

## MILESTONE 2

Kelompok 6 : | ?- letsGetRich(X).

1. 13521094 Angela Livia Arumsari
2. 13521044 Rachel Gabriela Chen
3. 13521046 Jeffrey Chow
4. 13521100 Alexander Jason
5. 13521074 Eugene Yap Jin Quan

## PEMBAGIAN TUGAS

No	Fitur	PIC
1	Papan	Eugene Yap Jin Quan
2	Lokasi	Rachel Gabriela Chen
3	Properti	Alexander Jason
4	Chance Card	
5	Perpajakan	Eugene Yap Jin Quan
6	Penjara	Rachel Gabriela Chen
7	Parkir Gratis	
8	World Tour	Angela Livia Arumsari
9	Pemain	
10	Dadu	Jeffrey Chow
11	Mekanisme Bangkrut	
12	Move	Eugene Yap Jin Quan, Rachel Gabriela Chen
13	Main	All
14	Utility	All
15	Bonus 1	Jeffrey Chow
16	Bonus 4	Rachel Gabriela Chen
17	Bonus 5	Eugene Yap Jin Quan

## LIST FACTS AND RULES FITUR

### 1. Papan

#### \*\*\*\* FAKTA \*\*\*\*

- **tile(X, Tile)**

Korespondensi jenis petak pada papan dengan indeks 0-31

- **grid(X, Y, Info)**

Matriks yang menyimpan teks papan

#### \*\*\*\* RULE \*\*\*\*

- **map**

Mencetak peta yang terdiri atas beberapa baris matriks papan, border, dan padding

- **border**

Mencetak border antara sisi atas dan bawah papan dengan tengah papan

- **padding(R)**

Mencetak padding antara informasi kepemilikan petak pada pencetakan papan jika R adalah 0 atau R adalah 10

- **write\_grid(R,C)**

Mencetak elemen dari matriks papan pada baris R dan kolom C

- **write\_cols(Row, StartCol)**

Mencetak baris ke-(Row) dari matriks papan, dimulai dari kolom ke-(StartCol)

- **write\_playerinfo**

Mencetak informasi posisi pemain pada papan

### 2. Lokasi

#### \*\*\*\* FAKTA \*\*\*\*

- **is\_property(X)**

Mendeclare X adalah properti.

- **nama\_lokasi(X, Y)**

Y adalah nama lokasi X.

- **desc\_lokasi(X,Y)**

Y adalah deskripsi lokasi X.

- **kepemilikan(X,Y)**

Y adalah pemilik X.

\*\*\*\* **RULE** \*\*\*\*

- **checkLocationDetail(X)**

Mencetak detail lokasi X.

### 3. Properti

\*\*\*\* **FAKTA** \*\*\*\*

- **hargaBeli(Loc, JenisTanah, HargaBeli)**

Mendeclare fakta harga beli suatu lokasi yang bergantung jenis tanahnya.

Tanah = tanah, bg1 = bangunan 1, bg2= bangunan 2, bg3 = bangunan3, lm = landmark

\*\*\*\* **RULE** \*\*\*\*

- **hargaSewa(Loc, JenisTanah, HargaSewa)**

Menyatakan harga sewa yaitu harga total dari tingkatan tanah lalu dibagi 2

- **hargaAmbil(Loc, JenisTanah, HargaAmbil)**

Menyatakan harga ambil alih yaitu harga total dari tingkatan tanah yang sudah dibeli

- **checkPropertyDetail(X)**

Menampilkan detail properti X.

