

**LAPORAN
TUGAS BESAR
IF2121 LOGIKA KOMPUTASIONAL**



Oleh:

Kelompok “Boku no Prolog”

Naufal Syifa Firdaus	13521050
Muhammad Fadhil Amri	13521066
Akmal Mahardika N. P.	13521070
Razzan Daksana Yoni	13521087
Muhamad Aji Wibisono	13521095

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2022**

DAFTAR ISI

DAFTAR ISI.....	2
BAB I PENDAHULUAN.....	5
1.1 Topik	5
1.2 Domain Permasalahan.....	5
BAB II PENJELASAN COMMAND	6
A. Command di Luar Game.....	6
2.1 startGame	6
2.2 inputHandling.....	6
2.3 printMap	7
2.4 throwDice	7
2.5 moveToLocation(Pemain, Lokasi).....	7
2.6 checkLokasi(Pemain)	8
2.7 changeBalance(Pemain, Balance)	8
2.8 addPosession(Pemain, Property, Level).....	8
2.9 removePosession(Pemain, Property).....	8
2.10 checkBangkrut(Pemain)	9
2.11 resetGame.....	9
2.12 checkLocationDetail(X)	9
2.13 checkPropertyDetail(X)	9
2.14 checkPlayerDetail(X)	10
2.15 playGameCenter.....	10
2.16 guessing_flow.....	10
2.17 inputGuessing.....	10
2.18 useJailCard	11
2.19 suapSipir.....	11

2.20 runCard.....	11
B. Command di Dalam Game	12
2.1 lempar.....	12
2.2 locationDetail	12
2.3 propertyDetail.....	12
2.4 playerDetail	12
2.5 useJailCard	13
2.6 suapSipir.....	13
2.7 help.....	13
2.8 debug	14
BAB III EKSPERIMEN	15
3.1 startGame	15
3.2 inputHandling.....	15
3.3 checkLocationDetail(X)	16
3.4 checkPropertyDetail(X)	16
3.5 checkPlayerDetail(X)	17
3.6. playGameCenter.....	17
3.7 useJailCard	17
3.8 suapSipir.....	17
3.9 Landing di properti kosong	18
3.10 Landing di properti sendiri.....	19
3.11 Landing di properti pemain lain	20
3.12 World Tour.....	21
3.13 Melewati dan mendarat di go	21
3.14 Tax.....	22
3.15 Chance Card	23
3.16 checkBangkrut dan resolveBangkrut	23
3.17 World Cup.....	24

3.18 Dadu dan double.....	25
3.19 Color Set.....	25
3.20 Debug	25
3.21 ResetGame	26
BAB IV	27
KESIMPULAN.....	27
LAMPIRAN.....	28

BAB I

PENDAHULUAN

1.1 Topik

Membuat permainan papan Monopoly untuk seorang *programmer* yang beralih pekerjaan dengan menggunakan bahasa pemrograman deklaratif GNU Prolog. Implementasi yang dibuat mengandung rekurens, *list*, *cut*, *fail*, dan *loop*.

1.2 Domain Permasalahan

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.

BAB II

PENJELASAN COMMAND

Pada pengerjaan tugas ini penulis memilih untuk membuat program yang menerima input terpisah dari terminal prolog. Tujuannya agar pemain dapat menjalankan permainan tanpa memanggil fungsi – fungsi yang dapat mengganggu keberjalanan permainan seperti mengubah uang atau properti secara langsung. Dengan demikian, pada bagian penjelasan command ini penulis akan membagi command yang ada menjadi dua bagian, yaitu command di luar game yang merupakan command inti yang mengontrol keberjalanan game namun tidak terpanggil langsung pada game, dan command di dalam game yang merupakan himpunan command terbatas yang diatur oleh suatu sistem input handling.

Pada laporan ini penulis hanya menuliskan command inti dan tidak menuliskan semua command yang terdapat pada program karena jumlahnya yang cukup banyak. Penulis juga tidak akan menuliskan semua command internal yang merupakan bagian dari rekursif.

A. Command di Luar Game

2.1 startGame

2.1.1 Kegunaan Command

Command ini digunakan untuk memulai permainan.

2.1.2 Skenario Penggunaan

Ketika command ini dijalankan, akan diinisialisasi loop pada permainan utama untuk input handling dan interface secara umum. Jika terjadi error atau input yang menyebabkan program keluar dari loop utama, command ini akan menyimpan permainan yang telah berlangsung.

