

# **Tugas Besar IF2121 Logika Komputasional**

## **I Quit My Job To Become A Professional Board Game Player**

**“Here’s The Reason Why I Did That. Number 8 Will Surprise You”**



### **Kelompok “ayo dapet kaya!!”**

Kelvin Rayhan Alkarim	13521005
Ditra Rizqa Amadia	13521019
Bernardus Willson	13521021
Raditya Naufal Abiyu	13521022
Kartini Copa	13521026

## Daftar Isi

<b>BAB I</b>	<b>3</b>
<b>Deskripsi Masalah</b>	<b>3</b>
1.1 Deskripsi Masalah	3
<b>BAB II</b>	<b>4</b>
<b>Teori Dasar</b>	<b>4</b>
2.1 Rekurens	4
2.2 List	4
2.3 Cut	5
2.4 Fail	6
2.5 Loop	7
<b>BAB III</b>	<b>8</b>
<b>Command Prolog dan Hasil Eksekusi</b>	<b>8</b>
3.1 Command Prolog	8
3.2 Hasil Eksekusi	9
<b>BAB IV</b>	<b>10</b>
<b>Kesimpulan</b>	<b>10</b>
5.1 Kesimpulan	10
5.2 Saran	11
5.3 Refleksi	11
5.4 Pembagian Tugas	11

## **BAB I**

### **Deskripsi Masalah**

#### **1.1 Deskripsi Masalah**

Steve adalah seorang *programmer* handal yang sudah mempunyai pengalaman setara 10 tahun dalam bidang Prolog, Assembly, NodeJS, TensorFlow, Flutter, Android Studio, dan (*sebutkan saja semua tech-stack yang kamu ketahui*). Dia sangat menyukai pekerjaannya hingga diangkat menjadi CTO (*Chief Technology Officer*) pada PT Gaib Ltd. Steve merasa semua pekerjaan *programming*-nya membosankan karena sudah terlanjur sangat handal di semua bidangnya sehingga ia mencari tujuan lain dalam hidupnya. Suatu ketika, saat Steve sedang melakukan *scrolling* TokTik, ia disuguhkan suatu permainan yang sangat menarik. Permainan ini merupakan *board game* bernama Monopoly. Terkagum-kagum dengan permainan tersebut, Steve segera membentuk tim untuk membuat pengembangan permainan Monopoly. Kebetulan, Anda mencantumkan *skill* yang sangat cocok untuk kebutuhan ini di platform LinkedOut, yaitu menamatkan *course* filosofi Monopoly. Anda langsung segera direkrut oleh Steve sebagai tim pengembang permainan ini agar Steve dapat menjadi pemain Monopoly profesional.

Tugas Anda adalah mengimplementasikan permainan papan Monopoly dengan fitur yang didefinisikan di bawah ini. Buatlah permainan sekreatif mungkin. Kreativitas Anda akan menjadi bagian dari penilaian asisten. Spesifikasi yang diberikan merupakan batas minimum yang harus dikerjakan. Oleh karena itu, sangat direkomendasikan untuk membuat permainan Monopoly yang lebih dari spesifikasi.

## BAB II

### Teori Dasar

#### 2.1 Rekurens

Aturan yang definisinya berdasarkan aturan itu sendiri disebut rekursif. Analisis rekurens harus mendefinisikan aturan basis dan aturan rekurens. Aturan basis merupakan jaminan pencocokan berhenti, sementara aturan rekurens merupakan suatu kondisi harus menuju basis. Analisis rekurens dalam pemrograman deklaratif adalah sebagai berikut.

1. Penalaran berdasarkan definisi predikat yang rekursif,
2. Berdasarkan tipe (simbolik) yang terdefinisi secara rekursif.

Manipulasi dengan analisis rekurens dapat menjumlahkan bilangan integer positif tidak berhingga. Contoh penjumlahan dua buah bilangan integer positif.

*/\* plus(X,Y,Z) benar artinya X ditambah dengan Y hasilnya adalah Z. Jika  $x + y = z$  maka  $(x + 1) + y = z + 1$ . Basis 0: bilangan ditambah 0 hasilnya adalah bilangan itu sendiri \*/*  
*plus(0,X,X).*

*/\* Rekurens: plus(X,Y,Z) menyatakan  $x + y = z$  plus((X+1),Y,(Z+1)) harus dilinearisasi suksesor(X,X') benar jika X' adalah suksesor dari X \*/ plus(X', Y, Z') :- suksesor(X,X'), plus(X, Y, Z), suksesor(Z,Z').*

#### 2.2 List

*List* merupakan sekumpulan elemen dengan keterurutan tertentu yang diketahui elemen pertamanya dan setiap elemen boleh muncul lebih dari satu kali. *List* kosong adalah sebuah *list*, elemen *list* tidak kosong terdiri atas *head* dan *tail*. *Head* adalah elemen *list*; *tail* adalah *list*. Elemen *list* merupakan *term* yang bisa berupa konstanta, variabel, struktur yang mungkin juga berupa *list*. *List* dapat digunakan untuk merepresentasikan *tree*, *grammar*, *mathematical entity*, dll. Penulisan *list* dalam pemrograman deklaratif adalah sebagai berikut.

