**LAPORAN TUGAS BESAR**

**IF2210/Pemrograman Berorientasi Objek**

**Monopolee**

Dipersiapkan oleh:

Fadel Nararia Rahman 18217005

David Petra Natanael 18217011

Nadya Anastasia 18217020

Lidya Jessica 18217037

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Sekolah Teknik Elektro dan Informatika ITB** | **Nomor Dokumen** | **Halaman** |
| *IF2210-TB-02(Ucul Banget Lah)* | 59 |

Daftar Isi

[1 Ringkasan 3](#_Toc6766374)

[2 Penjelasan Tambahan Spesifikasi Tugas 4](#_Toc6766375)

[*2.1* Implementasi Java Swing 4](#_Toc6766376)

[2.2 Jumlah Pemain/*Player* 4](#_Toc6766377)

[*2.3* Kartu Bebas Penjara 4](#_Toc6766378)

[*2.4* *Free Parking Space* 5](#_Toc6766379)

[2.5 Kondisi Mendarat 5](#_Toc6766380)

[2.6 Menjual Property 5](#_Toc6766381)

[2.7 Kondisi Menang 5](#_Toc6766382)

[2.8 Pembelian Rumah 5](#_Toc6766383)

[3 Program Utama 6](#_Toc6766384)

[*4* *Class Diagram* 6](#_Toc6766385)

[5 Deskripsi *Class Diagram* (Dokumentasi Kelas yang Dibuat) 6](#_Toc6766386)

[6 Fitur-fitur Monopolee 32](#_Toc6766387)

[*7* *Screenshot Program* 43](#_Toc6766388)

[8 Letak Implementasi Cakupan Tugas 51](#_Toc6766389)

[9 Pembagian Kerja dalam Kelompok 53](#_Toc6766390)

[10 Lampiran 55](#_Toc6766391)

[10.1 Deskripsi Tugas Besar 55](#_Toc6766392)

[*10.2* *Log Activity* Anggota Kelompok 56](#_Toc6766393)

[*10.3* *Class Diagram* 58](#_Toc6766394)

# **Ringkasan**

Monopoli adalah salah satu permainan *board game* yang sudah terkenal. Dalam permainan Monopoli, pemain dapat membeli petak-petak kota/tempat dan tujuannya yaitu menguasai semua petak hingga pemain lain yang mendarat di petak tersebut harus membayar. Kondisi menang dinyatakan ketika pemain lain tidak dapat membayar harga sewa lagi dan semua propertinya sudah habis.

Monopoli dapat dimainkan beberapa orang dengan cara melakukan lemparan dadu secara bergiliran. Monopoli yang akan dibangun ini memiliki sejumlah petak. Setiap petak memiliki tipenya masing-masing. Ada 2 tipe yaitu *Property* dan *Space*. *Property* dapat dibagi menjadi *Lot*, *Railroad*, dan *Utility*. Sedangkan *Space* adalah sisa petak yang tidak dapat dibeli pemain. Untuk sistem pembayaran *Property*, akan dilakukan hal-hal seperti ini :

1. *Lot* : harga sewa saja adalah ⅛ dari harga beli *Lot*. Jika punya 1 kompleks, maka harganya menjadi ¼ harga beli. Jika punya 1 rumah, harga sewa menjadi ½ harga beli. Untuk 2 rumah yaitu 1x harga beli. Untuk 3 rumah yaitu 2x harga beli dan untuk 4 rumah 4x harga beli. Syarat membeli rumah adalah pemain harus memiliki *Lot* dalam 1 kompleks.
2. *Railroad* : harga sewa saja adalah ⅛ harga beli. Jika punya 2 *railroad* maka harga sewa jadi ¼ harga beli. Lalu punya 3 menjadi ½ harga beli dan jika punya 4 jenis maka harga menjadi 1x harga beli.
3. *Utility* : harga sewa saja adalah ⅛ harga beli. Jika punya 2 *utility* yaitu harga menjadi ½ harga beli.

*Turn* suatu pemain dibatasi oleh satuan waktu 30 detik. Bila selama 30 detik itu pemain tidak melakukan gilirannya, maka setelah 30 detik berlalu, giliran pemain akan habis dan dilanjutkan ke pemain selanjutnya.

Terdapat pula kartu *Chance* dan *Community Chest* yang berisi hal-hal seperti perintah dapat duit, perintah maju ke suatu petak, perintah masuk penjara, dll. Untuk petak *Go To Jail* mengharuskan pemain masuk penjara dan bila ingin keluar maka ia harus menggunakan kartu bebas penjara yang diperoleh dari *Chance / Community Chest* atau juga membayar langsung. Selain itu, pemain juga dapat melakukan *roll* dadu dan dadu yang *double* akan membuat pemain keluar penjara. Apabila selama 3x kesempatan pemain tersebut tidak mendapatkan dadu *double* maka ia otomatis harus keluar penjara dengan cara membayar ke bank. Untuk petak *Tax*, pemain akan membayar sejumlah uang kepada bank.

Tugas ini akan didokumentasikan baik dalam *source code* maupun lewat laporan ini. Laporan ini akan berisi ringkasan tugas besar, spesifikasi dan asumsi tambahan, program utama, list *class* yang dibuat, fitur-fitur yang ada, *screenshot* program, dan pembagian kerja kelompok.

Beberapa tujuan tugas besar ini yaitu agar dapat mengerti dan mengaplikasikan konsep *object-oriented programming*, menggunakan kreativitas untuk membuat Monopoli, hingga menggunakan konsep *multi-threading*. Hasil permainan monopoli yang kami buat ini, kami berikan nama ***Monopolee****.*

# **Penjelasan Tambahan Spesifikasi Tugas**

Beberapa tambahan spesifikasi tugas yang belum tercakup di deskripsi tugas akan ditambahkan pada bagian ini. Bagian ini termasuk juga **asumsi-asumsi** yang kami gunakan.

## **Implementasi Java Swing**

Kami menggunakan Java Swing sebagai GUI dari game **Monopolee** kami. Java Swing akan menampilkan sebuah *screen* baru yang dapat menampilkan *board* permainan disusul dengan adanya *player* yang dapat bergerak pada petak-petak permainan. Terdapat juga tombol-tombol interaktif yang dipasang agar permainan semakin mudah dimainkan oleh *user*. Setiap petak juga ditandai dengan berbagai warna berbeda baik sesuai kompleksnya maupun sesuai warna untuk menghias.

## **Jumlah Pemain/*Player***

Pada spesifikasi disebutkan bahwa permainan ini dapat dimainkan oleh beberapa orang. Kami menggunakan asumsi jumlah pemain yang dapat memainkan ini yaitu 2 orang.

## **Kartu Bebas Penjara**

Kartu bebas penjara bisa didapatkan *player* ketika mengambil kartu *Chance / Community Chest*. Masing-masing kartu *Chance / Community Chest* terdapat 1 buah kartu bebas penjara. Ketika seorang pemain mendapatkannya, diasumsikan kartu bebas penjara yang ada di *Chance / Community Chest* tidak berkurang, hanya menambahkan atribut jumlah kartu penjara milik *player*, sehingga jumlah *deck cards* tetap 16 buah untuk masing-masing *Chance / Community Chest*.

## ***Free Parking Space***

Diasumsikan ketika seorang pemain mendarat di *Free Parking* maka ia akan mendapatkan tambahan uang 100, namun pemain tersebut hanya diam saja tanpa perintah pindah posisi.

## **Kondisi Mendarat**

Ketika seorang *player* mendapatkan perintah untuk mendarat di suatu tempat, baik itu dari perintah Card, ataupun setelah bebas dari penjara, giliran/*turn* pemain tersebut dilimit hanya 1x saja. Misalnya ketika *player* mendapatkan kartu ”Go To Kopo”, maka player tersebut hanya pergi ke Tile Kopo tanpa bisa melakukan perintah untuk membelinya.

## **Menjual Property**

Diimplementasikan fitur menjual property (bukan mortgage) dimana seorang player dapat menjual property yang dimilikinya (beserta rumah yang dibangun di Lot jika ada). Penjualan property ditetapkan mendapatkan harga penuh property beserta rumah pada Lot. Misalnya sebuah Lot harga belinya 200, maka ketika seorang player menjual Lot tersebut, ia akan mendapat tambahan uang 200.

## **Kondisi Menang**

Kondisi menang seorang player terjadi ketika pemain tersebut tidak dapat membayar lagi harga sewa suatu property yang sudah dimiliki dan juga tidak memiliki property untuk dijual. Maka player tersebut akan bangkrut.

## **Pembelian Rumah**

Untuk melakukan pembelian rumah, pemain cukup mendapatkan 1 kompleks terlebih dahulu kemudian nanti ketika pemain berada pada salah satu petak yang telah ia punyai maka otomatis akan ada pilihan mau beli rumah/tidak. Pembelian rumah hanya bergantung pada sudah punya 1 kompleks/belum dan hanya ketika pemain tersebut berada di petak itu sehingga dimungkinkan satu daerah tertentu bisa memiliki 3 rumah sekaligus tetapi daerah lainnya yang masih ada dalam satu kompleks belum memiliki rumah.

## **Program Utama**

Program utama pada permainan **Monopolee** ini akan berupa sebuah window baru yaitu **Menu Page** yang dapat memilih untuk bermain, melihat bantuan, dan juga keluar. Bila pemain memilih untuk bermain maka nanti akan disuruh untuk menuliskan nama pemain. Setelah menuliskan nama pemain, permainan sudah berjalan dan pemain pertama tinggal roll dadu untuk memulai permainan.

## ***Class Diagram***

Karena adanya keterbatasan tempat, maka class diagram akan dilampirkan di bagian lampiran dokumen ini.

## **Deskripsi *Class Diagram* (Dokumentasi Kelas yang Dibuat)**

|  |  |  |
| --- | --- | --- |
| ***Name*** | Tile | |
| ***Description*** | Merupakan sebuah interface class | |
| ***Attributes*** | - | |
| ***Methods*** | isOwnable() | Mengecek apakah tile dapat dibeli/rent |
| getOwner() | Mengembalikan Player yang memiliki tile |
| getPosition() | Mengembalikan posisi tile |
| handlePlayer(Player p) | Memproses kondisi Player ketika berada pada tile |
| handleBuyPlayer(Player p) | Memproses kondisi Player ketika Player ingin membeli tile |
| getName() | Mengembalikan nama tile |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | Player | |
| ***Description*** | Merupakan sebuah class | |
| ***Attributes*** | name, balance, cursor, jailcard, lots, utilities, railroads, diJail, countDiJail, countMagenta, countCyan, countPink, countOrange, countRed, countYellow, countGreen, countBlue, bangkrut | |
| ***Methods*** | Player(String name, int balance, int cursor) | Konstruktor Player |
| getPlayer() | Mengembalikan player saat ini |
| isBangkrut() | Mengecek apakah seorang player bangkrut/tidak |
| setBangkrut(boolean bangkrut) | Set kondisi suatu player untuk bangkrut/tidak tergantung masukan |
| getName() | Mengembalikan nama player |
| setName(String name) | Set nama player |
| getCursor() | Mengembalikan cursor player saat ini |
| setCursor(int cursor) | Set player cursor spesifik ke suatu tempat di board (0-40) |
| bisaBayar(double expense) | Mengecek apakah seorang player bisa membayar masukan expense/tidak |
| getBalance() | Mengembalikan balance player |
| setBalance(double balance) | Men-set balace player menjadi balance |
| addmoney(double vsmoney) | Menambah uang player sebanyak vsmoney |
| takemoney(double expense) | Membayar expense dengan mengecek apakah uang cukup |
| giveMoney(Player dest, double expense) | Memberikan uang sebanyak expense ke player dest |
| usefreejail() | Mengurangi jailcard player sebanyak 1 |
| addfreejailcard() | Menambah jailcard player sebanyak 1 |
| getJailcard() | Mengembalikan jailcard player |
| setJailcard(int jailcard) | Men-set jailcard player menjadi jailcard |
| isDiJail() | Mengembalikan apakah player berada di jail |
| setDiJail(boolean diJail) | Men-set diJail player menjadi diJail |
| getCountDiJail() | Mengembalikan countDiJail player |
| setCountDiJail(int countDiJail) | Men-set countDiJail player menjadi countDiJail |
| numLots() | Mengembalikan jumlah Lot yang dimiliki player |
| numUtilities() | Mengembalikan jumlah Utility yang dimiliki player |
| numRailroads() | Mengembalikan jumlah Railroad yang dimiliki player |
| addLots(Lot l) | Menambah Lot l ke array list Lot yang dimiliki player |
| addUtilities(Utility u) | Menambah Utility u ke array list Utility yang dimiliki player |
| addRailroads(Railroad r) | Menambah Railroad r ke array list Railroad yang dimiliki player |
| cekLot(String nama) | Mengecek apakah Lot dengan nama tertentu dimiliki oleh player |
| cekUtilities(String nama) | Mengecek apakah Utility dengan nama tertentu dimiliki oleh player |
| cekRailroads(String nama) | Mengecek apakah Railroad dengan nama tertentu dimiliki oleh player |
| getLots(String nama) | Mengembalikan Lot dengan nama tertentu yang dimiliki player |
| getUtilities(String nama) | Mengembalikan Utility dengan nama tertentu yang dimiliki player |
| getRailroads(String nama) | Mengembalikan Railroad dengan nama tertentu yang dimiliki player |
| removeLots(Lot p) | Menghapus Lot p yang dimiliki player |
| removeUtilities(Utility p) | Menghapus Utility p yang dimiliki player |
| removeRailroads(Railroad p) | Menghapus Railroad p yang dimiliki player |
| displayallLots() | Menampilkan seluruh Lot yang dimiliki player |
| displayallUtilities() | Menampilkan seluruh Utility yang dimiliki player |
| displayallRailroads() | Menampilkan seluruh Railroad yang dimiliki player |
| cekSekompleks(Color kompleks) | Mengecek apakah player telah memiliki sekompleks dengan warna kompleks |
| getCountMagenta() | Mengembalikan jumlah kompleks berwarna Magenta yang dimiliki player |
| setCountMagenta(int countMagenta) | Menset jumlah kompleks berwarna Magenta yang dimiliki player menjadi countMagenta |
| getCountCyan() | Mengembalikan jumlah kompleks berwarna Cyan yang dimiliki player |
| setCountCyan(int countCyan) | Menset jumlah kompleks berwarna Cyan yang dimiliki player menjadi countCyan |
| getCountPink() | Mengembalikan jumlah kompleks berwarna Pink yang dimiliki player |
| setCountPink(int countPink) | Menset jumlah kompleks berwarna Pink yang dimiliki player menjadi countPink |
| getCountOrange() | Mengembalikan jumlah kompleks berwarna Orange yang dimiliki player |
| setCountOrange(int countOrange) | Menset jumlah kompleks berwarna Orange yang dimiliki player menjadi countOrange |
| getCountRed() | Mengembalikan jumlah kompleks berwarna Red yang dimiliki player |
| setCountRed(int countRed) | Menset jumlah kompleks berwarna Red yang dimiliki player menjadi countRed |
| getCountYellow() | Mengembalikan jumlah kompleks berwarna Yellow yang dimiliki player |
| setCountYellow(int countYellow) | Menset jumlah kompleks berwarna Yellow yang dimiliki player menjadi countYellow |
| getCountGreen() | Mengembalikan jumlah kompleks berwarna Green yang dimiliki player |
| setCountGreen(int countGreen) | Menset jumlah kompleks berwarna Green yang dimiliki player menjadi countGreen |
| getCountBlue() | Mengembalikan jumlah kompleks berwarna Blue yang dimiliki player |
| setCountBlue(int countBlue) | Menset jumlah kompleks berwarna Blue yang dimiliki player menjadi countBlue |
| IncrementCountKompleks(Color kompleks) | Menambahkan jumlah kompleks yang dimiliki player berdasarkan warnanya |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | Property | |
| ***Description*** | Merupakan sebuah abstract class yang mengimplementasikan Tile | |
| ***Attributes*** | owner, name, position, harga | |
| ***Methods*** | Property(Player owner, String name, int position, int harga) | Konstruktor untuk membentuk objek property |
| isOwnable() | Mengecek apakah property dapat dibeli/sewa |
| getOwner() | Mengembalikan Player yang memiliki tile |
| setOwner(Player owner) | Men-*set* Owner Property dengan owner baru yang ditetapkan |
| getName() | Mengembalikan nama property |
| getPosition() | Mengembalikan posisi property |
| getHarga() | Mengembalikan harga property |
| handlePlayer(Player p) | Memproses kondisi Player ketika Player menempati tile, masih do nothing. Sub-class property yang akan mengimplementasikannya |
| handleBuyPlayer(Player p) | Memproses kondisi Player ketika Player ingin membeli property |
| getSewa(Player owner) | Abstract method yang bertujuan untuk mengembalikan harga property sesuai dengan penetapan harga property owner |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | Space | |
| ***Description*** | Merupakan class yang mengimplementasikan Tile | |
| ***Attributes*** | name, position | |
| ***Methods*** | Space(String name, int position) | Konstruktor untuk membentuk objek space |
| isOwnable() | Mengembalikan false, karena space tidak dapat dibeli |
| getOwner() | Mengembalikan null, karena space tidak dapat dibeli (tidak ada ownernya) |
| getPosition() | Mengembalikan posisi space |
| getName() | Mengembalikan nama space |
| handlePlayer(Player p) | Memproses kondisi Player ketika Player menempati space, masih do nothing. Sub-class space yang akan mengimplementasikannya |
| handleBuyPlayer(Player p) | Berisi do nothing, karena space tidak dapat dibeli |