2.2 inputHandling

2.2.1 Kegunaan Command

Command ini digunakan untuk membatasi perintah yang dapat dipanggil pemain di dalam keberjalanan program

2.2.2 Skenario Penggunaan

Ketika command ini dijalankan, akan diberikan suatu prompt yang menerima input dari pengguna dan mengarahkannya pada command di luar program yang bersangkutan dari input tersebut.

2.3 printMap

2.3.1 Kegunaan Command

Command ini digunakan untuk mencetak keseluruhan papan pada layar.

2.3.2 Skenario Penggunaan

Command digunakan untuk melihat kondisi papan. Pada program utama dilakukan hampir di setiap aksi yang dilakukan.

2.4 throwDice

2.4.1 Kegunaan Command

Command ini digunakan pemain untuk melempar dadu ketika pemain dapat giliran.

2.4.2 Skenario Penggunaan

Ketika command digunakan, jika pemain sedang menjadi player yang mendapatkan giliran dan akan melempar dadu, jika dadu double pemain dapat melempar lagi, dan jika sudah tiga kali double pemain akan dimasukkan ke dalam jail

2.5 moveToLocation(Pemain, Lokasi)

2.5.1 Kegunaan Command

Command ini digunakan untuk menggerakkan pemain sampai suatu petak yang diberikan. Penggerakan dilakukan dengan melalui tiap petak.

2.5.2 Skenario Penggunaan

Command digunakan untuk chance card dan world tour untuk berpindah tempat dengan per petak.

2.6 checkLokasi(Pemain)

2.6.1 Kegunaan Command

Command ini digunakan untuk mendeteksi petak pendaratan seorang pemain dan menentukan aksi yang perlu dilakukan sesuai dengan jenis dan kondisi petak tersebut.

2.6.2 Skenario Penggunaan

Dalam permainan utama, command selalu dipanggil setelah dilakukan pelemparan dadu.

2.7 changeBalance(Pemain, Balance)

2.7.1 Kegunaan Command

Command ini digunakan untuk mengubah uang yang dimiliki seorang pemain

2.7.2 Skenario Penggunaan

Command ini digunakan jika ada perubahan pada keuangan pemain. Dibuat juga command penyederhana yaitu addBalance dan SubtBalance.

2.8 addPosession(Pemain, Property, Level)

2.8.1 Kegunaan Command

Command ini digunakan untuk menambah properti pada kepemilikan pemain, command ini tidak mengatur perubahan uang

2.8.2 Skenario Penggunaan

Command ini digunakan jika ada pembelian atau transaksi properti.

2.9 removePosession(Pemain, Property)

2.9.1 Kegunaan Command

Command ini digunakan untuk mencabut kepemilikan properti pada pemain, command ini tidak mengatur perubahan uang

2.9.2 Skenario Penggunaan

Command ini digunakan jika ada penjualan atau transaksi properti.

2.10 checkBangkrut(Pemain)

2.10.1 Kegunaan Command

Command ini digunakan untuk mendeteksi jumlah uang yang dimiliki seorang pemain, jika uang tersebut kurang dari nol, akan diberi pilihan untuk menjual properti atau menyerah.

2.10.2 Skenario Penggunaan

Command ini digunakan ketika terjadi pengurangan uang pada pemain. Jika ada kondisi bangkrut benar yang aktif maka loop permainan akan berakhir.

2.11 resetGame

2.11.1 Kegunaan Command

Command ini digunakan untuk mengulangi semua kondisi game kepada tahap awal permainan baru dimulai.

2.11.2 Skenario Penggunaan

Command ini digunakan permainan telah berakhir atau permainan perlu diulang dari awal.

2.12 checkLocationDetail(X)

2.12.1 Kegunaan Command

Command ini digunakan untuk menampilkan detail dari sebuah lokasi (properti maupun non properti).

2.12.2 Skenario Penggunaan

Ketika command ini dijalankan, akan ditampilkan identitas lokasi (nama dan deskripsi), lalu menampilkan detail properti (kepemilikan, biaya sewa, biaya akuisisi, dan tingkatan properti) jika lokasi adalah properti.

2.13 checkPropertyDetail(X)

2.13.1 Kegunaan Command

Command ini digunakan melihat informasi umum dari property yang ada.

2.13.2 Skenario Penggunaan

Ketika command ini dijalankan, akan ditampilkan identitas property berupa nama, deskripsi, harga tiap tingkatan property, dan harga sewa tiap tingkatan property.