1. Dituliskan di antara tanda kurung siku
2. Pemisahan antara *head* dan *tail* menggunakan *pipe* (*|*)
- 3.

Contoh *list* sederhana: *list* dengan elemen integer seperti di bawah ini.

*listint([1,2,3]).*

*listint([X|Y]).*

Dari persamaan di atas X merupakan *head* yaitu [1]; Y merupakan *tail* yaitu [2,3].

## 2.3 Cut

Cut merupakan salah satu predikat sistem yang disediakan Prolog. Secara konsep, cut digunakan untuk merealisasikan analisa kasus pada Prolog. Cut digunakan agar eksekusi program lebih efisien tetapi tanpa cut juga dapat dibuat program Prolog yang dapat dieksekusi dengan benar. Proses pencarian solusi dapat dibatasi dengan menggunakan cut. Hal tersebut dimungkinkan dengan cara menghilangkan proses backtrack ke cabang. Berdasarkan tujuan penggunaan, cut dapat dibedakan menjadi dua sebagai berikut.

1. Green Cut: *cut* untuk efisiensi eksekusi, karena jika cut dihilangkan arti program tidak berubah
2. Red Cut: *cut* untuk kebenaran eksekusi, karena jika cut dihilangkan arti program berubah.

Misalnya terdapat *rule* sebagai berikut.

`grade(Mark, a) :- Mark >= 70.`

`grade(Mark, b) :- Mark < 70, Mark >= 63.`

`grade(Mark, c) :- Mark < 63, Mark >= 55.`

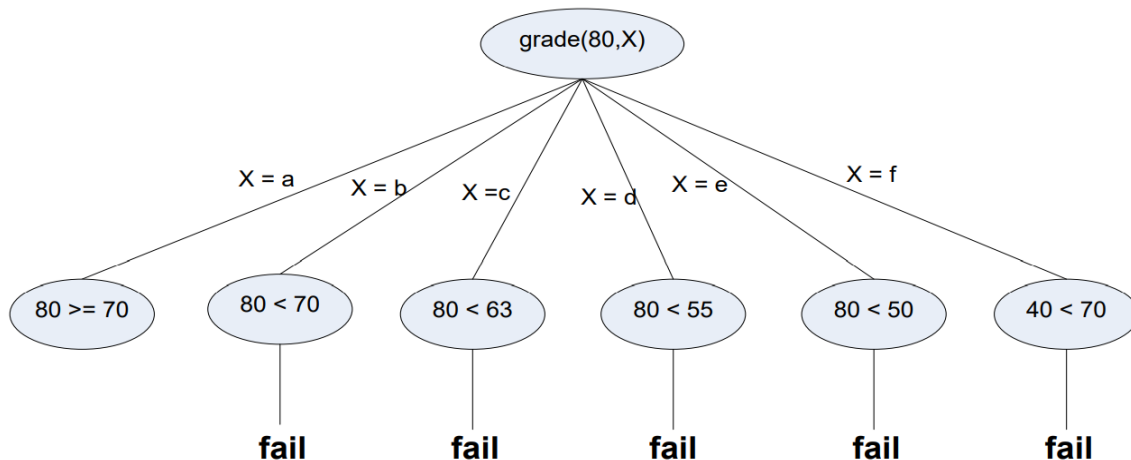
`grade(Mark, d) :- Mark < 55, Mark >= 50.`

`grade(Mark, e) :- Mark < 50, Mark >= 40.`

`grade(Mark, f) :- Mark < 40.`

*Query:* `grade(80, X)`

Proses eksekusi program ditunjukkan pada gambar 2.1.1 berikut.



**Gambar 2.1.1** Pohon Eksekusi tanpa cut

Program akan menghasilkan  $X = a$

Sedangkan *rule* dengan penggunaan *cut* sebagai berikut.