|  |  |  |
| --- | --- | --- |
| ***Name*** | Lot | |
| ***Description*** | Merupakan class yang meng-extend Property | |
| ***Attributes*** | Kompleks, countRumah, hargaR | |
| ***Methods*** | Lot(Player owner, String name, int position, Color kompleks, int harga, int countRumah, int hargaR) | Konstruktor untuk membentuk objek lot |
| getKompleks() | Mengembalikan warna kompleks lot |
| getCountRumah() | Mengembalikan banyak rumah yang ada di lot |
| setCountRumah(int countRumah) | Men-set banyak rumah yang ada di lot |
| getHargaR() | Mengembalikan harga rumah lot |
| getSewa(Player owner) | Mengembalikan harga sewa lot |
| getJualL(Player p, String namaLot) | Mengembalikan harga jual lot yang bernama namaLot yang dimiliki oleh Player p |
| handlePlayer(Player p) | Memproses kondisi Player ketika Player menempati lot |
| handleBuyPlayer(Player p) | Penanganan tombol beli lot yang ada di interface Monopolee |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | Railroad | |
| ***Description*** | Merupakan class yang meng-extend Property | |
| ***Attributes*** | - | |
| ***Methods*** | Railroad(Player owner, String name, int position, int harga) | Konstruktor untuk membentuk objek railroad |
| getSewa(Player owner) | Mengembalikan harga sewa railroad |
| getJualR(Player p, String namaRailroad) | Mengembalikan harga jual railroad yang bernama namaRailroad yang dimiliki oleh Player p |
| handlePlayer(Player p) | Memproses kondisi Player ketika Player menempati railroad |
| handleBuyPlayer(Player p) | Penanganan tombol beli property yang ada di interface Monopolee |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | Utility | |
| ***Description*** | Merupakan class yang meng-extend Property | |
| ***Attributes*** | - | |
| ***Methods*** | Railroad(Player owner, String name, int position, int harga) | Konstruktor untuk membentuk objek utility |
| getSewa(Player owner) | Mengembalikan harga sewa utility |
| getJualU(Player p, String namaUtility) | Mengembalikan harga jual utilityyang bernama namaUtility yang dimiliki oleh Player p |
| handlePlayer(Player p) | Memproses kondisi Player ketika Player menempati utility |
| handleBuyPlayer(Player p) | Penanganan tombol beli property yang ada di interface Monopolee |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | Chance | |
| ***Description*** | Merupakan class yang meng-extend Space | |
| ***Attributes*** | deck | |
| ***Methods*** | Chance(String name, Deck deck, int Position) | Konstruktor untuk membentuk objek chance |
| handlePlayer(Player p) | Memproses kondisi Player ketika Player menempati chance |
| handleBuyPlayer(Player p) | Penanganan tombol beli property yang ada di interface Monopolee agar tidak dapat membeli chance |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | Deck | |
| ***Description*** | Merupakan sebuah class | |
| ***Attributes*** | totalDeck, cards, gantiDeck, urutan, total, p | |
| ***Methods*** | Deck(String gantiDeck) | Konstruktor untuk membentuk objek deck berdasarkan inputan gantiDecknya, apakah chance deck atau community chest deck |
| getUrutan() | Mengembalikan urutan kartu di deck |
| getTotal() | Mengembalikan total deck |
| getGantiDeck() | Mengembalikan nama deck |
| setGantiDeck(String gantiDeck) | Men-set deck menjadi gantiDeck |
| printCardText() | Menampilkan isi card yang ada di deck |
| shuffleCard() | Mengacak urutan kartu pada deck |
| nextCard(Player p) | Mengarahkan ke kartu selanjutnya pada deck |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | Card | |
| ***Description*** | Merupakan sebuah class | |
| ***Attributes*** | cardText | |
| ***Methods*** | Card() | Konstruktor untuk membentuk objek card |
| executeCard(Player p) | Mengeksekusi perintah card sesuai yang didapatkan oleh player p, do nothing karena akan diemplementasikan di sub-classnya |
| printCardText() | Menampilkan isi card |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | MoneyCard | |
| ***Description*** | Merupakan class yang meng-extend Card | |
| ***Attributes*** | cardText, amount | |
| ***Methods*** | MoneyCard(String cardText, int amount) | Konstruktor untuk membentuk objek moneycard |
| executeCard(Player p) | Mengeksekusi perintah moneycard |
| printCardText() | Menampilkan isi card |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | GoToJailCard | |
| ***Description*** | Merupakan class yang meng-extend Card | |
| ***Attributes*** | cardText | |
| ***Methods*** | GoToJailCard(String cardText) | Konstruktor untuk membentuk objek GoToJailCard |
| executeCard(Player p) | Mengeksekusi perintah GoToJailCard |
| printCardText() | Menampilkan isi card |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | FreeJailCard | |
| ***Description*** | Merupakan class yang meng-extend Card | |
| ***Attributes*** | cardText | |
| ***Methods*** | FreeJailCard(String cardText) | Konstruktor untuk membentuk objek FreeJailCard |
| executeCard(Player p) | Mengeksekusi perintah FreeJailCard |
| printCardText() | Menampilkan isi card |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | CloseRailroadCard | |
| ***Description*** | Merupakan class yang meng-extend Card | |
| ***Attributes*** | cardText | |
| ***Methods*** | CloseRailroadCard(String cardText) | Konstruktor untuk membentuk objek CloseRailroadCard |
| executeCard(Player p) | Mengeksekusi perintah CloseRailroadCard |
| printCardText() | Menampilkan isi card |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | GoToCard | |
| ***Description*** | Merupakan class yang meng-extend Card | |
| ***Attributes*** | cardText, position, board | |
| ***Methods*** | GoToCard(String cardText, int position) | Konstruktor untuk membentuk objek GoToCard |
| executeCard(Player p) | Mengeksekusi perintah GoToCard |
| printCardText() | Menampilkan isi card |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | Board | |
| ***Description*** | Merupakan sebuah class | |
| ***Attributes*** | totalTile, board, chance, communityChest | |
| ***Methods*** | Board() | Konstruktor Board |
| getTotalTile() | Mengembalikan total tile yang ada |
| getTileInfo(int position) | Mengembalikan tile yang berada di position |
| getBoard() | Mengembalikan board |
| makeTile(int position) | Menempatkan tile-tile berdasarkan positionnya masing-masing |
| start(int position) | Menginisiasi start |
| freeParking(int position) | Menginisiasi freeParking |
| goToJail(int position) | Menginisiasi goToJail |
| jail(int position) | Menginisiasi jail |
| chanceTile(int position) | Menginisiasi chanceTile |
| communityChestTile(int position) | Menginisiasi communityChestTile |
| incomeTax(int position) | Menginisiasi incomeTax |
| luxuryTax(int position) | Menginisiasi luxuryTax |
| stKiaracondong(int position) | Menginisiasi railroad stKiaracondong |
| stBandung(int position) | Menginisiasi railroad stBandung |
| stPadalarang(int position) | Menginisiasi railroad stPadalarang |
| stCimahi(int position) | Menginisiasi railroad stCimahi |
| pdam(int position) | Menginisiasi utility pdam |
| pln(int position) | Menginisiasi utility pln |
| cicadas(int position) | Menginisiasi lot cicadas |
| padasuka(int position) | Menginisiasi lot padasuka |
| uber(int position) | Menginisiasi lot uber |
| cicaheum(int position) | Menginisiasi lot cicaheum |
| antapani(int position) | Menginisiasi lot antapani |
| cihapit(int position) | Menginisiasi lot cihapit |
| citarum(int position) | Menginisiasi lot citarum |
| cilaki(int position) | Menginisiasi lot cilaki |
| burangrang(int position) | Menginisiasi lot burangrang |
| kopo(int position) | Menginisiasi lot kopo |
| batununggal(int position) | Menginisiasi lot batununggal |
| braga(int position) | Menginisiasi lot braga |
| merdeka(int position) | Menginisiasi lot merdeka |
| riau(int position) | Menginisiasi lot riau |
| pajajaran(int position) | Menginisiasi lot pajajaran |
| cipaganti(int position) | Menginisiasi lot cipaganti |
| pasirkaliki(int position) | Menginisiasi lot pasirkaliki |
| sarijadi(int position) | Menginisiasi lot sarijadi |
| pasteur(int position) | Menginisiasi lot pasteur |
| sukajadi(int position) | Menginisiasi lot sukajadi |
| dago(int position) | Menginisiasi lot dago |
| ciumbuleiut(int position) | Menginisiasi lot ciumbuleiut |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | CommunityChest | |
| ***Description*** | Merupakan class yang meng-extend Space | |
| ***Attributes*** | deck | |
| ***Methods*** | CommunityChest(String name, Deck deck, int Position) | Konstruktor untuk membentuk objek CommunityChest |
| handlePlayer(Player p) | Memproses kondisi Player ketika Player menempati CommunityChest |
| handleBuyPlayer(Player p) | Penanganan tombol beli property yang ada di interface Monopolee agar tidak dapat membeli CommunityChest |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | FreeParking | |
| ***Description*** | Merupakan class yang meng-extend Space | |
| ***Attributes*** | - | |
| ***Methods*** | FreeParking(String name, int Position) | Konstruktor untuk membentuk objek FreeParking |
| handlePlayer(Player p) | Memproses kondisi Player ketika Player menempati FreeParking |
| handleBuyPlayer(Player p) | Penanganan tombol beli property yang ada di interface Monopolee agar tidak dapat membeli FreeParking |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | GoToJail | |
| ***Description*** | Merupakan class yang meng-extend Space | |
| ***Attributes*** | - | |
| ***Methods*** | GoToJail(String name, int Position) | Konstruktor untuk membentuk objek GoToJail |
| handlePlayer(Player p) | Memproses kondisi Player ketika Player menempati GoToJail |
| handleBuyPlayer(Player p) | Penanganan tombol beli property yang ada di interface Monopolee agar tidak dapat membeli GoToJail |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | IncomeTax | |
| ***Description*** | Merupakan class yang meng-extend Space | |
| ***Attributes*** | - | |
| ***Methods*** | IncomeTax(String name, int Position) | Konstruktor untuk membentuk objek IncomeTax |
| handlePlayer(Player p) | Memproses kondisi Player ketika Player menempati IncomeTax |
| handleBuyPlayer(Player p) | Penanganan tombol beli property yang ada di interface Monopolee agar tidak dapat membeli IncomeTax |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | Jail | |
| ***Description*** | Merupakan class yang meng-extend Space | |
| ***Attributes*** | - | |
| ***Methods*** | Jail(String name, int Position) | Konstruktor untuk membentuk objek Jail |
| handlePlayer(Player p) | Memproses kondisi Player ketika Player menempati Jail |
| handleBuyPlayer(Player p) | Penanganan tombol beli property yang ada di interface Monopolee agar tidak dapat membeli Jail |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | LuxuryTax | |
| ***Description*** | Merupakan class yang meng-extend Space | |
| ***Attributes*** | - | |
| ***Methods*** | LuxuryTax(String name, int Position) | Konstruktor untuk membentuk objek LuxuryTax |
| handlePlayer(Player p) | Memproses kondisi Player ketika Player menempati LuxuryTax |
| handleBuyPlayer(Player p) | Penanganan tombol beli property yang ada di interface Monopolee agar tidak dapat membeli LuxuryTax |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | Start | |
| ***Description*** | Merupakan class yang meng-extend Space | |
| ***Attributes*** | - | |
| ***Methods*** | Start(String name, int Position) | Konstruktor untuk membentuk objek Start |
| handlePlayer(Player p) | Memproses kondisi Player ketika Player menempati Start |
| handleBuyPlayer(Player p) | Penanganan tombol beli property yang ada di interface Monopolee agar tidak dapat membeli Start |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | Dice | |
| ***Description*** | Merupakan sebuah class | |
| ***Attributes*** | diceOne, diceTwo, total | |
| ***Methods*** | Dice() | Konstruktor untuk membentuk objek Dice |
| roll() | Memproses roll dice |
| isDouble() | Mengecek apakah mengeluarkan dadu kembar |
| getDiceOne() | Mengembalikan hasil diceOne |
| getDiceTwo() | Mengembalikan hasil diceTwo |
| setDiceOne(int diceOne) | Men-set diceOne menjadi diceOne yang baru |
| setDiceTwo(int diceTwo) | Men-set diceTwo menjadi diceTwo yang baru |
| getTotal() | Mengembalikan hasil total dadu |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | Timer | |
| ***Description*** | Merupakan class yang meng-extend Thread | |
| ***Attributes*** | Message, cek | |
| ***Methods*** | run() | Menjalankan timer |
| getMessage() | Mengembalikan message timer |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | ArrayListKita<E> | |
| ***Description*** | Merupakan class yang meng-extend AbstractList<E> dan mengimplementasi List<E> | |
| ***Attributes*** | MAX\_ARRAY\_SIZE, EMPTY\_ELEMENTDATA, elementData, size | |
| ***Methods*** | ArrayListKita(int initialCapacity) | Dari dokumentasi java |
| ArrayListKita() | Dari dokumentasi java |
| rangeCheck(int index) | Dari dokumentasi java |
| rangeCheckForAdd(int index) | Dari dokumentasi java |
| outOfBoundsMsg(int index) | Dari dokumentasi java |
| elementData(int index) | Dari dokumentasi java |
| ensureCapacity(int minCapacity) | Dari dokumentasi java |
| grow(int minCapacity) | Dari dokumentasi java |
| hugeCapacity(int minCapacity) | Dari dokumentasi java |
| get(int index) | Dari dokumentasi java |
| size() | Dari dokumentasi java |
| isEmpty() | Dari dokumentasi java |
| add (E e) | Dari dokumentasi java |
| add(int index, E element) | Dari dokumentasi java |
| remove(int index) | Dari dokumentasi java |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | Monopolee | |
| ***Description*** | Merupakan class yang meng-extend java swing | |
| ***Attributes*** | jButton1, jButton2, jButton3, jLabel1, jLabel2 | |
| ***Methods*** | Monopolee() | Menginisiasi halaman depan game |
| initComponents() | Menciptakan semua komponen yang ada di Monopolee |
| jButton1ActionPerformed(java.awt.event.ActionEvent evt) | Meng-handle ketika button start game ditekan untuk memulai game baru |
| jButton2ActionPerformed(java.awt.event.ActionEvent evt) | Meng-handle ketika button bantuan ditekan untuk memberi informasi sekilas |
| jButton3ActionPerformed(  java.awt.event.ActionEvent evt) | Meng-handle ketika button exit ditekan untuk keluar dari game |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | Board\_Game | |
| ***Description*** | Merupakan class yang meng-extend java swing | |
| ***Attributes*** | insets, daftarPemain[], currPlayer, Runtime, p, p2, buttons[][], labels[][], pemain1, pemain2, properti, rollDadu, endTurn, move, sell, properti2, rollDadu2, endTurn2, move2, sell2, timer, timerTimer, dadusatu, dadudua, totaldadu, countDouble, doubledadu, daduku, dadusatu2, dadudua2, totaldadu2, countDouble2, doubledadu2, daduku2, current, finale, xoxo, tidakMenang, statusP1, statusP2, gameLog, buy1, buy2 | |
| ***Methods*** | Board\_Game() | Membuat Board permainan yang berisi UI UI Monopolee maupun objek-objek inisialisasi seperti player, Tile, dll |
| getPosX(cursor : int) : void | Mentranslasi posisi X pada array berdimensi 1 menjadi array berdimensi 2 |
| getPosY(cursor : int) : void | Mentranslasi posisi Y pada array berdimensi 1 menjadi array berdimensi 2 |
| |  | | --- | | addComponent(container : Container,  component : Component, gridx : int, gridy  : int, gridwidth : int, gridheight : int, anchor  : int, fill : int) : void (static) | |  | | Menempelkan pemain 1 dan 2 pada Board permainan |
| moveCard(pemain : Player) : void | Handle pemain ketika mendapatkan kartu untuk berpindah |
| moveJail(pemain : Player) : void | Handle pemain yang masuk penjara dan kondisi-kondisinya |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | Jual\_Properti | |
| ***Description*** | Merupakan class yang meng-extend JFrame | |
| ***Attributes*** | isiNamaProp, namaProp, acceptSell | |
| ***Methods*** | JualProperti(Player p) | Meng-handle jika pemain ingin menjual propertinya |
| JualProperti(Player p, MoneyCard m) | Meng-handle jika pemain tidak mampu membayar denda dari moneycard |
| JualProperti(Player p, Lot l) | Meng-handle jika pemain tidak mampu membayar sewa dari lot player lain |
| JualProperti(Player p, Railroad r) | Meng-handle jika pemain tidak mampu membayar sewa dari railroad player lain |
| JualProperti(Player p, Utility u) | Meng-handle jika pemain tidak mampu membayar sewa dari utility player lain |
| JualProperti(Player p, IncomeTax it) | Meng-handle jika pemain tidak mampu membayar incometax |
| JualProperti(Player p, LuxuryTax lt) | Meng-handle jika pemain tidak mampu membayar luxurytax |