2.14 checkPlayerDetail(X)

2.14.1 Kegunaan Command

Command ini digunakan melihat informasi umum dari pemain yang ada.

2.14.2 Skenario Penggunaan

Ketika command ini dijalankan, akan ditampilkan lokasi saat ini, total uang beserta nilai property yang dimiliki, total asset, kepemilikan property, dan jika pemain yang diperiksa adalah dirinya sendiri maka dapat melihat kepemilikan kartu, pemain musuh tidak dapat dilihat kepemilikan kartunya.

2.15 playGameCenter

2.15.1 Kegunaan Command

Command ini digunakan untuk menjalankan fitur Game Center.

2.15.2 Skenario Penggunaan

Ketika command ini dijalankan, akan diinisialisasi status bermain, hadiah, dan jumlah tebakan yang benar dari pemain.

2.16 guessing_flow

2.16.1 Kegunaan Command

Command ini digunakan untuk mengatur jalannya permainan di Game Center.

2.16.2 Skenario Penggunaan

Ketika command ini dijalankan, akan ditampilkan pilihan untuk lanjut bermain atau berhenti, jika berhenti pemain keluar dari permainan Game Center, jika lanjut bermain, pemain akan lanjut menebak koin.

2.17 inputGuessing

2.17.1 Kegunaan Command

Command ini digunakan untuk menampilkan interaksi penembakan koin.

2.17.2 Skenario Penggunaan

Ketika command ini dijalankan, akan ditampilkan pilihan tebakan(Head or Tail), lalu mengatur pemrosesan data input tersebut.

2.18 useJailCard

2.18.1 Kegunaan Command

Command ini digunakan pemain untuk menggunakan kartu keluar penjara.

2.18.2 Skenario Penggunaan

Ketika command digunakan , jika pemain berada dalam penjara dan memiliki kartu, pemain bebas dari penjara dan kartu telah terpakai. Jika pemain tidak berada di penjara atau jika pemain tidak memiliki kartu penjara, pesan ditampilkan.

2.19 suapSipir

2.19.1 Kegunaan Command

Command ini digunakan pemain untuk membayar denda penjara, ketika ingin keluar dari penjara.

2.19.2 Skenario Penggunaan

Ketika command digunakan, jika pemain berada di penjara dan memiliki uang yang cukup, pemain bebas dari penjara dan uang berkurang (100 dollar). Jika pemain tidak berada dipenjara, atau jika pemain tidak memiliki uang yang cukup, pesan ditampilkan

2.20 runCard

2.20.1 Kegunaan Command

Command ini digunakan untuk mengocok kartu.

2.20.2 Skenario Penggunaan

Ketika command digunakan, jika pemain sedang berada pada chance card dapat

mengocok kartu

B. Command di Dalam Game

2.1 lempar

2.1.1 Kegunaan Command

Command ini digunakan untuk melempar dadu dan melanjutkan permainan.

2.1.2 Skenario Penggunaan

Ketika command ini dijalankan, akan dilemparkan dadu lalu pemain bergerak, setelah itu akan dilakukan aksi lanjutan sesuai dengan tempat pemain setelah bergerak.

2.2 locationDetail

2.2.1 Kegunaan Command

Command ini digunakan untuk menampilkan detail dari sebuah lokasi (property maupun non property).

2.2.2 Skenario Penggunaan

Ketika command ini dijalankan, akan ditampilkan identitas lokasi (nama dan deskripsi), lalu menampilkan detail property (kepemilikan, biaya sewa, biaya akuisisi, dan tingkatan property) jika lokasi adalah property.

2.3 propertyDetail

2.3.1 Kegunaan Command

Command ini digunakan melihat informasi umum dari property yang ada.

2.3.2 Skenario Penggunaan

Ketika command ini dijalankan, akan ditampilkan identitas property berupa nama, deskripsi, harga tiap tingkatan property, dan harga sewa tiap tingkatan property.

2.4 playerDetail

2.4.1 Kegunaan Command

Command ini digunakan melihat informasi umum dari pemain yang ada.

2.4.2 Skenario Penggunaan

Ketika command ini dijalankan, akan ditampilkan lokasi saat ini, total uang beserta nilai property yang dimiliki, total asset, kepemilikan property, dan jika pemain yang diperiksa adalah dirinya sendiri maka dapat melihat kepemilikan kartu, pemain musuh tidak dapat dilihat kepemilikan kartunya.

2.5 useJailCard

2.5.1 Kegunaan Command

Command ini digunakan untuk menggunakan kartu keluar dari penjara.