$\text{grade}(N, a) :- N \geq 70, !.$

$\text{grade}(N, b) :- N \geq 63, !.$

$\text{grade}(N, c) :- N \geq 55, !.$

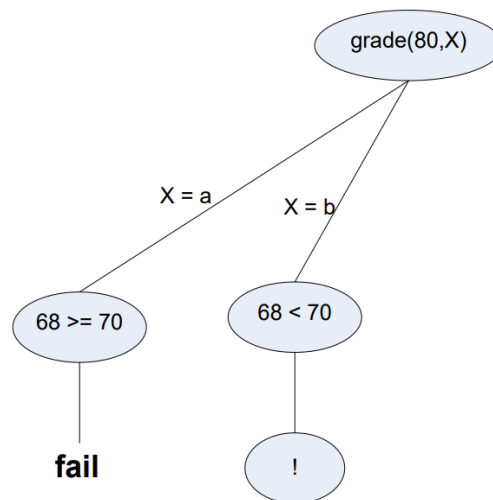
$\text{grade}(N, d) :- N \geq 50, !.$

$\text{grade}(N, e) :- N \geq 40, !.$

$\text{grade}(N, f) :- N < 40.$

*Query*:  $\text{grade}(68, X).$

Proses eksekusi program dengan penggunaan *cut* ditunjukkan pada gambar 2.1.2 berikut.



**Gambar 2.1.2** Pohon eksekusi dengan *cut*

Program menghasilkan nilai  $X = b$ . Setelah bertemu karakter *cut*(!), program tidak akan melakukan *backtrack*. Sehingga tidak perlu diperiksa untuk nilai  $X = d, e, f$ .

## 2.4 Fail

Fail merupakan salah satu predikat yang disediakan prolog. Secara konsep, Fail digunakan untuk memanipulasi program sehingga suatu proses tertentu harus dilakukan. Tetapi, jika dikombinasikan *cut* dan *fail*, secara konsep, digunakan untuk merealisasikan *STOP statement* pada Prolog, artinya untuk memaksa program Prolog berhenti pada satu kasus tertentu. Fail umumnya digunakan untuk merealisasikan *Error Message*. Dengan Fail, Prolog justru memaksa proses *backtrack* ke suatu cabang tertentu. Misalnya terdapat *rule* sebagai berikut.:

$Z :- A.$

$A :- B, \text{fail}.$

$A :- C$

Jika predikat Z dieksekusi, maka selanjutnya akan memproses predikat A. Pada saat A dieksekusi dengan menelusuri kalimat (1), jika B bernilai *true* maka (1) akan digagalkan atau bernilai *false*. Selanjutnya Prolog akan menelusuri kalimat (2), dan hasilnya tergantung pada C.

## 2.5 Loop

Loop merupakan suatu kondisi yang secara menerus diulang hingga suatu kondisi tertentu tercapai. Pada Prolog digunakan ‘repeat’ untuk melakukan loop. Berikut merupakan contoh implementasi loop pada Prolog.

```
command_loop:- write('Selamat bermain di AyoDapetKaya!! Search'), nl,
    repeat, write('>main> '),
    read(X), do(X), nl,
    end_condition(X).
end_condition(end).
end_condition(X) :- have(X),!,
    write('Selamat').
do(X):- have(X),!.
do(end).
do(_):- write('Invalid Command').
have(X):- X==main,!.
```

## BAB III

### Command Prolog dan Hasil Eksekusi

#### 3.1 Command Prolog

Pada permainan monopoly yang para penulis buat, umumnya command yang bisa digunakan tersebar ke beberapa file Prolog sesuai dengan nama command. Hal ini dimaksudkan untuk mempermudah melakukan perancangan permainan. Pemanggilan command dilakukan dengan menggunakan pemanggilan dari gabungan predikat-predikat yang terdapat dari file lainnya.

##### 1. start

*Command* 'start' merupakan inisiasi permainan monopoly untuk memulai permainan. Terdapat *dynamic(isOngoing/1)* untuk memastikan permainan dimulai dengan 'start'. Setelah melakukan kompilasi terhadap file 'main.pl' maka dengan memasukkan *command* 'start' akan muncul *splash screen*. Inisiasinya terdapat dua pemain yaitu, *player v* dan *player w* yang masing-masing memiliki *cash* sebesar dua puluh ribu (20000).

##### 2. help

*Command* 'help' digunakan untuk melihat informasi mengenai *command* apa saja yang dapat digunakan pada program. Command ini dapat digunakan dimanapun dan dalam waktu apapun. Sama seperti *command* lainnya, *command* 'help' baru dapat digunakan ketika program sudah berjalan, jadi harus melakukan start terlebih dahulu baru bisa memanggil command ini. Penggunaannya tinggal memanggil command 'help.' maka semua informasi mengenai command yang terdapat pada program akan muncul.