## 

|  |  |  |
| --- | --- | --- |
| ***Name*** | Winning\_Frame | |
| ***Description*** | Merupakan class yang meng-extend java swing | |
| ***Attributes*** | - | |
| ***Methods*** | Winning\_Frame(Player p) | Menampilkan player yang menang dan langsung keluar dari program |

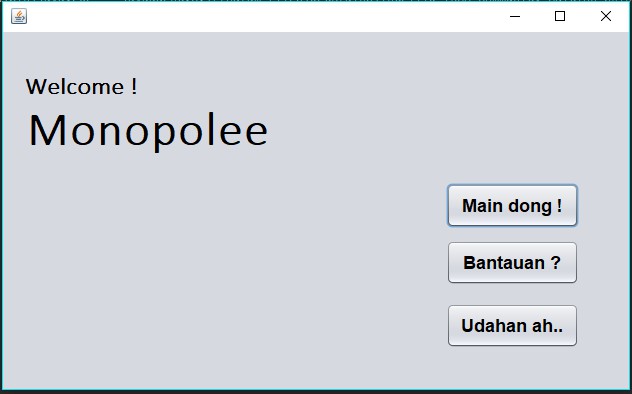
## 

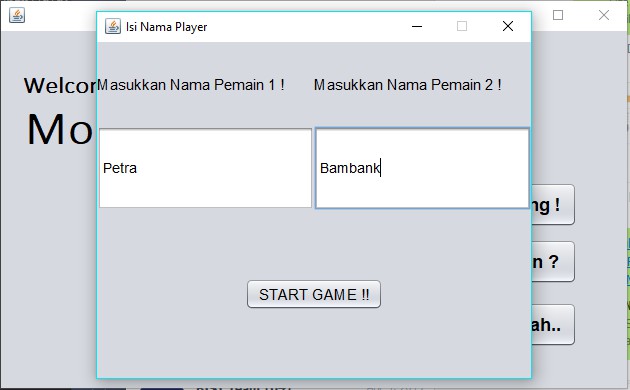
|  |  |  |
| --- | --- | --- |
| ***Name*** | New\_Game\_IsiNama\_2 | |
| ***Description*** | Merupakan class yang meng-extend java swing dan mengimplement ActionListener | |
| ***Attributes*** | isi1, isi2, startGame | |
| ***Methods*** | New\_Game\_IsiNama2() | Menginisiasi halaman pengisian nama pemain game |
| actionPerformance(ActionEvent e) | Penanganan button start game dari halaman pengisian nama |