2.5.2 Skenario Penggunaan

Command ini hanya akan muncul jika pemain yang akan bergerak berada di penjara. Pemain akan keluar dari penjara jika memiliki kartu tersebut dan akan ditolak jika tidak memilikinya.

2.6 suapSipir

2.6.1 Kegunaan Command

Command ini digunakan pemain untuk membayar denda penjara, ketika ingin keluar dari penjara.

2.6.2 Skenario Penggunaan

Ketika command digunakan, jika pemain berada di penjara dan memiliki uang yang cukup, pemain bebas dari penjara dan uang berkurang (100 dollar). Jika pemain tidak berada dipenjara, atau jika pemain tidak memiliki uang yang cukup, pesan ditampilkan

2.7 help

2.1.1 Kegunaan Command

Command ini digunakan untuk memberikan semua command dalam game yang dapat dipanggil oleh pemain.

2.1.2 Skenario Penggunaan

Ketika command ini dijalankan, akan diberikan daftar command yang tersedia. Command ini tersedia pada hampir semua kasus yang meminta input dari pengguna.

2.8 debug

2.8.1 Kegunaan Command

Command ini digunakan untuk keluar dari loop utama permainan dan memungkinkan untuk memanggil command yang tidak terbatas oleh input handling permainan.

2.8.2 Kegunaan Command

Ketika command digunakan, keluar dari overlay permainan menuju terminal prolog. Kegunaannya untuk mencoba semua fitur dan command yang terdapat pada game.

3.2 inputHandling

```

-----
Masukkan perintah: help.

Perintah yang tersedia
lempar.: mulai melempar dadu
locationDetail.: mengecek informasi lokasi yang ada
propertyDetail.: mengecek informasi properti yang ada
playerDetail.: mengecek informasi properti yang ada
debug.: keluar dari overlay game untuk memasukkan command secara direct

Masukkan perintah: |

```

Gambar 3.2.1
Uji coba inputHandling pada input help

```

Masukkan perintah: lempar.

p1
Dadu 1: 6
Dadu 2: 1
Anda maju sebanyak 7 langkah

```

Gambar 3.2.2
Uji coba inputHandling pada input lempar

3.3 checkLocationDetail(X)

```

|--
| ?- checkLocationDetail(a1).
Nama Lokasi      : jakarta
Deskripsi Lokasi : ibukotaIndonesia

Kepemilikan      : Tidak ada
Biaya Sewa Saat Ini : 0
Biaya Akuisisi    : 0
Tingkatan Properti : Tanah

yes
| ?- checkLocationDetail(h2).
Nama Lokasi      : kairo
Deskripsi Lokasi : ibukotaMesir

Kepemilikan      : Tidak ada
Biaya Sewa Saat Ini : 0
Biaya Akuisisi    : 0
Tingkatan Properti : Tanah

Masukkan perintah: locationDetail.
Masukkan nama lokasi: e1.
Nama Lokasi      : berlin
Deskripsi Lokasi : ibukotaJerman

Kepemilikan      : Tidak ada
Biaya Sewa Saat Ini : 0
Biaya Akuisisi    : 0
Tingkatan Properti : Tanah

```

Gambar 3.3.1
Uji coba checkLocationDetail di dalam dan di luar program

3.4 checkPropertyDetail(X)

```

(125 ms) yes
| ?- checkPropertyDetail(b3).
Nama Properti    : New Delhi
Deskripsi Properti : Ibukota India

Harga Tanah      : 120
Harga Bangunan 1 : 170
Harga Bangunan 2 : 220
Harga Bangunan 3 : 270
Harga Landmark   : 320

Harga Sewa Tanah : 16
Harga Sewa Bangunan 1 : 50
Harga Sewa Bangunan 2 : 100
Harga Sewa Bangunan 3 : 300
Harga Sewa Landmark : 600

yes
| ?-

Masukkan perintah: propertyDetail.
Masukkan nama lokasi: b3.
Nama Properti    : New Delhi
Deskripsi Properti : Ibukota India

Harga Tanah      : 120
Harga Bangunan 1 : 170
Harga Bangunan 2 : 220
Harga Bangunan 3 : 270
Harga Landmark   : 320

Harga Sewa Tanah : 16
Harga Sewa Bangunan 1 : 50
Harga Sewa Bangunan 2 : 100
Harga Sewa Bangunan 3 : 300
Harga Sewa Landmark : 600