##### 3. map

*Command* 'map' digunakan untuk melihat informasi peta atau lokasi pada program. *Command* ini dapat diakses ketika permainan sudah dimulai, dan dapat diakses kapanpun dan dimanapun pemain berada. *Command* ini dipanggil dengan memanggil 'map.' pada program, lalu akan ditampilkan peta dan posisi pemain. *Map* akan selalu *ter-update* setiap pemain melakukan *move* ke posisi lain.

##### 4. playerdetail

*Command* 'playerdetail' digunakan untuk melihat informasi rinci pemain. Jika *input player* yang diberikan valid maka akan keluar tampilan informasi pemain yang dimaksud. Informasi tersebut berupa lokasi terkini pemain, total uang yang dimiliki pemain, total nilai properti yang dimiliki pemain, dan daftar kepemilikan kartu pemain.



5. payRent

*Command* 'payRent' digunakan saat pemain berada di properti yang telah diakuisisi oleh pemain lain yang mengharuskan pemain membayar sewa sesuai dengan tingkatannya properti. Selanjutnya, pemain boleh memilih untuk mengakuisisi properti tersebut dengan membayar biaya akuisisi kepada pemilik semula sejumlah dua kali nilai aset properti saat ini dengan syarat properti belum memiliki *landmark*, kemudian memilih untuk melakukan peningkatan properti atau tidak.

6. buyProperty

*Command* 'buyProperty' digunakan untuk membeli properti pada permainan monopoly. *Command* ini dapat dilakukan jika pemain berada di lokasi yang valid pada *map*.

7. payTax

*Command* 'payTax' digunakan sebagai penalti untuk membayar pajak apabila pemain berada di posisi 'TX' pada *map*. Pemain yang mencapai daerah 'TX' wajib membayar pajak sejumlah 10% dari total aset yang dimilikinya. Aset tersebut termasuk uang yang dimiliki dan semua nilai aset properti yang dimiliki.

8. sellProperty

*Command* 'sellProperty' digunakan saat pemain kekurangan uang untuk membayar sewa. Jika uang yang dipegang pemain tidak cukup, tetapi pemain memiliki kekayaan properti yang bisa dijual (dan cukup untuk melunasi hutang), pemain dapat memilih untuk menjual propertinya untuk melanjutkan permainan atau menyerah (*surrender*). Jika memilih *command* 'sellProperty' maka hasil penjualan properti adalah 80% dari harga beli. Pemain hanya dapat menjual *tile* atau kotak secara keseluruhan.

9. chanceCard

*Command* 'chanceCard' merupakan kondisi pemain bisa mendapatkan keuntungan atau kerugian. Terdapat empat 'chanceCard' yaitu 'Kartu Tax', 'Kartu Hadiah', 'Kartu Get Out From Jail', dan 'Kartu Go To Jail'. Jika pemain mendapat 'Kartu Tax' maka pemain harus membayar pajak. Jika pemain mendapat 'Kartu Hadiah' maka pemain akan memperoleh hadiah berupa uang. Jika pemain mendapat 'Kartu Get Out From Jail' maka kartu tersebut dapat digunakan nantinya oleh pemain untuk keluar dari penjara. Jika pemain dapat 'Kartu Go To Jail' maka pemain akan berpindah lokasi ke 'jl' pada *map* yang artinya pemain terjebak dalam penjara selama tiga giliran.

10. Bonus Chance Card

Bonus ini merupakan bagian dari *command* 'chanceCard' yaitu 'Kartu Go To World Tour'. Jika mendapat kartu ini, pemain berkesempatan pergi ke kotak 'WT'.

#### 11. bankrupt

*Command* 'bankrupt' merupakan kondisi suatu pemain tidak lagi mempunyai total kekayaan (uang yang dipegang + total kekayaan properti) yang cukup jika dikenai biaya denda atau biaya sewa. Jika uang yang dipegang pemain tidak cukup, tetapi pemain memiliki kekayaan properti yang bisa dijual (dan cukup untuk melunasi hutang), pemain dapat memilih untuk menjual propertinya untuk melanjutkan permainan atau menyerah (*surrender*). Jika pemain memilih menyerah maka pemain akan kalah dan permainan berakhir.

#### 12. worldTour

*Command* 'worldTour' merupakan kesempatan pemain untuk pergi kemana saja pada map kecuali ke kotak World Tour. Jika pemain mendarat di kotak "World Tour", maka pemain akan diberikan kesempatan untuk berpindah ke lokasi manapun di map kecuali kotak World Tour (WT). Perpindahan pemain tidak menggunakan konsep teleportasi atau langsung dapat berpindah dari kotak World Tour ke kotak tujuan akhir tetapi berjalan melewati kotak-kotak lainnya. Artinya, untuk mencapai kotak tujuan pemain harus melewati titik GO, maka pemain akan mendapatkan uang sesuai dengan nominal yang telah ditentukan. Pemain yang ingin menggunakan World Tour harus membayar sejumlah nominal.