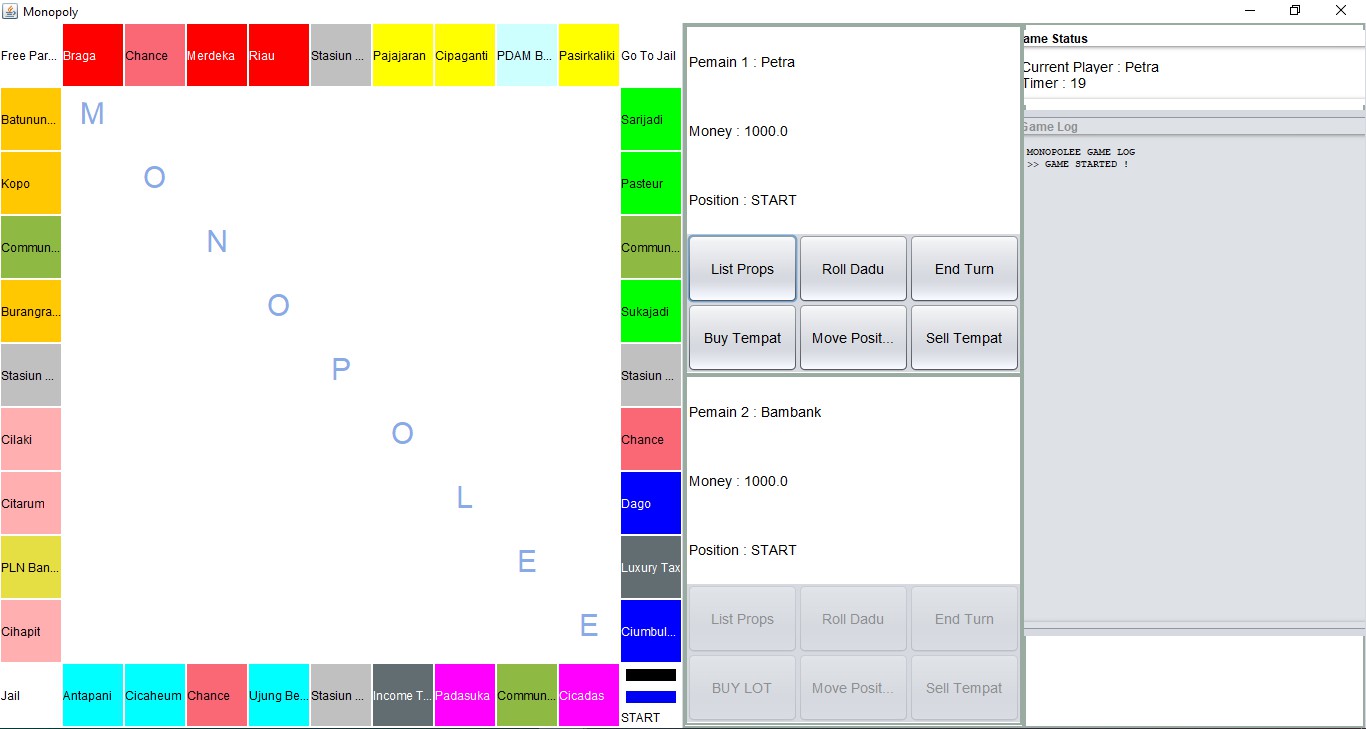
# **Fitur-fitur Monopolee**

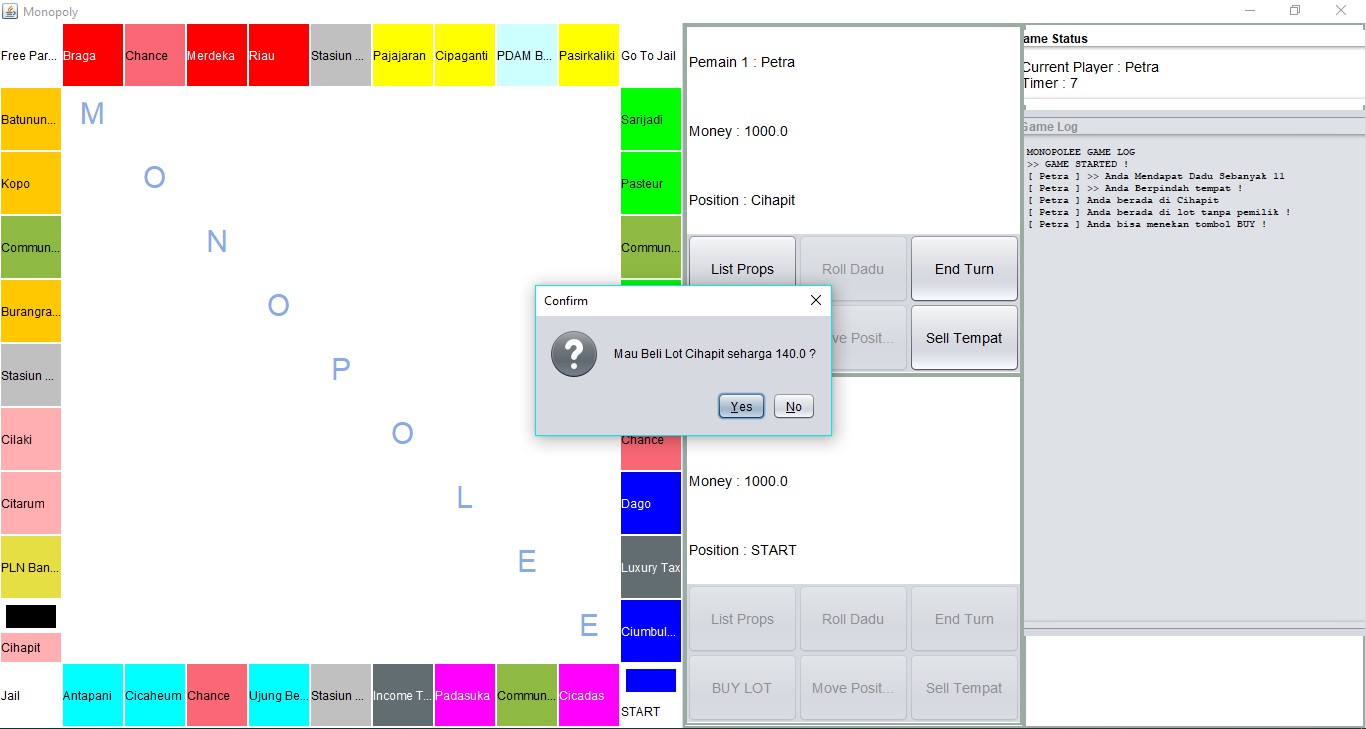
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Fitur yang Dites** | **Tujuan Testing** | **Langkah-Langkah Testing** | **Input Data Test** | **Hasil yang Diharapkan** | **Hasil yang Keluar** |
| 1 | Main Dong | Mengecek apakah jika tombol main dong ditekan maka akan masuk ke tampilan board Monopoly | 1. Menekan tombol Main Dong ! | Menekan tombol Main Dong | Tombol Main Dong ketika ditekan akan terhubung dengan board Monopoly dan game pun akan ,mulai | Tombol Main Dong telah terlaksana degngan baik |
| 2 | Bantuan | Mengecek apakah jika tombol bantuan ditekan maka akan mengeluarkan pop up bantuan permainan Monopoly | 1. Menekan tombol bantuan | Menekan tombol bantuan | Tombol bantuan akan terhubung dengan pop up JOptionPane berisi bantuan | Tombol Bantuan telah terlaksana dengan baik |
| 3 | Udahan ah | Mengecek apakah jika tombol udahan ah ditekan maka akan keluar dari game Monopoly | 1. Menekan tombol udahan ah | Menekan tombol udahan ah | Tombol udahan ah jika ditekan akan keluar dari game | Tombol udahan ah telah terlaksana dengan baik |
| 4 | Fitur Isi Nama Pemain | Mengetahui apakah Pemain 1 dan Pemain 2 telah masuk ke konstruktor Player 1 dan Player 2 | 1. Menjalankan Board\_Game.java 2. Memasukan nama player 1 3. Memasukan nama player2 4. Masuk ke dalam GUI Monopoly 5. Game dapat menampilkan nama player 1 pada kolom yang ada 6. Game dapat menampilkan nama player 2 pada kolom yang ada | 1. Nama Player 1  2. Nama Player 2 | Nama pemain 1 dan Nama pemain 2 berhasil ditampilkan pada GUI Monopoly | 1. Nama player 1 dapat ditampilkan serta digunakan dan dipanggil dalam method lainnya  2. Nama player 2 dapat ditampilkan serta digunakan dan dipanggil dalam method lainnya |
| 5 | Timer | Mengetahui apakah timer 30 sekon pada Monopoly.java berjalan dengan baik | 1. Menjalankan Board\_Game.java  2. Memasukan nama pemain 1 dan nama pemain 2  3. Game akan menampilkan GUI Monopoly  4. Kemudian jika 30 sekon telah habis dan player belum memasukan input command apapun maka akan berganti ke turn player selanjutnya | 1. Input Nama Player | Setelah timer sudah mencapai 0 dan player pertama tidak menginput masukan apapun maka turn berganti ke player selanjutnya | 1.Timer berhasil dijalankan dikarenakansetelah 30 sekon dan pemain tidak menginput apapun maka turn langsung berganti ke player selanjutnya |
| 6 | Roll Dice | Mengetahui apakah command Roll dice pada player 1 dan player 2 dapat mentrigger dice untuk menroll | 1. Menjalankan Board\_Game.java  2. Memasukan nama pemain 1 dan nama pemain 2  3. Game akan menampilkan GUI Monopoly  4. Player satu akan menekan command roll dice  5. Setelah turn player 1 maka player 2 dapat menekan command roll dice | 1.Input nama Player  2.Menekan tombol roll | Ketika tombol ROLL ditekan maka mengocok dadu | Command roll berhasil dijalankan dan mengeluarkan jumlah dadu yang didapat pada log |
| 7 | Move | Mengetahui apakah command move dapat menggerakan player 1 dan player 2 sesuai dengan jumlah total dadu yang telah di roll | 1. Menjalankan Board\_Game.java  2. Memasukan nama pemain 1 dan nama pemain 2  3. Game akan menampilkan GUI Monopoly  4. Player menekan tombol roll  5. Player menekan tombol move | 1.Input nama Player  2.Menekan tombol roll  3. Menekan tombol move | Ketika tombol move player akan bergerak sesuai dengan total dadu yang telah di-roll | Command move berhasil dijalankan dikarenakan dapat menggerakan pemain pada board Monopoly sesuai dengan total dadu yang diperoleh |
| 8 | List Props | Mengetahui apakah jika button listprops ditekan dapat menampilkan list properties yang dimiliki oleh kedua pemain saat ni | 1. Menjalankan Board\_Game.java  2. Memasukan nama pemain 1 dan nama pemain 2  3. Game akan menampilkan GUI Monopoly  4. Player menekan tombol ListProps | 1.Input nama Player  2.Menekan tombol list props | Ketika tombol list player ditekan maka dapat mengeluarkan list properties yang dimiliki oleh player1 atau player 2 saat ini | Command list props berhasil dijalankan dikarenakan bisa mengeluarkan list props apa saja yang dimiliki oleh player saat ini |
| 9 | Buy | Mengetahui apakah jika player berada properties yang belom dibeli maka dapat menekan button buy untuk membeli properties tersebut | 1. Menjalankan Board\_Game.java  2. Memasukan nama pemain 1 dan nama pemain 2  3. Game akan menampilkan GUI Monopoly  4. Player menekan tombol roll  5. Player menekan tombol move  6. Player menekan tombol buy | 1.Input nama Player  2.Menekan tombol roll  2.Menekan tombol move  2.Menekan tombol buy | Ketika tombol move player akan bergerak sesuai dengan total dadu yang telah di-roll | Command buy berhasil dijalankan dikarenakan dapat memberi fitur bagi player untuk membeli  properties |
| 10 | Sell | Mengetahui apakah jika button sell ditekan dapat menampilkan frame JProperti dan player dapat melihat properties apa saja yang dimiliki dan menulis nama properti yang ingin dijual | 1. Menjalankan Board\_Game.java  2. Memasukan nama pemain 1 dan nama pemain 2  3. Game akan menampilkan GUI Monopoly  4. Player menekan tombol roll  5. Player menekan tombol move  6. Jika player berada pada properties yang bisa dibeli player dapat menekan tombol buy | 1.Input nama Player  2.Menekan tombol roll  3.Menekan tombol move  4. Menekan tombol sell | Ketika tombol sell ditekan player dapat melihat list properties yang dipunya dan mengisi nama properties yang ingin dijual | Command sell berhasil dijalankan karena dapat memberi fitur bagi pemain untuk menjual properties yang dimiliki |
| 11 | Penjara | Mengetahui apakah game dapat mengatasi kondisi keluar dari penjara dan masuk ke penjara  . Player dapat masuk ke penjara jika mendapat dadu double lebih dari 3x , mendapatkan kartu GoToJail pada tile 38. Player dapat keluar dari penjara dengan membayar denda sebesar 100, menggunakan kartu free jailcard serta mendapat double ketika mengocok dadu | 1. Menjalankan Board\_Game.java  2. Memasukan nama pemain 1 dan nama pemain 2  3. Game akan menampilkan GUI Monopoly  4. Player menekan tombol roll  5. Player menekan tombol move  6. Jika player masuk jail maka akan keluar pop up nama player masuk penjara  7. Kemudian akan muncul pop up command command yang dapat digunakan untuk keluar dari penjara | 1.Input nama Player  2.Menekan tombol roll  3.Menekan tombol move  4. Menekan bayar / kartu / double | Ketika command buy ditekan dapat mengecek apakah uang player cukup, jika tidak cukup akan looping ke pop up command awal kalo cukup maka player keluar dari penjara. Command kartu ditekan maka dapat mengecek apakah player memiliki kartu free jailcard, jika tidak maka akan looping ke command awal, jika memiliki maka akan keluar dari penjara. Ketiga jika player roll dadu lagi jika double maka akan keluar dari penjara jika tidak mendapat double maka akan tetap di penjara | Kondisi penjara dapat dihandle oleh game Monopoly |
| 12 | Bayar Denda | Mengetahui apakan player terkena denda jika berada pada property milik player lainnya | 1. Menjalankan Board\_Game.java  2. Memasukan nama pemain 1 dan nama pemain 2  3. Game akan menampilkan GUI Monopoly  4. Player menekan tombol roll  5. Player menekan tombol move  6. Jika player berada pada properties milik player 2 maka player 1 harus membayar denda  7. Jika balance uang player lebih kecil dari denda yang harus dibayar maka otomatis akan masuk ke dalam pop up sell properties  8. Jika player tidak memiliki properties sama sekali maka akan keluar message bahwa player kalah | 1.Input nama Player  2.Menekan tombol roll  3.Menekan tombol move  5..Input Property yang ingin dijual (jika balance < denda sewa) | Ketika player berada di properti player lain maka player akan dicek balancenya , jika cukup maka akan dikurangi sesuai dengan denda sewa, tetapi jika kurang maka player akan menjual properties sesuai dengan input property yang ingin dijual, dan jika player tidak mempunyai property maka player akan bangkrut dan game selesai | Balance player berhasil di charge sesuai dengan denda sewa properti tersebut, dan jika balance player kurang dari apa yang harus dibayar maka player akan menjual properties nya sesuai dengan input property yang ingin dijual, dan jika player tidak mempunyai property maka player akan bangkrut dan game selesai |
| 13 | Mendapatkan kartu Chance Card | Jika player berada di Tile =7 player akan mendapat kartu Chance Card | 1. Menjalankan Board\_Game.java  2. Memasukan nama pemain 1 dan nama pemain 2  3. Game akan menampilkan GUI Monopoly  4. Player menekan tombol roll  5. Player menekan tombol move  6. Jika player berada pada tile Chance maka akan mendapat kartu Chance Card | 1 Input nama player  2 Menekan tombol roll dan move | Ketika player mendapat Chance Card maka akan diimplementasikan sesuai dengan action masing masing kartu | Kartu Chance Card berhasil dieksekusi dikarenakan ketika player mendapatkan kesempatan mengambil Chance Card contohnya ”Kamu memenangkan kontes kecantikan kamu mendapat 150” maka balance player tersebut akan ditambah 150 |
| 14 | Mendapatkan kartu CommunityChest | Jika player berada di Tile =2,17 dan 33 player akan mendapat kartu Community Chest Card | 1. Menjalankan Board\_Game.java  2. Memasukan nama pemain 1 dan nama pemain 2  3. Game akan menampilkan GUI Monopoly  4. Player menekan tombol roll  5. Player menekan tombol move  6. Jika player berada pada tile Community Chest maka akan mendapat kartu Community Chest | 1 Input nama player  2 Menekan tombol roll dan move | Ketika player mendapat Community Chest maka akan diimplementasikan sesuai dengan action masing masing kartu | Kartu Community Chest berhasil dieksekusi dikarenakan ketika player mendapatkan kesempatan mengambil Community Chest contohnya ”Tagihan asuranasinya membludak 100” maka balance player tersebut akan dikurang 100 |
| 15 | Mendapatkan kartu GotoJail | Jika player berada di Tile =38 player akan mendapat kartu GotoJail | 1. Menjalankan Board\_Game.java  2. Memasukan nama pemain 1 dan nama pemain 2  3. Game akan menampilkan GUI Monopoly  4. Player menekan tombol roll  5. Player menekan tombol move  5. Jika player berada pada tile tersebut maka player akan dipindahkan ke tile penjara dan status setDiJail menjadi true | 1 Input nama player  2 Menekan tombol roll dan move | Ketika player mendapat GotoJail maka player akan ”ditahan” (dipindahkan ke tile penjara) dan setDiJail = true | Kartu GotoJail berhasil dieksekusi dikarenakan ketika player mendapatkan kartu tersebut maka player akan berada di penjara dan status diJail menjadi true |
| 16 | Mendapatkan kartu FreeParking | Jika player berada di Tile =30 player akan mendapat kartu FreeParking | 1. Menjalankan Board\_Game.java  2. Memasukan nama pemain 1 dan nama pemain 2  3. Game akan menampilkan GUI Monopoly  4. Player menekan tombol roll  5. Player menekan tombol move 6. Player balance ditambah 100 | 1 Input nama player  2 Menekan tombol roll dan move | Ketika player mendapat FreeParking maka balance player akan bertambah 100 | Kartu Chance Card berhasil dieksekusi dikarenakan ketika player mendapatkan kartu ini maka balance akan bertambah 100 |
| 17 | Mendapatkan kartu Incometax | Jika player berada di Tile =7,22 dan 36 player akan mendapat kartu IncomeTask | 1. Menjalankan Board\_Game.java  2. Memasukan nama pemain 1 dan nama pemain 2  3. Game akan menampilkan GUI Monopoly  4. Player menekan tombol roll  5. Player menekan tombol move 6. Balance player dicharge 200 | 1 Input nama player  2 Menekan tombol roll dan move | Ketika player mendapat kartu Incometax maka akan diimplementasikan sesuai dengan action masing masing kartu | Kartu ini berhasil dieksekusi dikarenakan ketika player mendapatkan kartu ini maka balance player dicharge 200 |
| 18 | Mendapatkan kartu LuxuryTax | Jika player berada di Tile =38 player akan mendapat kartu LuxuryTask | 1. Menjalankan Board\_Game.java  2. Memasukan nama pemain 1 dan nama pemain 2  3. Game akan menampilkan GUI Monopoly  4. Player menekan tombol roll  5. Player menekan tombol move  6. Balance player dicharge 75 | 1 Input nama player  2 Menekan tombol roll dan move | Ketika player mendapat LuxuryTax maka akan balance akan dicharge 75 | Kartu berhasil dieksekusi balance pemain di kurangi 75 |
| 19 | Melewati Start | Jika player melewat Start maka akan diberikan tambahan uang sebesar 200 | 1. Menjalankan Board\_Game.java  2. Memasukan nama pemain 1 dan nama pemain 2  3. Game akan menampilkan GUI Monopoly  4. Player menekan tombol roll  5. Player menekan tombol move  6. Jika player melewati start ketika dia di roll maka balance akan bertambah 200 | 1 Input nama player  2 Menekan tombol roll dan move | Ketika player melewati Start maka balance akan bertambah sebesar 200 | Fitur melewati Start dapat berjalan dengan baik karena jika player melewati START akan mendapat 200 |
| 20 | Bangkrut | Kondisi end game dari monopoly ini yaitu jika salah satu pemain balancenya sudah tidak bisa membayar utang dan properties yang dimilikinya sudah habis | 1. Menjalankan Board\_Game.java  2. Memasukan nama pemain 1 dan nama pemain 2  3. Game akan menampilkan GUI Monopoly  4. Player memainkan gamenya  6. Player 1 harus membayar utang sewa tetapi balancenya sudah < dari utang sewa dan properties yang dimiliki sudah habis  7.Game akan selesai dan player 2 menang | 1 Input nama player  2 Menekan tombol roll dan move | Ketika player sudah tidak bisa membayar utang dan properties yang dimiliki sudah habis | End Game berhasil dibuat dikarenakn ketika balance player habis dan properties yang dimiliki habis maka player yang lain akan menang |

