil berinvestasi pada warna unta tersebut, dan kemudian meminta masukan yang valid kembali.

### **2.3. Command papan\_investasi**

#### **2.3.1 Kegunaan Command**

Perintah ini dipanggil untuk menampilkan urutan pemain yang berinvestasi pada tiap unta

#### **2.3.2 Skenario Penggunaan**

- Permainan sudah berjalan

Permainan sudah berjalan Program akan mengeluarkan siapa saja nama-nama pemain yang menginvestasikan unta tertentu. Apabila tidak ada yang menginvestasikan pada unta tertentu. Maka akan muncul tulisan ‘belum ada investasi.

- Permainan belum berjalan

Command ini akan memperlihatkan permainan game sebelumnya atau bila pernah memulai akan inisialisasi belum ada yang investasi.

### **2.4. Command jalankan\_unta**

#### **2.4.1 Kegunaan Command**

#### **2.4.2 Skenario-Skenario Penggunaan**

### **2.5. Command pasang\_trap**

#### **2.5.1 Kegunaan Command**

Command ini dipanggil ketika pemain ingin memasang trap atau jebakan di suatu petak.

### 2.5.2 Skenario-Skenario Penggunaan

- Permainan sudah berjalan

Program akan meminta pemain pada giliran sekarang untuk memasukkan lokasi mana yang ingin dipasang trap, kemudian pemain menentukan apakah ingin memasang trap maju atau mundur.

- Permainan sudah berjalan, terdapat unta pada lokasi pemasangan

Program akan memberikan pemberitahuan bahwa terdapat unta pada lokasi pemasangan trap, sehingga trap tidak dapat dipasang, dan akan meminta pemain untuk memasukkan masukan yang valid.

- Permainan sudah berjalan, terdapat trap lain pada sebelum atau setelah lokasi pemasangan

Program akan memberikan pemberitahuan bahwa terdapat trap lain pada tepat satu petak sebelum atau setelah lokasi pemasangan trap, sehingga trap tidak dapat dipasang, dan akan meminta pemain untuk memasukkan masukan yang valid.

## 2.6. Command `tampilkan_peta`

### 2.6.1 Kegunaan Command

Perintah ini digunakan untuk menampilkan arena bermain, yang berisi lokasi-lokasi tempat unta-unta dan trap berada.

### 2.6.2 Skenario-Skenario Penggunaan

- Permainan sudah berjalan Program akan mengeluarkan siapa saja nama-nama pemain yang menginvestasikan sebuah unta tertentu.

## 2.7. Command `tampilkan_atribut_pemain_sekarang`

## BAB 3: Hasil Eksekusi Program

### 3.1. Command startGame

#### 3.1.1. Melakukan command startGame.

```
| ?- startGame.  
Masukkan jumlah pemain (2-4): |
```

#### 3.1.2. Melakukan tanpa startGame

```
| ?- jalankan_unta.  
uncaught exception: error(existence_error(procedure,jalankan_unta/0),top_level/0)  
| ?- investasi.  
uncaught exception: error(existence_error(procedure,investasi/0),top_level/0)
```

Game tidak akan berjalan karena belum dimulai

#### 3.1.3. Memasukan jumlah pemain

```
| ?- startGame.  
Masukkan jumlah pemain (2-4): 3.  
Masukkan nama pemain 3: |
```

#### 3.1.4. Memasukan jumlah pemain yang bukan angka 2-4

```
Masukkan jumlah pemain (2-4): a.  
uncaught exception: error(type_error(evaluable,a/0),(<)/2)
```

#### 3.1.5. Memasukan nama pemain yang sama

```
Masukkan jumlah pemain (2-4): 2.  
Masukkan nama pemain 2: andi.  
Masukkan nama pemain 1: andi.  
Pemain sudah ada! Silakan masukkan nama lain.
```

#### 3.1.6. Memasukan nama pemain yang bukan huruf

```
Masukkan nama pemain 1: 20.  
uncaught exception: error(type_error(atom,20),atom_chars/2)
```

#### 3.1.7. Memasukan nama pemain yang valid pada game



```
| ?- startGame.
Masukkan jumlah pemain (2-4): 2.
Masukkan nama pemain 2: andi.
Masukkan nama pemain 1: dina.
Urutan pemain: [dina,andi]
Setiap pemain mendapatkan 4 kartu, 30 poin, dan 1 trap.
Kartu dina: [merah,kuning,hijau,biru]
Poin dina: 30
Trap dina: 1
Kartu andi: [merah,kuning,hijau,biru]
Poin andi: 30
Trap andi: 1

Permainan dimulai!

--- Putaran Baru Dimulai ---
Giliran pemain: andi
Pilih aksi Anda: (1) Investasi, (2) Jalankan Unta, (3) Pasang Trap, (4) Tampilkan Peta, (5) Info Pemain, (6) Keluar:
```

### 3.2. Command next\_turn

#### 3.2.1. Melakukan command next\_turn sebelum startGame

```
| ?- next_turn.
uncaught exception: error(existence_error(procedure,next_turn/0),top_level/0)
```

#### 3.2.2. Melakukan command next\_turn sebelum pemain menjalankan sebuah aksi

```
Permainan dimulai!

--- Putaran Baru Dimulai ---
Giliran pemain: andi
Pilih aksi Anda: (1) Investasi, (2) Jalankan Unta, (3) Pasang Trap, (4) Tampilkan Peta, (5) Info Pemain, (6) Keluar: next_turn.
Aksi tidak valid! Silakan coba lagi.
```

### 3.3. Command cek\_info

#### 3.3.1. Melakukan command cek\_info sebelum startGame

```
| ?- cek_info.
uncaught exception: error(existence_error(procedure,cek_info/0),top_level/0)
```

#### 3.3.2. Melakukan command cek\_info

```
Kartu andi: [merah,kuning,hijau,biru]
Poin andi: 30
Trap andi: 1
```

### 3.4. Command displayMap

#### 3.4.1. Melakukan command displayMap sebelum startGame

```

uncaught exception: error(existence_error(procedure,cek_untar/0),top_level/0)
| ?- displayMap.
uncaught exception: error(existence_error(procedure,displayMap/0),top_level/0)

```

### 3.4.2. Melakukan command displayMap

Posisi unta di papan 5x5:

	1	2	3	4	5
1		[UP],[UK],[UB],[UH],[UY],[UM]			
2					
3					
4					
5					

### 3.5. Command investasi

#### Pemain melakukan investasi

Pilih unta untuk diinvestasikan (merah/kuning/hijau/biru): merah.  
 andi berinvestasi pada unta merah.

### 3.6. Command

## Lampiran

NIM	Persentase Kerja
13523141	40 persen
13523143	20 persen
13523157	20 persen
13523159	20 persen

## Referensi

[https://www.tutorialspoint.com/prolog/prolog\\_tutorial.pdf](https://www.tutorialspoint.com/prolog/prolog_tutorial.pdf)

<https://docs.google.com/document/d/1t1URggu2cMrDLkYdedUL6IRXinKrHvML98Yo8rBCnac/edit?tab=t.0#heading=h.gjdgxs>

<http://www.gprolog.org/>