```

Gambar 3.4.1
Uji coba checkPropertyDetail di dalam dan di luar program

3.5 checkPlayerDetail(X)

```
?- checkPlayerDetail(p1).
Informasi p1
Lokasi          : go
Total Uang      : 2000
Total Nilai Properti : 0
Total Aset      : 2000
Daftar Kepemilikan Properti :
Kamu tidak memiliki kartu
(16 ms) yes

Masukkan perintah: playerDetail
.
Masukkan nama pemain: p2.
Informasi p2
Lokasi          : go
Total Uang      : 2000
Total Nilai Properti : 0
Total Aset      : 2000
Daftar Kepemilikan Properti :
```

Gambar 3.5.1

Uji coba checkPlayerDetail di dalam dan luar program

3.6. playGameCenter

```
playGameCenter.
Sisi manakah yang akan muncul?? (masukkan angka 1 atau 2)
1. HEAD
2. TAIL
Guess: 1.
Selamat!! Kamu berhasil menebak dengan benar, hadiah yang berhasil dikumpulkan: $150

Siap untuk mendapatkan hadiah yang lebih besar?? (masukkan angka 1 atau 2)
1. Ya, Lanjut Bermain
2. Tidak, Klaim Hadiah
Pilihan: 1.

Sisi manakah yang akan muncul?? (masukkan angka 1 atau 2)
1. HEAD
2. TAIL
Guess: 1.
Selamat!! Kamu berhasil menebak dengan benar, hadiah yang berhasil dikumpulkan: $300

Siap untuk mendapatkan hadiah yang lebih besar?? (masukkan angka 1 atau 2)
1. Ya, Lanjut Bermain
2. Tidak, Klaim Hadiah
Pilihan: 2.

Terima Kasih telah bermain di Game Center! Hadiah yang berhasil dikumpulkan: $300
```

Gambar 3.6.1

Uji coba playGameCenter pada program

3.7 useJailCard

```
useJailCard.
Kamu tidak berada di penjara...
```

Gambar 3.7.1

Uji coba useJailCard pada program ketika pemain tidak berada di penjara

```
Anda berada di penjara!!!
pilih input berikut untuk keluar:
lempar      : menggunakan keburuntungan double dadu untuk keluar dari penjara
useJailCard : menggunakan kartu Jail
suapSipir  : menggunakan uang untuk menyuap sipir!
Masukkan perintah: useJailCard.
Kamu menggunakan Keluar dari Jail!

?- useJailCard.
Kamu menggunakan Keluar dari Jail!
```

Gambar 3.7.1

Uji coba useJailCard pada program ketika pemain berada di penjara di dalam dan di luar program

3.8 suapSipir

```

Posisi
P1 : j1
Saldo Player P1: 2000

P2 : go
Saldo Player P2: 2000

Sekarang giliran: p1
Tulis 'help.' untuk memberikan daftar perintah yang tersedia

Anda berada di penjara!!!
pilih input berikut untuk keluar:
lempar      : menggunakan keburuntungan double dadu untuk keluar dari penjara
useJailCard : (kamu tidak memilikinya :'(
suapSipir   : menggunakan uang untuk menyuap sipir!
Masukkan perintah: suapSipir.
Sipir tertarik dengan uang yang kamu berikan
Kamu berhasil keluar dari penjara!
rasa bersalah menghantui dirimu...

```

Gambar 3.8.1
Uji coba suapSipir di dalam program

3.9 Landing di properti kosong

JL	B3	B2	B1	CC1	A2
		p1			

```

Posisi
P1 : b2
Saldo Player P1: 2000

P2 : go
Saldo Player P2: 2000

Sekarang giliran: p1
Tulis 'help.' untuk memberikan daftar perintah yang tersedia

b2 belum memiliki pemilik, mau beli?

Aksi: |

```

Gambar 3.9.1
Uji coba pergerakan yang berakhir di properti kosong

```

Aksi: beli.
Nama Properti      : Islamabad
Deskripsi Properti : Ibukota Pakistan

Harga Tanah        : 100
Harga Bangunan 1   : 150
Harga Bangunan 2   : 200
Harga Bangunan 3   : 250
Harga Landmark     : 300

Harga Sewa Tanah   : 10
Harga Sewa Bangunan 1 : 40
Harga Sewa Bangunan 2 : 90
Harga Sewa Bangunan 3 : 270
Harga Sewa Landmark : 550

Uang Anda: 2000
Properti yang mau dibeli:

```

Gambar 3.9.2
Uji coba aksi beli setelah mendarat di properti kosong

Uang Anda: 2000
 Properti yang mau dibeli: landmark.
 landmark belum bisa dibeli

Gambar 3.9.3

Uji coba aksi membeli landmark sebelum memiliki bangunan 3

Uang Anda: 2000
 Properti yang mau dibeli: bangunan3.
 Bangunan3 berhasil dibeli!
 Islamabad Sekarang menjadi milikmu

Gambar 3.9.4

Uji coba aksi membeli bangunan 3 dengan uang cukup

3.10 Landing di properti sendiri

JL	B3	B2 p1	B1	CC1	
----	----	----------	----	-----	--

p1
Gedung 3

Posisi
 P1 : b2
 Saldo Player P1: 1500
 P2 : c1
 Saldo Player P2: 2000
 Sekarang giliran: p1
 Tulis 'help.' untuk memberikan daftar perintah yang tersedia

Mau menambah properti?
 Aksi:

Gambar 3.10.1

Uji coba pergerakan yang berakhir di properti sendiri

Mau menambah properti?

Aksi: tambah.
 Nama Properti : Jakarta
 Deskripsi Properti : Ibukota Indonesia

Harga Tanah : 50
 Harga Bangunan 1 : 100
 Harga Bangunan 2 : 150
 Harga Bangunan 3 : 200
 Harga Landmark : 250

Harga Sewa Tanah : 5
 Harga Sewa Bangunan 1 : 18
 Harga Sewa Bangunan 2 : 30
 Harga Sewa Bangunan 3 : 90
 Harga Sewa Landmark : 250

Uang Anda: 1700
 Properti yang mau dibeli:

Gambar 3.10.2

Uji coba tambah setelah mendarat di properti sendiri

```

Uang Anda: 1700
Properti yang mau dibeli: landmark.
Landmark berhasil dibeli!

```

Gambar 3.10.3

Uji coba memilih landmark setelah memiliki bangunan3 dan melewati go satu kali

3.11 Landing di properti pemain lain

	A1		Posisi	Posisi
	p2		P1 : jl	P1 : jl
			Saldo Player P1: 1850	Saldo Player P1: 1940

p1
Gedung 3

	P2 : go		P2 : a1
	Saldo Player P2: 2000		Saldo Player P2: 1910

Sekarang giliran: p2
Tulis 'help.' untuk memberikan daftar perintah yang tersedia
Ambil Alih?(ya/tidak)

Gambar 3.11.1

Uji coba perubahan saldo ketika mendarat di properti pemain lain

JL	B3	B2	B1	CC1	A2	GC	A1
p1							p2

p2
Gedung 3

```

Posisi
P1 : jl
Saldo Player P1: 2940

P2 : a1
Saldo Player P2: 910

Sekarang giliran: p1
Tulis 'help.' untuk memberikan daftar perintah yang tersedia

Mau menambah properti?

Aksi: |

```

Gambar 3.11.2

Uji coba ketika mengambil alih properti

JL	B3	B2	B1	CC1	A2	GC	A1
							p1 p2

p2
Gedung 3

```

Posisi
P1 : a1
Saldo Player P1: 3050

P2 : a1
Saldo Player P2: 1000

Sekarang giliran: p1
Tulis 'help.' untuk memberikan daftar perintah yang tersedia

Apakah kamu ingin menggunakan Kartu Sakti?[y/n]

```

Gambar 3.11.3

Uji coba ketika mendarat di properti pemain lain dan memiliki kartu sakti, uang yang ditampilkan sudah dikurangi


```
yes
| ?- moveToLocation(p1, a1).

p1 melewati go, uangmu sekarang, 1700
```

Gambar 3.13.1
Uji coba pergerakan yang melewati go

```
(63 ms) yes
| ?- moveToLocation(p1, go).

p1 melewati go, uangmu sekarang, 1650

yes
| ?- checkLokasi(p1).

Masuk go

Pilih properti yang ingin kamu bangun: |
```

Gambar 3.13.2
Uji coba ketika mendarat di petak go

```
Posisi
P1 : go
Saldo Player P1: 2150

P2 : go
Saldo Player P2: 2000

Sekarang giliran: p1
Tulis 'help.' untuk memberikan daftar perintah yang tersedia

Mau menambah properti?

Aksi: tambah.
Nama Properti      : Jakarta
Deskripsi Properti : Ibukota Indonesia
```