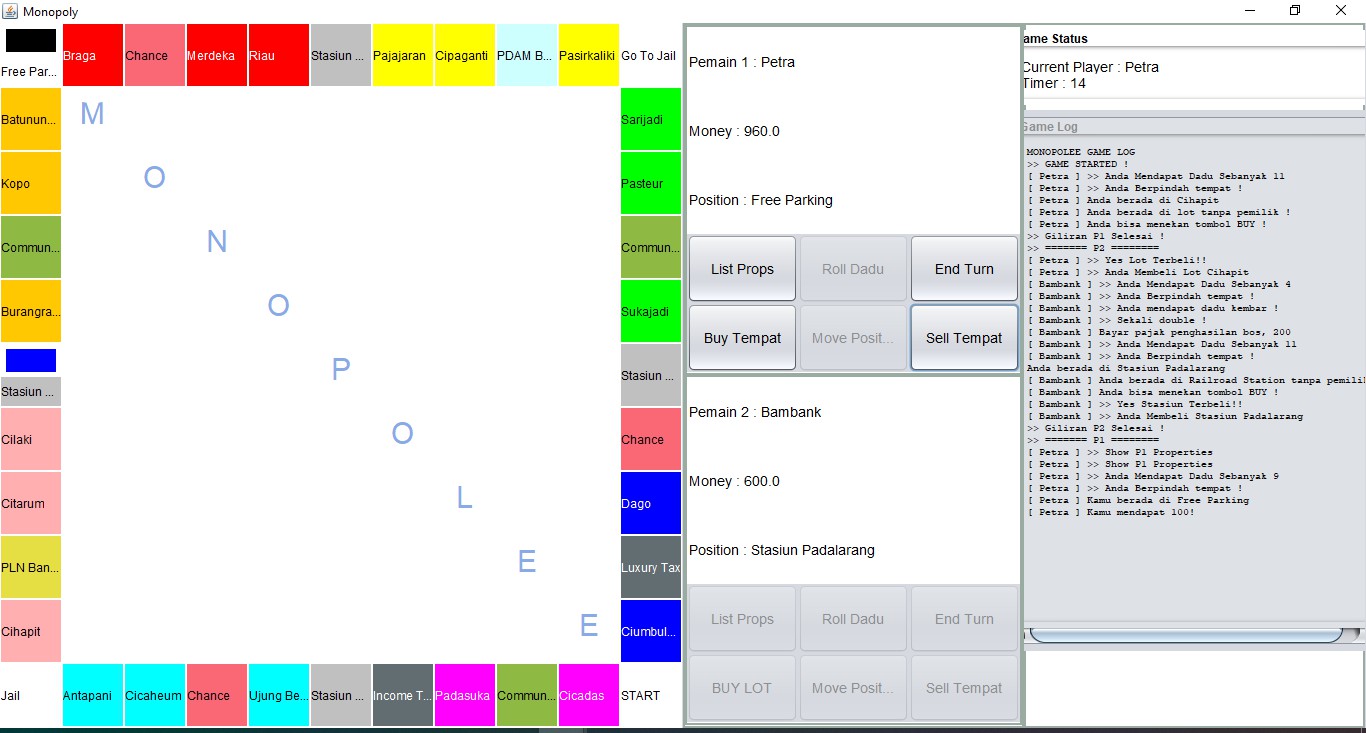
# ***Screenshot Program***

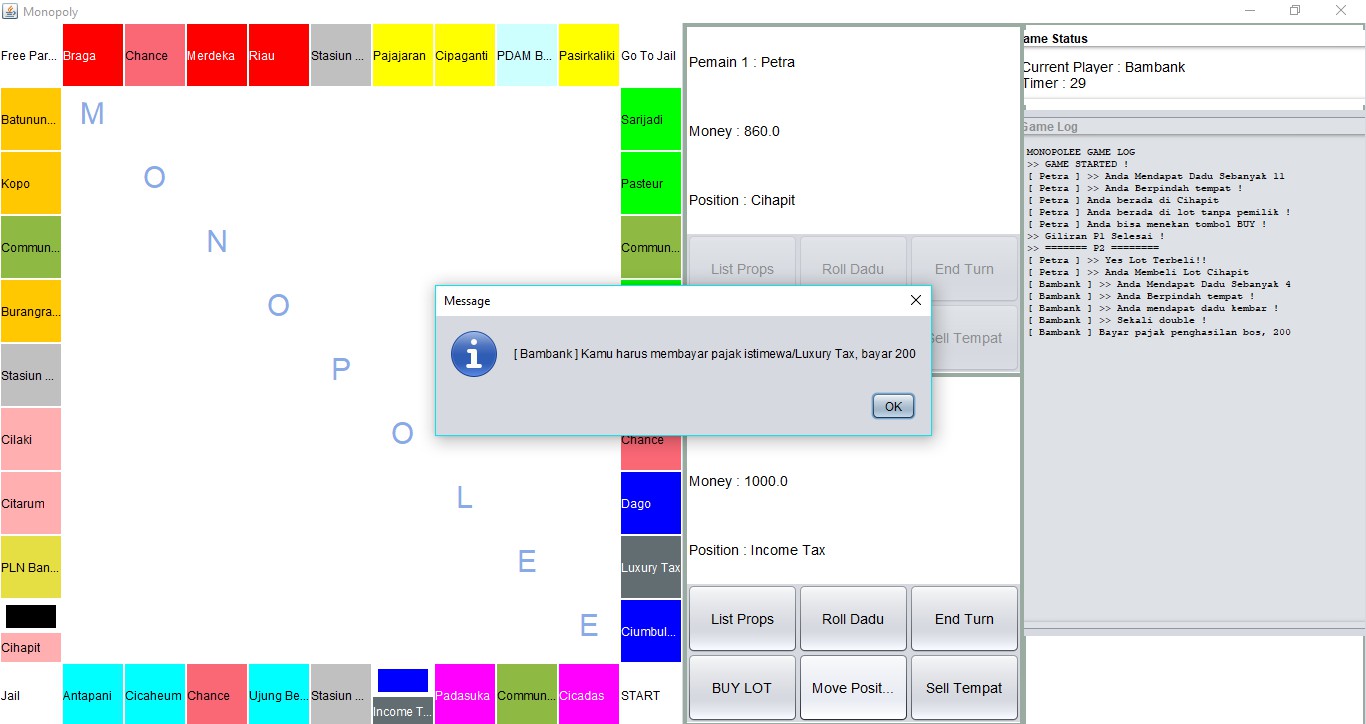


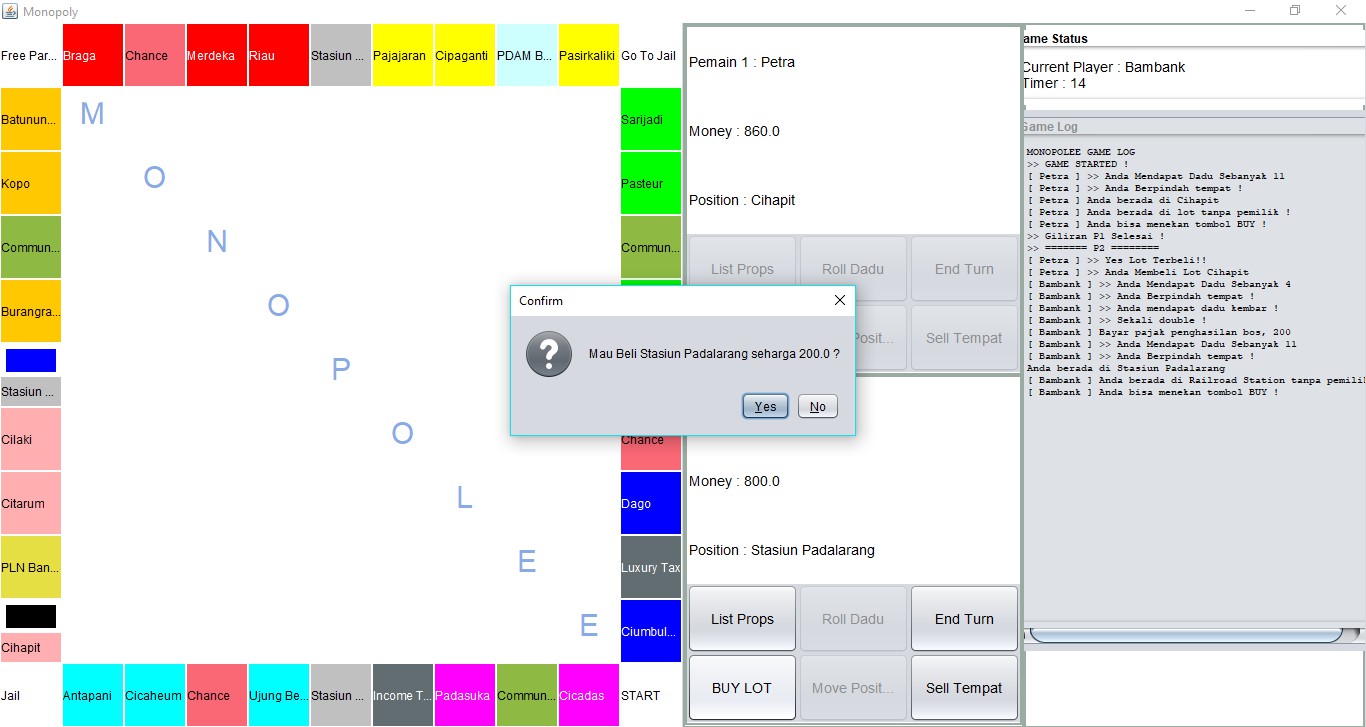


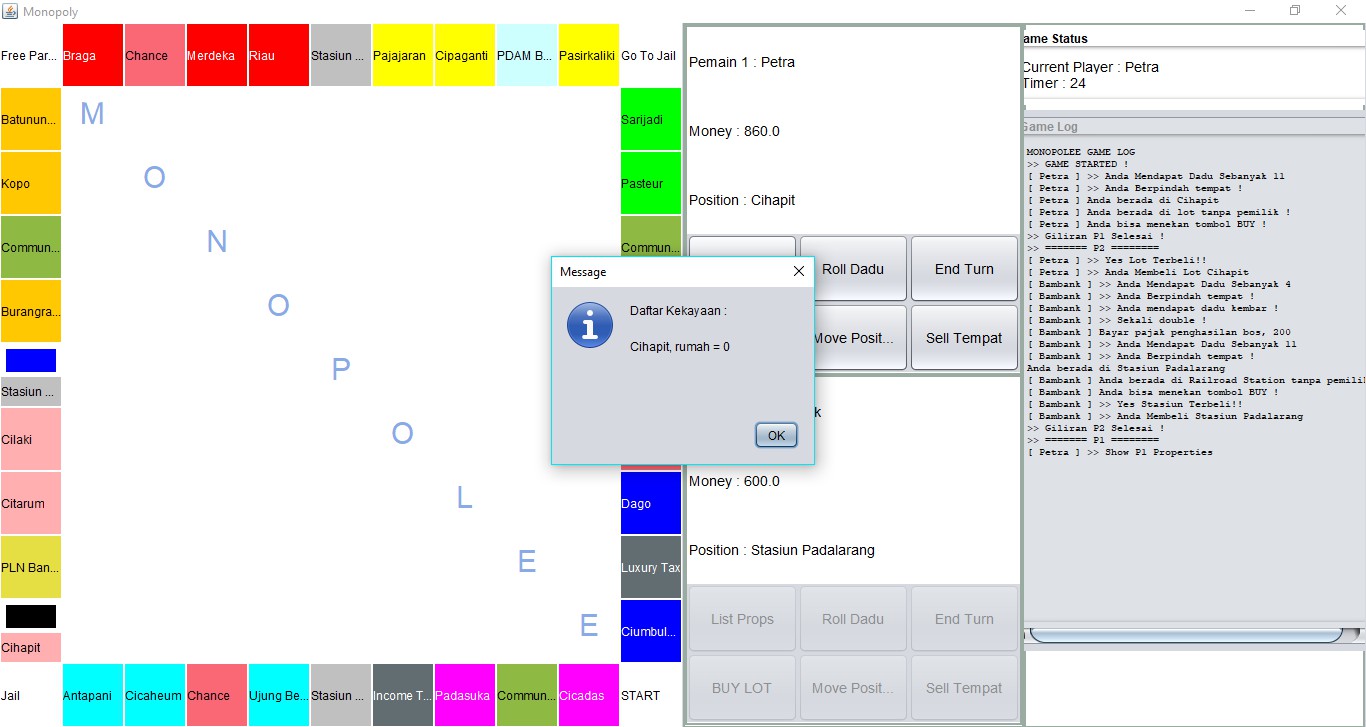