#### 13. locDetail

*Command* 'locDetail' digunakan untuk melihat informasi mengenai suatu lokasi secara rinci. Terdapat dua jenis lokasi, yakni properti dan non-properti. Deskripsi informasi untuk lokasi properti menampilkan nama lokasi, deskripsi lokasi, kepemilikan lokasi, biaya sewa untuk yang menghampiri, biaya akuisisi (selain *landmark*), dan tingkatan properti saat ini. Jika belum ada yang memiliki properti tersebut, akan ada tampilan bahwa properti belum dimiliki oleh siapapun. Sedangkan deskripsi untuk lokasi non-properti, informasi yang ditampilkan antara lain nama lokasi dan deskripsi yang menjelaskan spesifikasi lokasi berkaitan.

#### 14. propDetail

*Command* 'propDetail' digunakan untuk melihat informasi mengenai suatu properti. Jika *input* kode properti valid maka akan ditampilkan deskripsi properti berupa biaya pembelian dan biaya sewa. Jika *input* kode properti tidak valid akan diminta ulang kode properti yang valid.

#### 15. Bonus Flip Coin

Bonus 'Flip Coin' merupakan *minigame* berupa *coin flip*. Tujuannya adalah pemain mendapatkan kesempatan untuk memperoleh uang. Implementasi dari bonus ini adalah dengan mengganti salah satu kotak pada papan menjadi kotak *coin flip*. Pemain yang menginjak kotak ini dapat memainkan *mini game* berupa *coin flip* dengan aturan pemain menebak koin yang dilempar (*head/tail*). Jika pemain menebak dengan benar, pemain dapat menebak untuk berikutnya sampai maksimal 3 kali. Jika salah, *coin flip* diakhiri. Di akhir *minigame*, pemain diberikan uang yang sebanding dengan jumlah tebakan yang benar.

### 3.2 Hasil Eksekusi

1. start

[illegible]

**Gambar 3.2.1** Tampilan *command* ‘start’

## 2. help

```
| ?- help.
=====
      Commands
-----

1. map. -- Show map
2. help. -- Show available commands
3. checkPlayerDetail(Player). -- Show player detail
3. checkLocationDetail(Location). -- Show location detail
3. checkPropertyDetail(Property). -- Show property detail
4. buy. -- Buy or upgrade property
5. roll. -- Roll the dice
6. endTurn. -- End the turn
7. checkTurn. -- Show turn

=====

(47 ms) yes
| ?-
```

**Gambar 3.2.2** Tampilan *command* ‘help’

## 3. map

```
| ?- map.
```

```
MAP
```

FP	E1	E2	E3	CC	F1	F2	F3	WT
D3								G1
D2								G2
D1								G3
TX	M O N O P O L Y							TX
C3								CC
C2								H1
C1								H2
JL	B3	B2	B1	CC	A3	A2	A1	GO

```
Posisi pemain:
```

```
V = GO
```

```
W = GO
```

```
(62 ms) yes
```

```
| ?-
```

**Gambar 3.2.3** Tampilan *command* 'map'

## 4. playerDetail

Pergerakan pemain pada papan permainan

Anda maju sebanyak 4 langkah.

```
(16 ms) yes
```

```
| ?- map.
```

```
MAP
```

FP	E1	E2	E3	CC	F1	F2	F3	WT
D3								G1
D2								G2
D1								G3
TX	M O N O P O L Y							TX
C3								CC
C2								H1
C1								H2
JL	B3	B2	B1	CC	A3	A2	A1	GO

```
Posisi pemain:
```

```
V = C2
```

```
W = FP
```

**Gambar 3.2.4** Pergerakan pemain

## 5. Pelemparan dadu dan implementasi 'double'

```

?~ roll.
Sekarang adalah giliran pemain V

  o o
  o o

  o

Anda maju sebanyak 5 langkah.
Anda berada di properti kosong (f2)

yes
?~ roll.

-----
Maaf, kesempatan roll dadu sudah habis!
-----

ves

```

**Gambar 3.2.5** Pelemparan dadu biasa

```

?~ roll.
Sekarang adalah giliran pemain V

  o

  o

Anda maju sebanyak 2 langkah.
Anda berada di properti kosong (a2)

yes
?~ endTurn.

-----
Anda masih punya kesempatan roll dice!
-----