Gambar 3.13.3
Uji coba memilih properti, selanjutnya mirip landing di properti sendiri

3.14 Tax

```
(78 ms) yes
| ?- moveToClosestTax(p1).

yes
| ?- checkLokasi(p1).

Masuk pajak

Anda harus membayar pajak sebesar 200.0 dolar.

(15 ms) yes
| ?-
```

Gambar 3.14.1
Uji coba mendarat pada petak pajak

```

P1 : tx
Saldo Player P1: 1300.0
Saldo Player P1: 1500
P2 : a1
Saldo Player P2: 2000
Saldo Player P2: 2000

```

Gambar 3.14.1

Uji coba perubahan uang setelah mendarat pada petak pajak

3.15 Chance Card

Masuk chance card
Kamu Mendapatkan Kartu Untuk Keluar dari Jail

Gambar 3.16.1

Uji coba ketika player masuk ke chance card

```

Informasi p1

Lokasi                : cc1
Total Uang            : 2000
Total Nilai Properti  : 0
Total Aset            : 2000

Daftar Kepemilikan Properti :

Kartu Kamu :

Keluar dari Jail

```

Gambar 3.16.1

Uji coba kepemilikan kartu

3.16 checkBangkrut dan resolveBangkrut

```

Posisi
P1 : a1
Saldo Player P1: 3590

P2 : a1
Saldo Player P2: -90

Sekarang giliran: p2
Tulis 'help.' untuk memberikan daftar perintah yang tersedia

Uangmu tidak mencukupi untuk membayar, kamu harus memilih properti untuk dijual

Utangmu senilai -90
properti yang kamu miliki adalah sebagai berikut
1. a2 - Bangunan3

atau tulis 'menyerah.' jika tidak ingin melanjutkan permainan
Properti yang akan dijual:

```

Gambar 3.16.1

Uji coba ketika uang pemain mencapai nilai negatif dan pemain tersebut memiliki aset yang cukup

```

Posisi
P1 : a1
Saldo Player P1: 3950

P2 : a1
Saldo Player P2: -18.0

Sekarang giliran: p2
Tulis 'help.' untuk memberikan daftar perintah yang tersedia

Pemain p2 telah bangkrut
Pemenangnya adalah p1

Permainan telah berakhir, terima kasih telah bermain monopoli boku no prolog
panggil 'resetGame.' untuk mereset kembali ke kondisi semula

(62 ms) yes
| ?-

```

Gambar 3.16.2

Uji coba ketika uang pemain mencapai nilai negatif dan pemain tersebut tidak memiliki aset yang cukup

atau tulis 'menyerah.' jika tidak ingin melanjutkan permainan
 Properti yang akan dijual: menyerah.

Pemain p2 telah bangkrut
 Pemenangnya adalah p1

Permainan telah berakhir, terima kasih telah bermain monopoli boku no prolog
 panggil 'resetGame.' untuk mereset kembali ke kondisi semula

```

(172 ms) yes
| ?-

```

Gambar 3.16.3


Uji coba ketika pemain memilih untuk menyerah

3.17 World Cup

```

| ?- checkLokasi(p1).

```



```

Masukkan properti yang akan menjadi tuan rumah: a1.
a1 menjadi Tuan rumah World cup di Dunia Boku no Prolog!
(nilai properti a1 naik 2 kali lipat)
(akhirnya cewek-cewek akan terpukau denganku!!)
pikirmu..

```

Gambar 3.17.1

Uji coba ketika masuk petak world cup

no	?- checkLocationDetail(a1).	yes	?- checkLocationDetail(a1).
Nama Lokasi	: Jakarta	Nama Lokasi	: Jakarta
Deskripsi Lokasi	: Ibukota Indonesia	Deskripsi Lokasi	: Ibukota Indonesia
Kepemilikan	: p1	Kepemilikan	: p1
Biaya Sewa Saat Ini	: 30	Biaya Sewa Saat Ini	: 60
Biaya Akuisisi	: 600	Biaya Akuisisi	: 600
Tingkatan Properti	: Bangunan2	Tingkatan Properti	: Bangunan2

Gambar 3.17.2

Uji coba perubahan harga petak yang dipilih untuk world cup

3.18 Dadu dan double

```

p1                               Saldo Player P2: 2000
Dadu 1: 6
Dadu 2: 6                       Sekarang giliran: p1
Anda maju sebanyak 12 langkah   Tulis 'help.' untuk memberikan

Dobel!                           Masukkan perintah:

```

Gambar 3.18.1

Uji coba ketika mendapatkan dobel, giliran pemain tidak berubah

```

Masukkan perintah: lempar.
Dadu 1: 6
Dadu 2: 6
Anda masuk ke jail karena mendapatkan Double 3 kali berturut-turut
Anda masuk ke penjara!!!