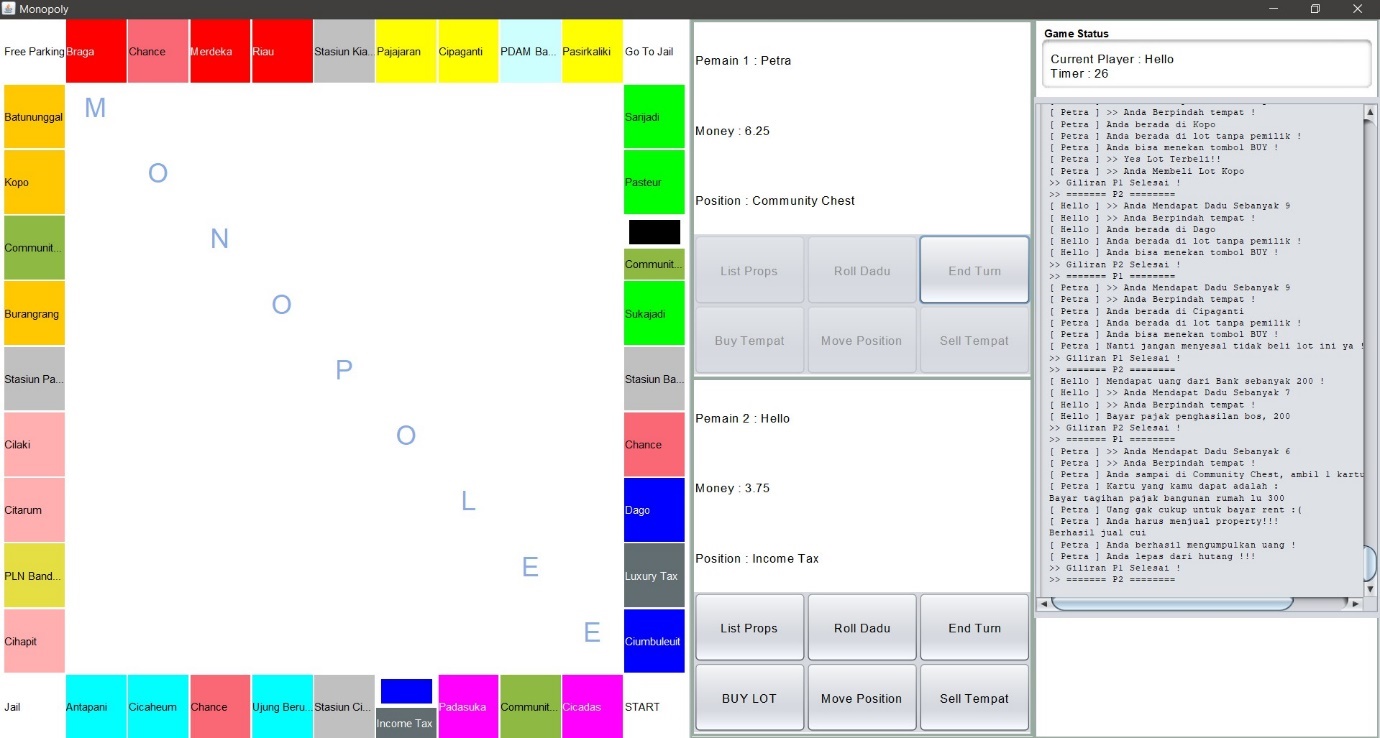


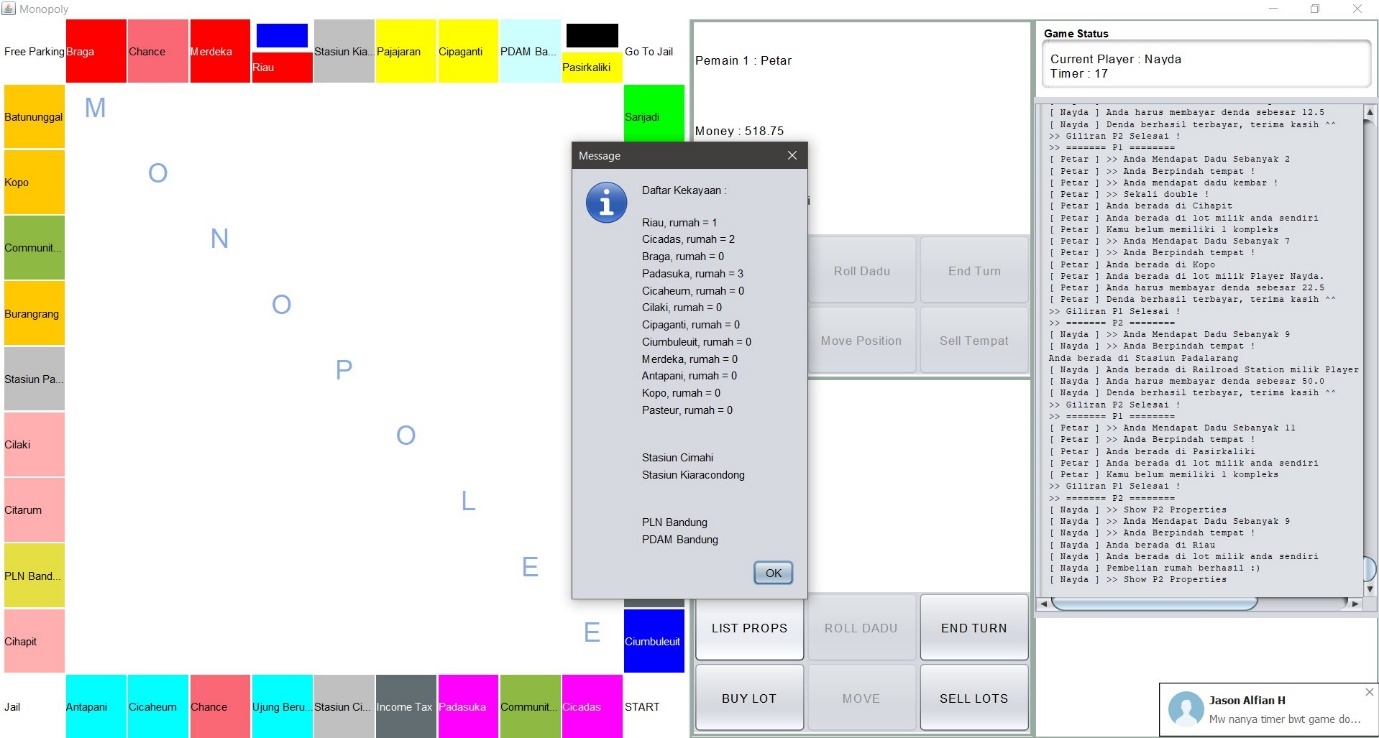


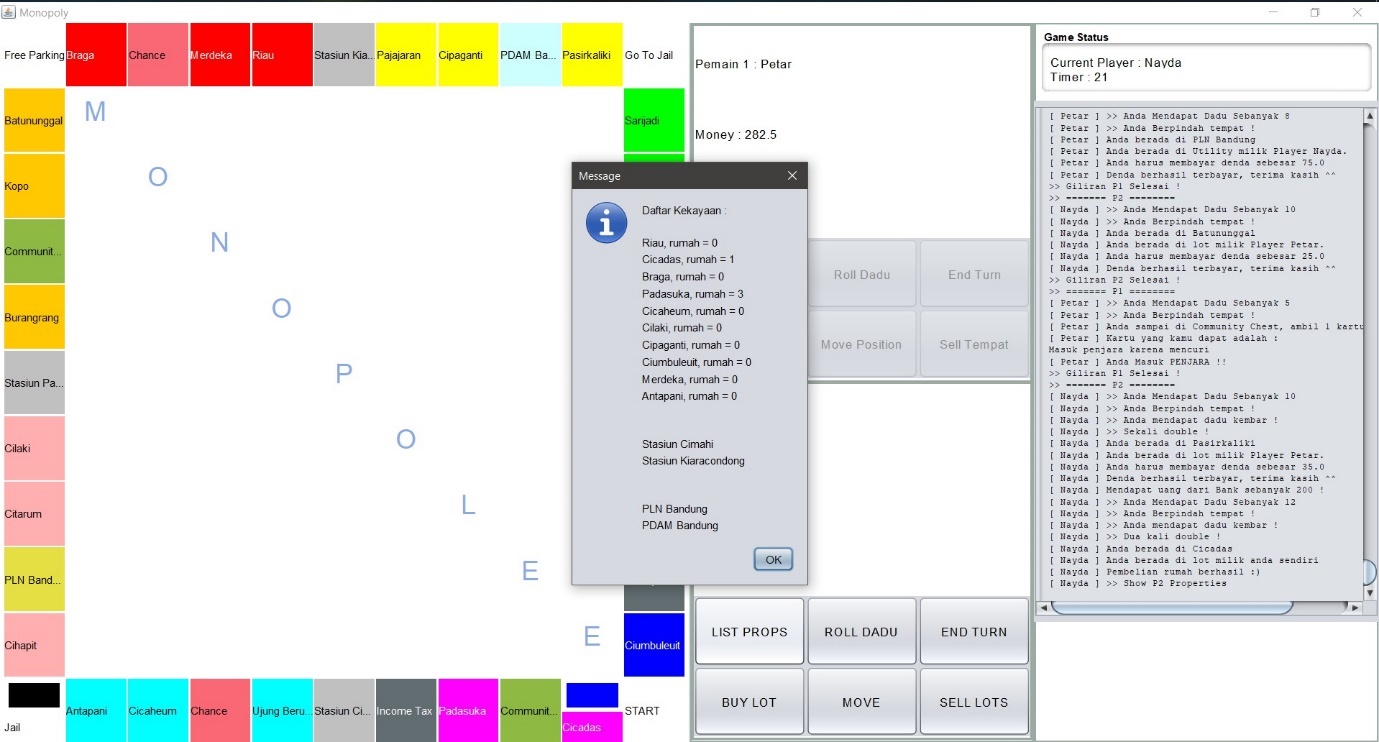


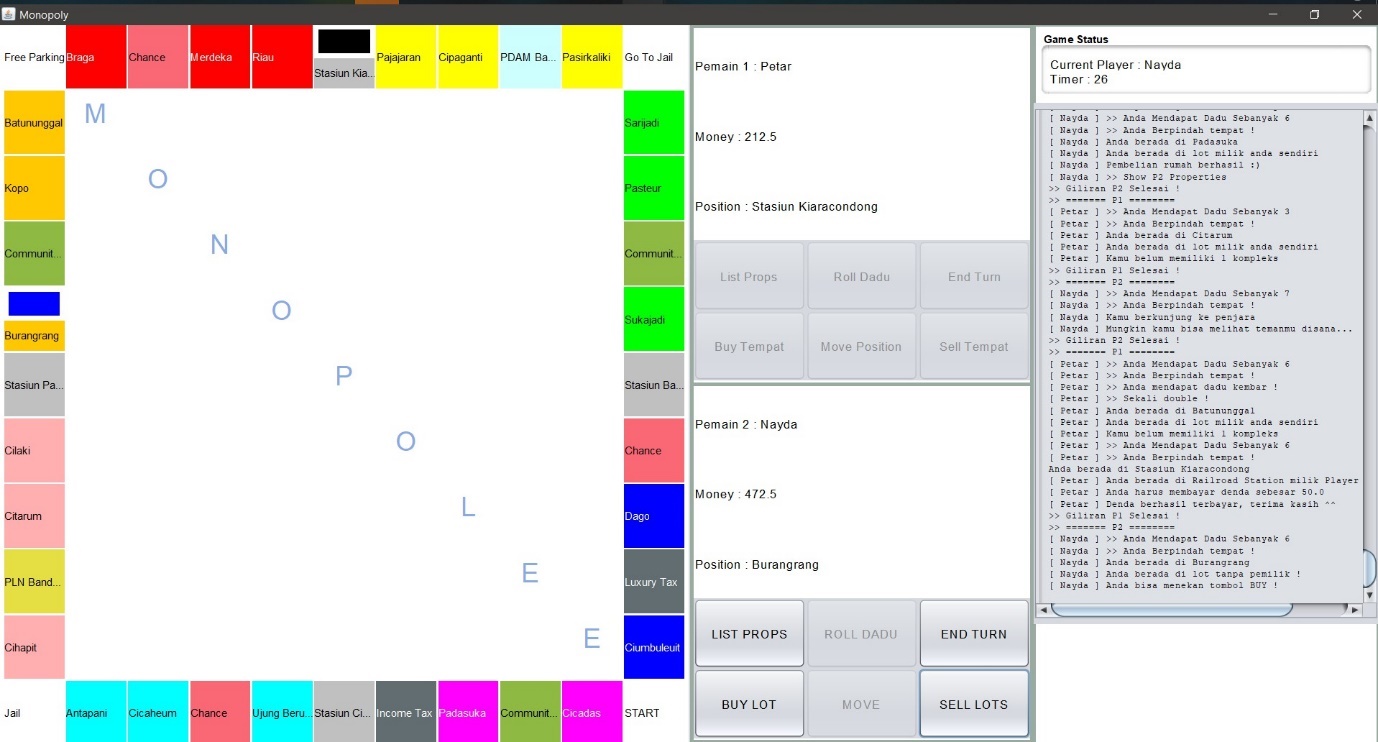


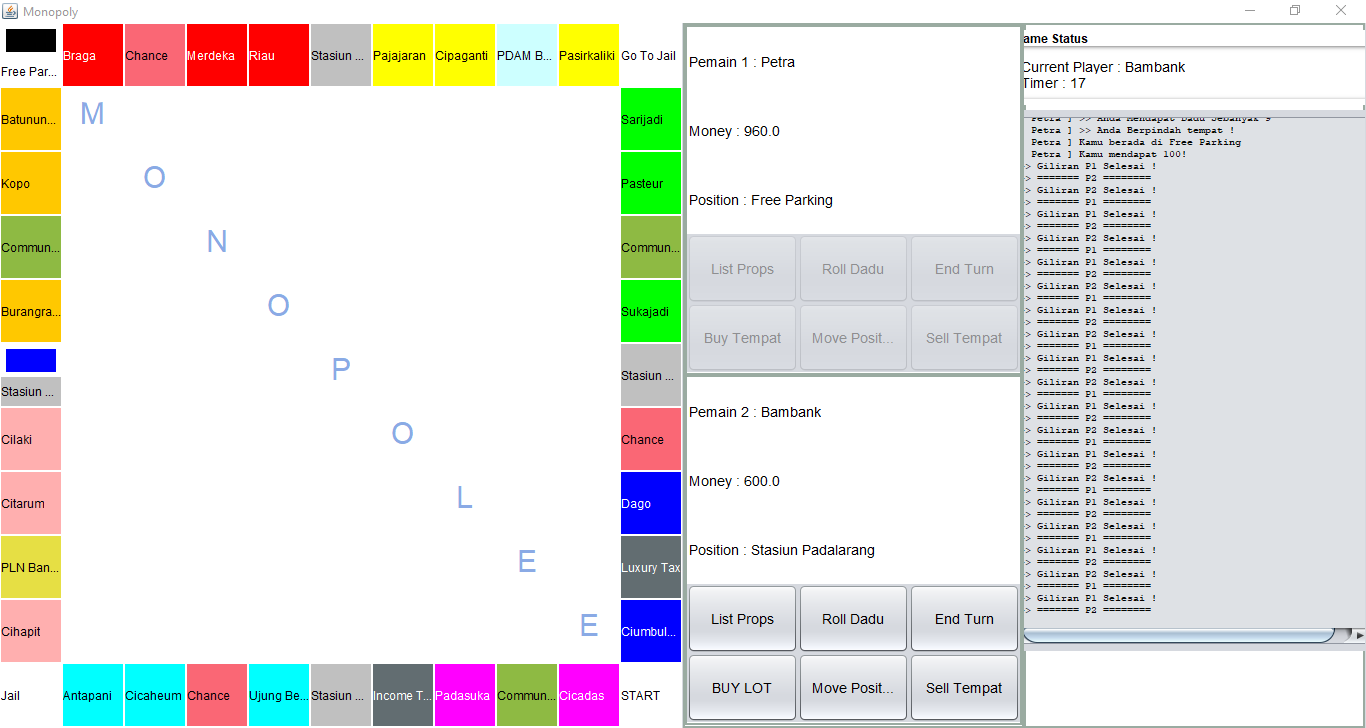