```

**Gambar 3.2.6** Pelemparan dadu 'double'

## 6. Pembelian properti pada papan permainan

```

?~ buy.

-----
Beli Property
-----

Property D2 - Cibiru:

1. Tanah      : 470
2. Bangunan 1 : 200
3. Bangunan 2 : 600
4. Bangunan 3 : 1000
5. Landmark   : Beli bangunan 3 untuk membangun Landmark

-----

Pilih properti yang akan dibeli: 4

-----
Properti berhasil dibeli!
-----

Bangunan 3 D2 - Cibiru
berhasil dibeli.
-----

```

**Gambar 3.2.7** Pembelian properti

```

| ?- buy.
-----
Beli Property
-----

Property H2 - Jatinangor:
    1. Tanah      : 1000
    2. Bangunan 1 : 400
    3. Bangunan 2 : 1200
    4. Bangunan 3 : 2000
    5. Landmark   : Beli bangunan 3 untuk membangun Landmark

-----

Pilih properti yang akan dibeli: 4
-----
Properti gagal dibeli!
-----

Uang Anda tidak cukup untuk membeli properti ini!
-----

```

**Gambar 3.2.8** Pembelian properti uang tidak cukup

## 7. Akuisisi properti oleh lawan/pemain

```

| ?- buy.
-----
Beli Property
-----

Property E2 - Cinunuk:
Property ini dimiliki Player v.
Harga properti ini menjadi
2850 x 2 = 5700

-----

Apakah Anda yakin? (yes/no): yes
-----
Beli Property
-----

Property berhasil diakuisisi!
-----

yes

```

**Gambar 3.2.9** Akuisisi properti

## 8. Penambahan uang pada pemain saat melewati kotak 'GO'

```

yes
| ?- roll.
Sekarang adalah giliran pemain W

  +-----+
  |   o   o   |
  |   o   o   |
  +-----+

  +-----+
  |   o o o   |
  +-----+

Anda maju sebanyak 7 langkah.
Anda telah menyelesaikan satu putaran!
Anda mendapatkan uang tambahan sebesar 2000!

Anda berada di properti kosong (b2)

```

**Gambar 3.2.10** Penambahan uang saat lewat kotak 'GO'

## 9. Mekanisme pemain yang berada di dalam penjara

Anda maju sebanyak 8 langkah.

Go to Jail Card

Anda masuk penjara :(

**Gambar 3.2.11** Pemain masuk penjara

```
1 7- roll.
Sekarang adalah giliran pemain V
ANDA SEDANG DI PENJARA
Pilih cara anda keluar dari penjara(kartu, roll, bayar)
kartu
Anda tidak memiliki kartu keluar dari penjara!
Pilih cara anda keluar dari penjara(kartu, roll, bayar)
roll
```



(78 ms) yes  
| ?- map.

MAP

[illegible]

Posisi pemain:  
V = JL  
W = TX

**Gambar 3.2.12 Mekanisme keluar penjara**

## 10. *Chance card*

Anda maju sebanyak 4 langkah.

=====

Tax Card

Oh tidak, Anda terkena pajak.

Anda terkena pajak!

Anda harus membayar pajak sebesar 2000

```
=====
Berhasil membayar pajak!
```

**Gambar 3.2.13** *Chance card* ‘Kartu Tax’

```

Anda maju sebanyak 2 langkah.
=====
Gift Card
=====

Anda mendapatkan hadiah uang sebesar 3294.
=====

```

**Gambar 3.2.14** *Chance card* ‘Kartu Hadiah’

```

Anda maju sebanyak 5 langkah.
=====
Get Out From Jail Card
=====

Kartu ini dapat dipakai untuk keluar dari penjara.
=====

```

**Gambar 3.2.15** *Chance card* ‘Kartu Out To Jail’

```

Anda maju sebanyak 8 langkah.
=====
Go to Jail Card
=====

Anda masuk penjara :(
=====

```

**Gambar 3.2.16** *Chance card* ‘Kartu Go To Jail’

## 11. Implementasi world tour

```

=====
Anda berada di World Tour!
=====

Anda bisa pergi kemana saja!
=====

MAP

      W3
      |-----|
      | FP | E1 | E2 | E3 | CC | F1 | F2 | F3 | WT |
      | D3 |-----| G1 |
      | D2 |-----| G2 |
      | D1 |-----| G3 | W3
      | TX |-----| TX |
      | C3 |-----| CC |
      | C2 |-----| H1 |
      | C1 |-----| H2 |
      | JL | B3 | B2 | B1 | CC | A3 | A2 | A1 | GO |
      |-----|
      W3

Posisi pemain:
V = WT
W = G3
Please choose where you wanna go!!(no capital): |