```

Gambar 3.18.2

Uji coba ketika mendapatkan dobel tiga kali berturut - turut

3.19 Color Set

H1	p1	yes	?- checkLocationDetail(h1).	--	?- checkLocationDetail(h1).
Tanah		Nama Lokasi	: Athena	Nama Lokasi	: Athena
		Deskripsi Lokasi	: Ibukota Yunani	Deskripsi Lokasi	: Ibukota Yunani
H2	p1	Kepemilikan	: p1	Kepemilikan	: p1
p1	Tanah	Biaya Sewa Saat Ini	: 60	Biaya Sewa Saat Ini	: 120
		Biaya Akuisisi	: 720	Biaya Akuisisi	: 720
		Tingkatan Properti	: Tanah	Tingkatan Properti	: Tanah

Gambar 3.19.1

Uji coba melihat harga sewa suatu petak jika tidak memiliki dan jika memiliki satu warna penuh

3.20 Debug

Masukkan perintah: debug.

Masuk ke mode debug

```

true ?

(78 ms) yes
| ?-

```

Gambar 3.20.1

Uji coba command debug, untuk 'keluar' dari dalam game (mode debug)

3.21 ResetGame

Posisi		Posisi
P1 : d2		P1 : go
Saldo Player P1: 1800		Saldo Player P1: 4000
	(204 ms) yes	
P2 : tx	?- resetGame.	P2 : go
Saldo Player P2: 1800.0		Saldo Player P2: 4000

Gambar 3.21.1

Uji coba reset game pada program, (kiri) gambar sebelum di reset dan (kanan) gambar sesudah di reset

BAB IV

KESIMPULAN

Monopoli merupakan papan permainan dengan mekanik yang tidak sederhana. Pengimplementasian dalam bahasa pemrograman GNU Prolog yang berparadigma logic adalah tantangan tersendiri bagi penulis mengingat Prolog memiliki syntax dan cara kerja yang berbeda dengan bahasa pemrograman yang biasa penulis pakai dalam perkuliahan. Tapi pada akhirnya dengan bantuan dari asistensi dan riset yang mendalam, program inipenulis, permainan Monopoli dengan dua pemain, dapat rampung dan dimainkan.

Hambatan terbesar penulis dalam penyelesaian tugas besar ini adalah memahami cara kerja Prolog serta mengaplikasikannya ke dalam fitur-fitur dan aturan permainan Monopoli. Khususnya pada fitur yang memerlukan algoritma rekursif, cut, dan fail. Milestone yang diadakan mingguan dalam pengerjaan tugas besar ini membantu penulis untuk menetapkan tengat waktu pengerjaan dan tetap menghasilkan kemajuan sehingga dapat menyelesaikan tugas ini tepat waktu.

Akhir kata, tim Boku no Prolog mengucapkan terima kasih atas semua yang mendukung penyelesaian tugas besar program Monopoli ini. Walaupun jauh dari kata sempurna, satu hal yang penulis pastikan adalah penulis telah mengeluarkan performa terbaik yang penulis bisa pada waktu yang diberi dan semoga dapat memenuhi ekspektasi pemain dari Monopoli versi Boku no Prolog ini. Selamat Bermain!

LAMPIRAN

PEMBAGIAN TUGAS

No.	Nama	NIM	Persentase	Pembagian Tugas
1.	Naufal Syifa Firdaus	13521050	20%	Map, Tax, ASCII Art, Main, Laporan
2.	Muhammad Fadhil Amri	13521066	20%	Lokasi, Properti, Game Center, World Cup, Color Set, Main, Laporan
3.	Akmal Mahardika P.	13521070	20%	Jail, World Tour, ASCII Art, World Cup, Main, Laporan
4.	Razzan Daksana Yoni	13521087	20%	Dadu, Chance Card, World Cup, Variasi Chance Card, Main, Laporan
5.	Muhamad Aji Wibisono	13521095	20%	Player, Bangkrut, Buildable Go, World Cup, Color Set, Main, Laporan

Repository Github

Berikut adalah link repository GitHub untuk program penulis.

<https://github.com/GAIB20/tugas-besar-logika-komputasional-2022-boku-no-prolog>