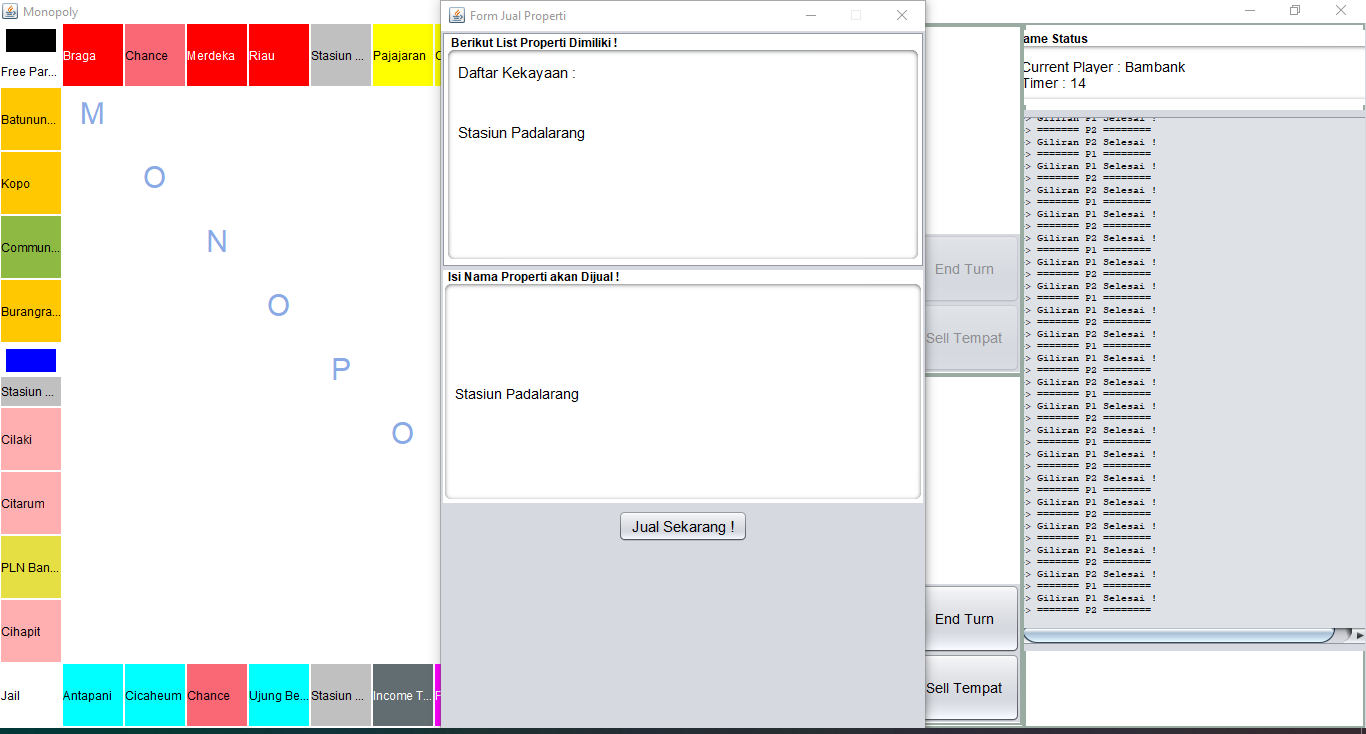


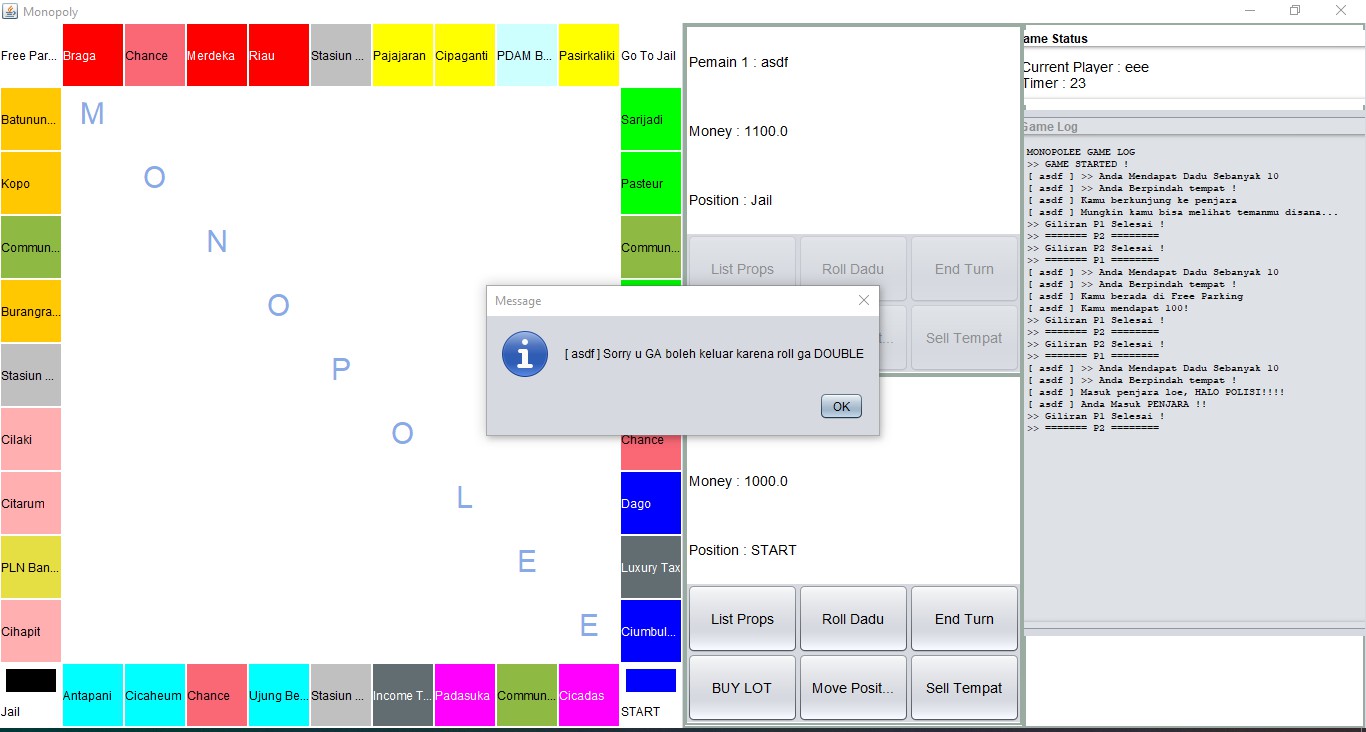
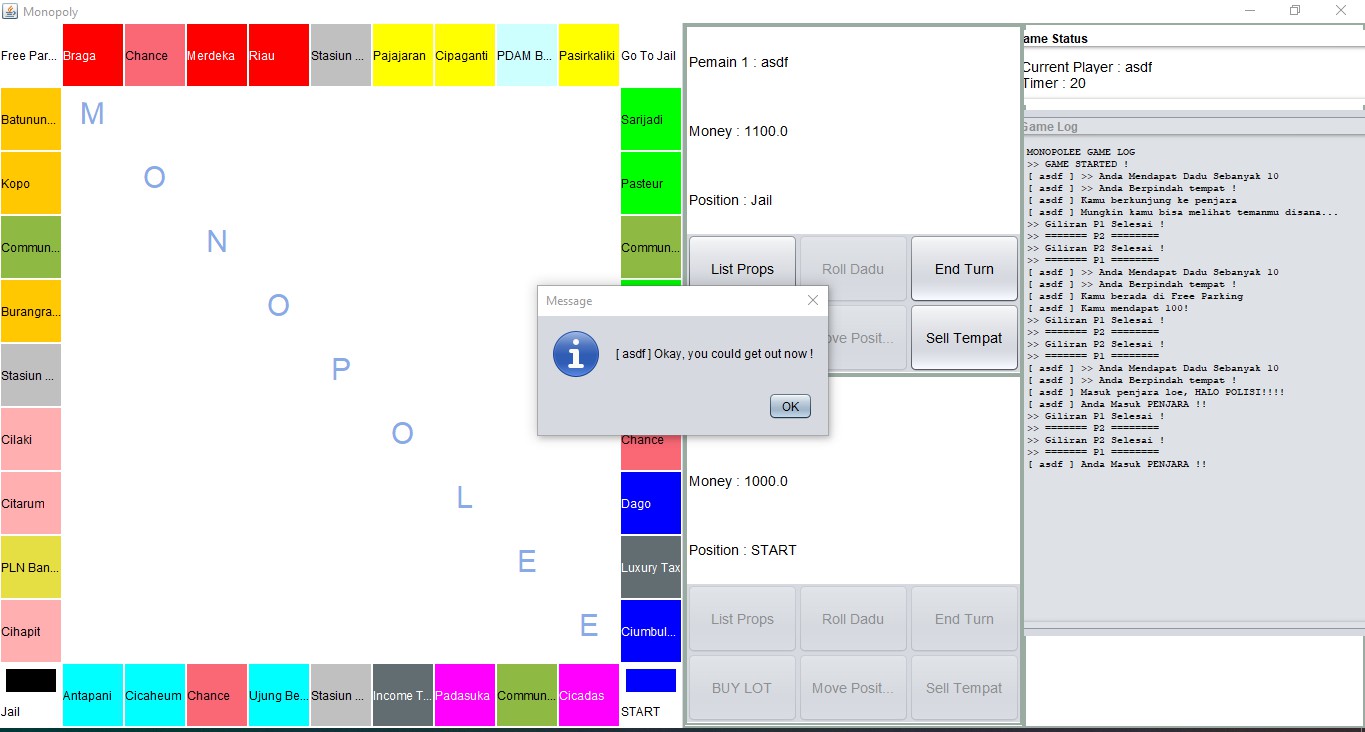
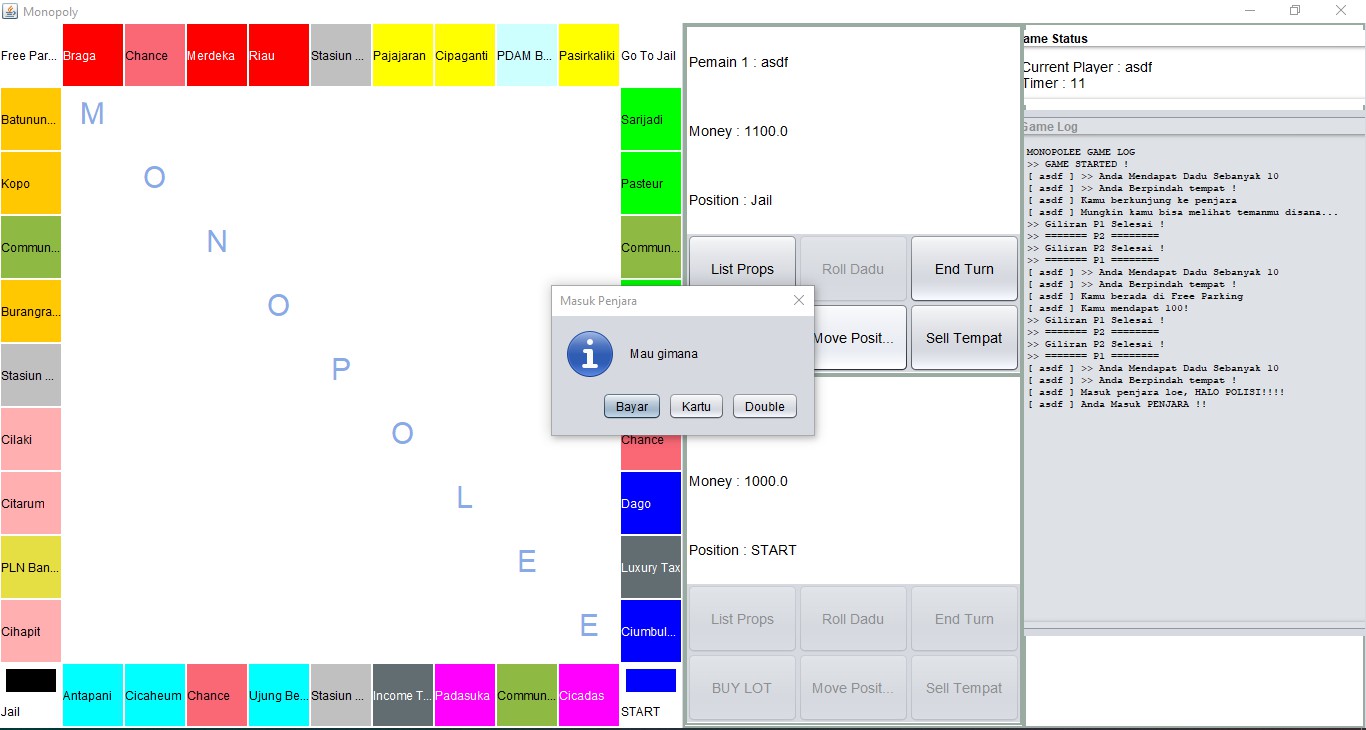
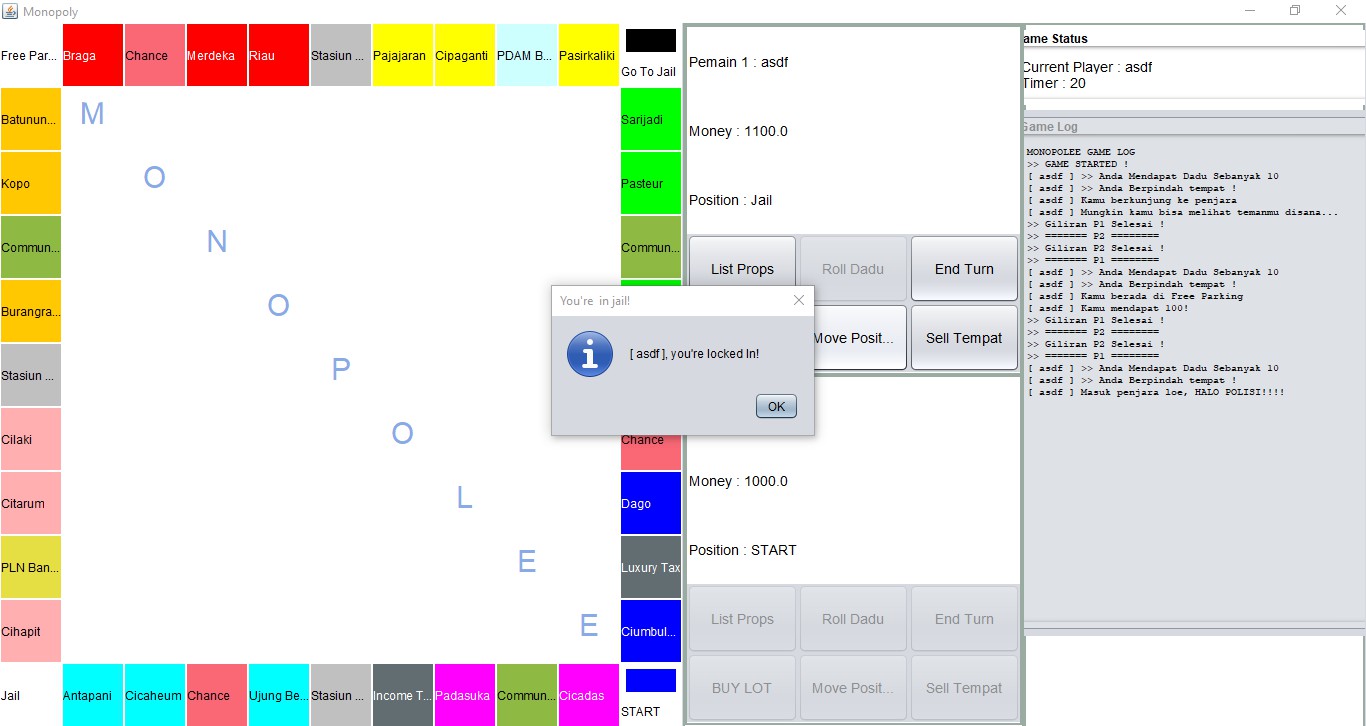