- **buy**

Mengakuisisi properti

- **rent**

Menyewa properti

- **sell**

Menjual properti

- **propertyMechanism**

Mekanisme saat player sampai di properti

#### 4. Chance card

##### \*\*\*\* FAKTA \*\*\*\*

- **chance\_cards(L)**

Memuat daftar chance card yang ada

- **card\_change(X,Y)**

Y adalah rarity dari X

- **card\_info(X,Y)**

Y adalah info dari X

##### \*\*\*\* RULE \*\*\*\*

- **drawChanceCard**

Mengambil chance card secara random

- **useChanceCard**

Menggunakan chance card

- **displayCards(Cards)**

Menampilkan isi list Cards

- **cardMechanism(X)**

Mekanisme kartu X

- **insertCardToList(X,N)**

Memasukkan kartu X sebanyak N kali ke list chance cards

- **initChanceCard**

Inisiasi chance card

## 5. Perpajakan

\*\*\*\* FAKTA \*\*\*\*

-

\*\*\*\* RULE \*\*\*\*

- **payTax(P)**

Player P membayar pajak 10% dari total aset yang dimilikinya

## 6. Penjara

\*\*\*\* FAKTA \*\*\*\*

- **in\_jail(X)**

X berada dalam penjara.

- **dice\_after\_jail(X, Y)**

Y adalah banyak dadu terlempar saat X dipenjara

- **hasPickedJailMechanism**

Jail Mechanism sedang dijalankan

\*\*\*\* RULE \*\*\*\*

- **jailMechanism**

Mencetak bahwa player berada dalam jail.

- **escapeJail**

Membebaskan player dari jail.

- **initJail**

Membuat state awal jail

- **visitJail**

Mekanisme player yang hanya menginjak jail

- **incDiceAfterJail**

Melakukan increment jumlah dice dilempar saat berada di penjara

## 7. Parkir gratis

\*\*\*\* FAKTA \*\*\*\*

-

\*\*\*\* RULE \*\*\*\*

- **parkirGratisMechanism**

Mencetak informasi bahwa player sedang berada di parkir gratis

## 8. World tour

\*\*\*\* FAKTA \*\*\*\*

\*\*\*\* RULE \*\*\*\*

- **apparateMechanism**

Melakukan mekanisme pemilihan destinasi dan pemindahan lokasi current player

- **worldTourMechanism**

Melakukan mekanisme menawarkan world tour ke current player dan mengurangi uang player jika melakukan world tour

## 9. Pemain

\*\*\*\* FAKTA \*\*\*\*

- **currentPlayer(X)**

Mendeklarasikan sekarang adalah giliran player X

- **namePlayer(X,Y)**

Mendeklarasikan Y adalah nama dari player X

- **locPlayer(X,Y)**

Mendeklarasikan lokasi player X dengan Y sebagai nomor tile pada board

- **cashPlayer(X,Y)**

Mendeklarasikan uang yang dimiliki player X adalah sebesar Y

- **listPropPlayer(X,Y)**

Mendeklarasikan Y adalah list lokasi yang dimiliki oleh player X

- **cardPlayer(X,Y)**

Mendeklarasikan Y adalah list chance card yang dimiliki oleh player X

- **is\_player(X)**

X adalah player

- **tempList**

Temporary list untuk looping

- **tempIndeks**

Temporary indeks untuk looping

\*\*\*\* **RULE** \*\*\*\*

- **initPlayer**

Meminta nama player dan melakukan inisiasi player

- **daftarProp(X)**

Membuat list daftar properti yang dimiliki player X

- **countProp(X,Y)**

Menghitung nilai properti yang dimiliki player X dan menyimpan hasilnya di Y

- **addCashGo(X)**

Menambah uang player X ketika player melewati GO

- **displayProp(X)**

Mencetak daftar properti yang dimiliki player X

- **displayCard(X)**

Mencetak daftar chance card yang dimiliki player X

- **checkPlayerDetail(X)**

Menampilkan detail player X

- **incCash(X)**

Menambahkan duit current player sebanyak X

- **decCash(X)**

Mengurangi duit current player sebanyak X

- **totalAssets(X)**

Menghitung total aset currentPlayer, X adalah total aset

## 10. Dadu

\*\*\*\* FAKTA \*\*\*\*

- **playerDouble(X,Y)**

Mencatat player X mendapatkan double dice sebanyak Y kali.