```

**Gambar 3.2.17** World Tour



## 12. Implementasi Tax

```
Anda maju sebanyak 2 langkah.
-----
Anda terkena pajak!
-----
Anda harus membayar pajak sebesar 1474
-----
-----
Berhasil membayar pajak!
-----
```

**Gambar 3.2.18** Perpajakan

## 13. Implementasi Sewa

```
Anda maju sebanyak 6 langkah.
-----
Properti ini dimiliki V
-----
Anda mendarat di E2-Cinunuk dan dikenakan biaya
sewa sebesar 3700.
-----
-----
Bayar Sewa
-----
Berhasil membayar sewa!
-----
```

**Gambar 3.2.19** Pembayaran sewa

```
Anda maju sebanyak 5 langkah.
-----
Properti ini dimiliki V
-----
Anda mendarat di E1-Cileunyi dan dikenakan biaya
sewa sebesar 3700.
-----
-----
Uang Anda tidak cukup!
-----
Anda harus menjual properti untuk membayar sewa.
-----
Apakah Anda ingin melanjutkan? (yes/no):
```

**Gambar 3.2.20** Pembayaran sewa uang tidak cukup

## 14. Mekanisme ‘Bangkrut’ dan ‘Menang’

```

=====
BANGKRUT!
=====

Oh tidak! Anda bangkrut :(

=====

Gambar 3.2.21 Pemain bangkrut

      0 000000000 0 0000      .0. 0. 0000.      0 0 000000000 0 000000000.      0. 000b      0
      0 0000 00. 0 0000      .000. 0. 0000.      0 0000 0 0000 00.      0. 000b      0
      0 0000 00 0 0000      :00000. 0. 0000.      0 0000 0 0000 00      0. 000b      0
      0 0000 .00 0 0000      .00000. 0. 0000.      0 0000 0 0000 .00      0. 000b      0
      0 0000. .00 0 0000      .0. 00000. 0. 00000      0 00000000000 0 0000. .00      0. 000b      0
      0 000000000P 0 0000      .0 0. 00000. 0. 00000      0 0000 0 000000000P      0. 000b      0
      0 0000 0 0000      .0 0. 00000. 0 0000      0 0000 0 0000 0b      0. 000b      0
      0 0000 0 0000      .0 0. 00000. 0 0000      0 0000 0 0000 0b.      0. 000b      0
      0 0000 0 0000      .000000000. 00000. 0 0000      0 0000 0 0000 0b.      0. 000b      0
      0 0000 0 0000000000 .0      0. 00000. 0 0000      0 00000000000 0 0000      00.      0.

```

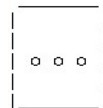
```

      .0.      .0.      0 0000000000 b.      0      .0.      b.      0      00000000.
      .000.      .000.      0 0000      0000.      0      .000.      0000.      0      0000      00.
      .0000.      .0000.      0 0000      Y000000.      0      :00000.      Y000000.      0 0 0000      0.
      .0. 0000. .0. 0000.      0 0000      . Y0000000.      0      . 00000.      . Y0000000.      0 00 0000
      .0 0. 00000.0^0. 0000.      0 00000000000000 0o. Y0000000. 0      .0. 00000.      0o. Y0000000. 0 00 0000
      .0 0. 0000 0. 0000.      0 0000      0 Y0o. Y0000000      .0 0. 00000.      0 Y0o. Y0000000 00 0000
      .0 0. 00      0. 0000.      0 0000      0 Y0o. Y0000      .0 0. 00000.      0 Y0o. Y0000 00 0000      0000000
      .0 0. 00      0. 0000.      0 0000      0 Y0o. Y0      .0 0. 00000.      0 Y0o. Y0 0 0000      0
      .0 0      0. 0000.      0 0000      0 Y0o. Y0      .000000000. 00000. 0      Y0o.      0000      00
      .0      0. 0000. 0 00000000000 0      Y0o.      0. 00000. 0      Y0o      000000P

```

**Gambar 3.2.22** Pemain menang

## 15. Bonus Flip Coin



Anda maju sebanyak 4 langkah.  
Anda telah menyelesaikan satu putaran!  
Anda mendapatkan uang tambahan sebesar 2000!

Selamat Datang di BONUS GAME!  
BONUS GAME kali ini adalah FLIP COIN, anda harus menebak HEAD atau TAIL.  
Setiap tebakan benar akan diberi hadiah sebesar 1000, maks jawaban benar adalah 3x.

Untuk koin pertama, apa tebakan anda (1 untuk HEAD, 2 untuk TAIL): 1  
Selamat Anda berhasil menebak lemparan koin pertama! Lanjutttt...