# **Letak Implementasi Cakupan Tugas**

Berikut ini adalah letak implementasi cakupan tugas yang ada pada tugas besar ini.

|  |  |  |
| --- | --- | --- |
| No | Nama Tugas/Deliverable | Letak Cakupan |
| 1 | Inheritance | 1. Superclass : Property.java  Subclass : Lot.java, Railroad.java, Utility.java 2. Superclass : Space.java Subclass : Chance.java, CommunityChest.java, Start.java, Jail.java, FreeParking.java, GoToJail.java, IncomeTax.java, LuxuryTax.java 3. Superclass : Card.java Subclass : MoneyCard.java, GoToCard.java, GoToJailCard.java, FreeJailCard.java, CloseRailroadCard.java |
| 2 | Polymorphism | Implementasi ada pada Deck.java yang memanfaatkan file Card.java, MoneyCard.java, GoToCard.java, GoToJailCard.java, FreeJailCard.java, CloseRailroadCard.java |
| 3 | Abstract Class/Interface | Abstract Class : Property.java  Interface : Tile.java yang diimplementasi oleh Property.java dan Space.java |
| 4 | Generic Class | Pembuatan class baru sebagai arraylist yaitu ArrayListKita.java yang menampung Property milik pemain berdasarkan tipenya yaitu Lot, Railroad, atau Utility |
| 5 | Java Swing | Board\_Game.java Monopolee.java Jual\_Properti.java, New\_Game\_IsiNama2.java Winning\_Frame.java |

# **Pembagian Kerja dalam Kelompok**

Pembagian kerja secara garis besar setiap *member* dalam kelompok ini adalah sebagai beriku. **Untuk detail lebih jelas** mengenai pekerjaan akan ada pada bagian **Lampiran** dokumen ini dalam subbab *Log Activity* Kelompok.

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama** | **NIM** | **Pekerjaan** |
| 1 | Fadel Nararia Rahman | 18217005 | Tugas 1: Membuat GUI **Monopolee** dengan Java Swing mencakup pengembangan UI dan UX serta beberapa fitur dasar game Monopolee.  Tugas 2 : Melanjutkan GUI **Monopolee** dengan Java Swing dengan mengembangkan seluruh UI dan UX game beserta mengintegrasikan dan konversi logic game **Monopolee** yang telah dibuat pada tugas 1 pada tugas 2 dengan Swing |
| 2 | David Pe  tra Natanael | 18217011 | Tugas 1 : Membuat Board permainan, membuat Deck, Card, dan jenis-jenisnya beserta anak-anak Space  Tugas 2 : Membantu testing dan debugging dalam bagian logic GUI **Monopolee,** membantu pembuatan laporan dan class diagram |
| 3 | Nadya Anastasia | 18217020 | Tugas 1 : Membuat Player, Dice,Timer dan membuat serta merevisi laporan  Tugas 2 : Membantu testing dan debugging logic GUI Monopolee.java dan membuat fitur kondisi double dadu serta keluar dari penjara pada Board\_Game.java |
| 4 | Lidya Jessica | 18217037 | Tugas 1 : Membuat Property, beserta implementasi anak-anak classnya dan juga Space  Tugas 2 : Membuat laporan bagian deskripsi class diagram |

# **Lampiran**

## **Deskripsi Tugas Besar**

Sebuah tile pada monopoli bisa dikategorikan menjadi 2 tipe, yaitu ​Property ​dan ​Space​. Property adalah tile yang dapat dibeli oleh pemain, dan terdiri atas 3 tipe, yaitu ​Lot​, ​Railroad​, dan ​Utility​. Pada contoh diatas, Borobudur adalah sebuah Lot, Bandara Kemayoran adalah sebuah Railroad, dan Perusahaan Listrik adalah sebuah Utility. Space adalah tiles yang tidak dapat dibeli oleh pemain. Beberapa contoh space adalah Go To Jail space, Free Parking space, Tax space, Chance space, dan Community Chest space .

Monopoli ini dapat dimainkan oleh beberapa orang. Jika pemain mendarat di Chance space atau Community Chest space maka pemain tersebut akan mengambil satu kartu di deck yang bersesuaian dan melakukan instruksi pada kartu tersebut. Implementasi kartu dibebaskan, beberapa contoh kartu yang dapat diimplementasikan seperti pemain mendapatkan uang, pemain pergi ke suatu tile, ataupun pemain harus membayar uang ke bank. Peluang pemain mendapatkan uang lebih besar jika mendarat di Chance space, dan peluang pemain harus membayar uang ke bank lebih besar jika mendarat di Community Chest space.

Pemain harus bergerak dalam batas waktu 30 tick, dan jika pemain tidak melakukan action selama 30 tick, maka giliran pemain tersebut akan hangus dan menjadi giliran pemain selanjutnya. Uang awal pemain dibebaskan, dan setiap kali pemain melewati titik start, maka pemain akan mendapatkan sejumlah uang.

Ketika pemain mengenai Go To Jail space, pemain akan pergi ke Jail space. Untuk pergi dari penjara, pemain harus mendapatkan dadu double atau membayar sejumlah uang. Ketika pemain mengenai Property yang dimiliki pemain lain, maka pemain tersebut harus membayar sejumlah uang. Aturan harganya adalah sebagai berikut :

1. Lot ​: Harga rent (tanpa bangunan) adalah ⅛ harga beli Lot. Jika memiliki sekomplek, maka harganya menjadi ¼. Jika memiliki 1 rumah, harganya menjadi ½ kalinya, 2 rumah menjadi 1 kali harga beli Lot, 3 rumah menjadi 2 kalinya, dan 4 rumah menjadi 4 kali harga beli Lot. Untuk membangun rumah, seorang player harus memiliki Lot lainnya yang terdapat pada 1 komplek.
2. Utilities​: Harga rent jika hanya memiliki 1 jenis adalah ⅛ kali harga beli Lot, sedangkan jika memiliki 2 jenis maka harganya menjadi ½ kali harga beli Lot.
3. Railroad ​: Harga rent jika hanya memiliki 1 jenis adalah ⅛ kali harga beli Lot, 2 jenis menjadi ¼ kali, 3 jenis menjadi ½ kali, dan 4 jenis menjadi 1 kali harga beli Lot.

Jika pemain mendarat di Tax space, maka pemain harus membayar sejumlah uang ke bank.

## ***Log Activity* Anggota Kelompok**

LOG ACTIVITY

|  |  |  |
| --- | --- | --- |
| **Nama Mahasiswa** | : | Nadya Anastasia |
| **NIM** | : | 18217020 |
| **Tugas** | **KEGIATAN** | |
| 1 | Membuat class Player,Dice,Tiimer | |
| Menggabungkan fungsi Timer thread dengan game | |
| Membuat dokumen laporan serta merevisi laporan | |
| Membuat fungsi serta kondisi masuk penjara dan kondisi player boleh keluar penjara | |
| 2 | Membuat fungsi double dadu pada Board\_Game.java | |
| Membuat fungsi keluar dari penjara pada method movejail di Board\_Game.java | |
| Membuat laporan bagian fitur fitur dan merevisi laporan | |

|  |  |  |
| --- | --- | --- |
| **Nama Mahasiswa** | : | David Petra Natanael |
| **NIM** | : | 18217011 |
| **Tugas** | **KEGIATAN** | |
| 1 | Membuat implementasi Card.java beserta anak-anaknya dan juga kumpulan Card (Deck.java) | |
| Membuat Space.java dan anak-anaknya | |
| Testing Main.java dan debugging | |
| Membuat dokumen laporan serta merevisi laporan | |
| 2 | Membantu debugging pada translasi logic file dari file sebelumnya ke pembuatan GUI | |
|  | Membantu pembuatan laporan tugas ke 2 & membuat class diagram | |
|  | Membantu testing program | |

|  |  |  |
| --- | --- | --- |
| **Nama Mahasiswa** | : | Fadel Rahman |
| **NIM** | : | 18217005 |
| **Tugas** | **KEGIATAN** | |
| 1 | Membangun General User Interface (GUI) untuk Game Monopoly dan interaksi user dengan game  dengan konsep Java Swing. (Button, Label, Frame, Panel, Timer on jFrame, Layout Manager, dll) | |
| Menggabungkan *logic game* dari beberapa file .Java dalam *runtime* Game dengan konsep Java Swing (Controller dengan Listener, Action Event, dll) | |
| Memberi Inputan terhadap konstruksi beberapa file class yang akan digunakan dalam keberjalanan game, misal terkait dengan pengaksesan atribut, methods,dll. | |
| 2 | Mengembangkan keseluruhan UI dan UX Game Monopolee pada GUI dengan menggunakan Java Swing, mencakup keseluruhan keberjalanan game dari welcome screen hingga game berakhir dengan dimenangkan oleh pemain tertentu | |
| Melakukan konversi dan integrasi terhadap seluruh logic game Monopolee yang telah dibuat pada tugas 1 sebelumnya menggunakan metode *system.out* menjadi menggunakan Java Swing. Misalkan beberapa kondisi yang dihadapi dalam permainan seperti *turn Player*, dll. | |
| Melakukan proses pengecekan, *testing* terhadap keseluruhan *bug* yang masih terjadi selama tahap pengembangan dan menyelesaikan permasalahan tersebut langsung pada *source code* yang dibuat. | |

|  |  |  |
| --- | --- | --- |
| **Nama Mahasiswa** | : | Lidya Jessica |
| **NIM** | : | 18217037 |
| **Tugas** | **KEGIATAN** | |
| 1 | Mengerjakan Property dan anak-anaknya :  Lot, Railroad, Utility. | |
| Menambahkan beberapa method pada Player.java seperti givemoney, cekLot dll, getLots dll, cekSekompleks. | |
| Melakukan pengetesan terhadap apa yang dibuat. | |
| Memperbaiki hal-hal yang belum sesuai dengan yang diharapkan. | |
| 2 | Membuat laporan bagian deskripsi class diagram dan men-test program yang telah dibuat. | |

## ***Class Diagram***