\*\*\*\* RULE \*\*\*

- **rand(X)**

Random angka untuk dadu,  $0 < X < 7$

- **throwDice**

Melempar dadu untuk currentPlayer

- **incPlayerDouble**

Melakukan increment jumlah double dice currentPlayer

- **switchPlayer**

Mengganti player yang sedang bermain

## 11. Mekanisme bangkrut

\*\*\*\* FAKTA \*\*\*\*

-

\*\*\*\* RULE \*\*\*

- **checkBankrupt(Amount)**

Mekanisme untuk mengecek bangkrut



- **bankruptMechanism(Amount)**

Mekanisme player terindikasi bangkrut

- **endGame**

Menyelesaikan permainan karena salah satu player sudah bangkrut.

- **surrender**

currentPlayer menyerah

## 12. Mekanisme move

\*\*\*\* FAKTA \*\*\*\*

-

\*\*\*\* RULE \*\*\*

- **move(P, X)**

Mengubah lokasi player P dengan memajukan posisi P sejumlah X dimana X didapatkan dari hasil throwDice.

## 13. Main

\*\*\*\* FAKTA \*\*\*\*

- **menu\_status(X)**

Menyimpan status menu, untuk mengecek rule mana yang bisa dijalankan.

- **curr\_turn(X)**

Menyimpan player yang sedang memiliki giliran

\*\*\*\* RULE \*\*\*

- **changeStatus(P)**

Mengubah status P dan aksi-aksi yang dapat dilakukan sesuai lokasi P.

## 1. Utility

\*\*\*\* RULE \*\*\*

- **insertLast(X, L, Result)**

Memasukkan X ke dalam L pada akhir list L

- **deleteAtList(Idx, L, Result)**

Menghapus elemen pada index ke Idx pada list L

- **getItemAtIdx(List, Idx, Item)**

Item adalah elemen pada List dengan index Idx

- **displayList(List)**

Menuliskan ke layar elemen list

- **insertNTimes(Val,List,NewList,N)**

Insert Val sebanyak N kali kedalam List

- **getValAtIdx(List, Idx, Val)**

Val adalah element ke-Idx pada List

- **subset(List1,List2,Ans)**

Ans adalah 1 jika List1 adalah subset dari List2, dan 0 jika bukan.

- **subsetRec(List1,List2,Ans)**

Ans adalah jumlah elemen List1 pada List2

- **isIn (X, List, Ans)**

Ans adalah 1 jika X ada didalam List, 0 jika tidak ada

- **getIndex(List,X,Index)**

Index adalah index X pada List

2. **Bonus 1 (Colorset)**

3. **Bonus 4 (Chance Card)**

4. **Bonus 5 (flip coin)**

\*\*\*\* FAKTA \*\*\*

- **coin\_side(Face, Index)**

Indeks dari muka koin yang digunakan dalam proses pengacakan, 0 adalah 'head', 1 adalah 'tail'.

\*\*\*\* RULE \*\*\*

- **guessFlip(Success)**

Menerima masukan pemain berupa tebakan 'head' atau 'tail', mengembalikan Success bernilai 0 jika tebakan salah, atau 1 jika tebakan benar. Apabila tebakan benar, uang pemain akan ditambahkan sebanyak 100.

- **loopGuess(lter, Success)**

Melakukan perulangan menebak muka koin (dipanggil pada playCoinFlip maksimal sebanyak 3 kali). Jika lter bernilai kurang dari atau sama dengan 0, atau Success bernilai 0, perulangan akan berhenti.

- **playCoinFlip**

Memulai permainan coin flip jika pemain berada di kotak CF.

UNSTARTED TASKS	ONGOING TASKS	COMPLETED TASKS
Bonus 1 (colorset)	Player, Property, Main, bonus 4 (chance card)	Papan, Lokasi, Parkir Gratis, Dadu, Perpajakan, World Tour, Mekanisme Bangkrut, Utility, Jail, Move, bonus 5 (flip coin), Chance Card