Untuk koin kedua, apa tebakan anda (1 untuk HEAD, 2 untuk TAIL): 2  
Maaf, anda kurang beruntung! Anda mendapat uang sebesar 1000.

```
(31 ms) yes
| ?- checkPlayerDetail(v).
```

```
=====
Informasi Player V
=====
```

```

Lokasi      : BG
Total Uang   : 23000
Total Nilai Properti : 0
Total Aset   : 23000

```

```
Daftar Kepemilikan Properti :
```

```
Daftar Kepemilikan Card :
```

```
=====
```

**Gambar 3.2.23** Bonus Flip Coin

## 16. Bonus Chance Card

```

?~ roll.
Sekarang adalah giliran pemain V
ANDA SEDANG DI PENJARA
Pilih cara anda keluar dari penjara(kartu, roll, bayar)
kartu
Anda tidak memiliki kartu keluar dari penjara!
Pilih cara anda keluar dari penjara(kartu, roll, bayar)
roll

```



```

(78 ms) yes
?~ map.

```

MAP

		W1					W1				
V3	FP	E1	E2	E3	CC	F1	F2	F3	WT		
	D3								G1		
	D2								G2		
	D1								G3		
	TX	M O N O P O L Y							TX		
	C3								CC		
	C2								H1		
	C1								H2		
	JL	B3	B2	B1	CC	A3	A2	A1	GO		

```

Posisi pemain:
V = JL
W = TX

```

Gambar 3.2.24 Bonus 'Kartu Go To World Tour'

## **BAB IV**

### **Kesimpulan**

#### **5.1 Kesimpulan**

Tugas besar IF2121 Logika Komputasional, membuat permainan papan monopoly sudah selesai dan sesuai spesifikasi yang diberikan. Para penulis mengimplementasikan materi-materi yang telah dipelajari di kelas IF2121 Logika Komputasional. Hal-hal yang diimplementasikan pada game ini adalah penggunaan bahasa prolog yang memenuhi beberapa materi, yaitu rekurens, list, cut, fail, dan loop. Semua materi ini diimplementasikan pada program dan dapat dijalankan sesuai dengan yang diharapkan. Bonus juga diimplementasikan dengan baik pada program ini sehingga dapat dikatakan program ini sudah terimplementasikan dengan baik.

#### **5.2 Saran**

Penggunaan bahasa Prolog di masa kini merupakan hal yang tidak lazim bagi beberapa orang, termasuk bagi para penulis. Oleh karena itu, eksplorasi lebih jauh dan dalam mengenai bahasa Prolog sangat dianjurkan sebelum memulai tugas besar ini agar menghindari terjadinya *syntax error* yang tidak diharapkan. Selain itu, dalam pengerjaan tugas besar ini, para penulis menyarankan untuk menuangkan kreativitas yang lebih dalam pengerjaan.

#### **5.3 Refleksi**

Tugas besar IF2124 Logika Komputasional merupakan suatu tantangan menarik bagi para penulis. Mempelajari bahasa prolog merupakan hal yang menarik karena dapat meningkatkan *skill* dan juga pemahaman akan suatu bahasa pemrograman. Setelah mengerjakan tugas besar ini, timbul rasa apresiasi dari para penulis kepada semua pihak yang terlibat dalam pengerjaan tugas besar ini. Mulai dari pemberian materi hingga pemberian spesifikasi tugas yang diberikan. Selain itu, para penulis juga menyadari bahwa dengan mengerjakan tugas besar ini, para penulis mulai bisa mengembangkan permainan sederhana dan mungkin dapat lebih dikembangkan lagi di masa depan. Hal yang paling penting lagi dalam pengerjaan tugas besar ini adalah kami dapat memperbaiki kinerja kami dalam berkelompok mulai dari pembagian tugas hingga *timeline* pengerjaan sehingga semuanya dapat selesai sesuai dengan yang diharapkan.

## 5.4 Pembagian Tugas

Nama	NIM	Pembagian tugas	Persentase
Kelvin Rayhan Alkarim	13521005	locDetail.pl, debug buyProperty.pl, laporan,	17%
Ditra Rizqa Amadia	13521019	playerdetail.pl, payRent.pl, sellProperty.pl, help.pl, buyProperty.pl, chanceCard.pl	22%
Bernardus Willson	13521021	map.pl, help.pl, dice.pl, debug map.pl, debug dice.pl, bonus Flip Coin, bonus chance card	22%
Raditya Naufal Abiyu	13521022	locDetail.pl, taxApp.pl, worldTour.pl, propDetail.pl, jail.pl, dice.pl	22%
Kartini Copa	13521026	locDetail.pl, freeParkingApp, update Readme.md, laporan	17%

## REFERENSI

Slide kuliah IF2121 Logika Komputasional Tahun 2022/2023