

# BOOKLET



KELOMPOK 4

400P

18219031 Elisabeth Gunawan	18219061 Gian Denggan B	18219082 Cihintya Wijaya	18219094 Muhammad Ichsandro D	18219114 Parnaek Routther S
-------------------------------	----------------------------	-----------------------------	----------------------------------	--------------------------------



# Panduan Pengguna (User Manual)

## Deskripsi Game

**HIJI** adalah game yang diadaptasi dari permainan kartu “UNO.” Ketentuan permainan adalah sebagai berikut.

- 1) **HIJI** dimainkan oleh 2-6 pemain.
- 2) Di awal permainan, semua pemain akan mendapatkan 7 buah kartu, dan satu kartu angka dipilih secara acak untuk dijadikan kartu awal.
- 3) Pemain yang akan memulai giliran pertama akan diacak.
- 4) Aturan permainan adalah sebagai berikut.
  - a) Pada setiap giliran, pemain boleh mengeluarkan satu atau lebih kartu yang dapat dimainkan pada giliran tersebut. Ketentuan kartu apa saja yang boleh dikeluarkan ada pada tabel Jenis Kartu.
  - b) Apabila pemain tidak mengeluarkan kartu, pemain wajib mengambil satu kartu dari *deck*.
  - c) Apabila kartu yang baru diambil tersebut bisa dikeluarkan, pemain boleh mengeluarkan kartu tersebut (tidak wajib).
  - d) Apabila kartu tersebut tidak dapat dimainkan, maka giliran diselesaikan tanpa mengeluarkan kartu.
- 5) Beberapa jenis kartu memiliki *power* tertentu yang dapat memengaruhi jalannya permainan (lihat di bagian selanjutnya).
- 6) Apabila pemain memiliki sisa satu kartu, maka pemain harus melakukan “Declare HIJI” dalam waktu 3 detik. Apabila tidak, pemain wajib mengambil dua kartu dari *deck*.
- 7) Pemain dinyatakan menang apabila kartu yang dipegangnya sudah habis, dan permainan selesai.

## Jenis Kartu

Berikut adalah jenis kartu yang dapat dimainkan.

Jenis Kartu	Dapat dikeluarkan jika	Power
Angka 0-9 warna Merah/Hijau/Kuning/Biru	Kartu yang dimainkan sebelumnya berwarna sama, atau memiliki angka yang sama.	-
DrawTwo (+2) warna Merah/Hijau/Kuning/Biru	Kartu yang dimainkan sebelumnya berwarna sama, atau juga kartu DrawTwo.	Pemain selanjutnya harus mengambil 2 kartu. Apabila pemain tersebut mengeluarkan Draw 2, maka pemain selanjutnya mengambil 4, dan seterusnya. Pemain yang mengambil kartu tambahan kehilangan gilirannya dan dilewati.
DrawFour (+4)	Kapanpun	Pemain selanjutnya harus mengambil 4 kartu, dan pemain yang mengeluarkan kartu DrawFour dapat memilih warna yang dapat dimainkan selanjutnya.
Skip warna Merah/Hijau/Kuning/Biru	Kartu yang dimainkan sebelumnya berwarna sama, atau juga kartu Skip.	Pemain selanjutnya kehilangan giliran untuk bermain pada putaran tersebut. Apabila pemain mengeluarkan 2 Skip, maka 2 kehilangan giliran untuk bermain pada putaran tersebut
Wildcard	Kapanpun	Pemain dapat memilih warna yang dapat dikeluarkan oleh pemain selanjutnya.

Reverse warna Merah/Hijau/Kuning/Biru	Kartu yang dimainkan sebelumnya berwarna sama, atau kartu berjenis Reverse.	Arah permainan akan berubah menjadi kebalikan dari arah sebelumnya. Apabila pemain mengeluarkan 2 reverse, maka Arah permainan akan berubah 2 kali.
---------------------------------------	---	---

## Multiple Discard

Apabila suatu pemain memiliki lebih dari satu kartu yang sama persis, pemain tersebut dapat mengeluarkan semua atau sebagian kartu tersebut dalam satu giliran. Apabila kartu tersebut memiliki power, maka:

- 1) Untuk kartu DrawTwo, pemain selanjutnya akan mengambil kartu sebanyak dua kali jumlah kartu yang dikeluarkan.
- 2) Untuk kartu DrawFour, pemain selanjutnya akan mengambil kartu sebanyak empat kali jumlah kartu yang dikeluarkan. Namun, pemain yang mengeluarkan hanya dapat memilih satu warna saja.
- 3) Untuk kartu Skip, jumlah pemain yang dilewati sesuai dengan jumlah kartu Skip yang dikeluarkan.
- 4) Untuk kartu Reverse, urutan akan membolak-balik sesuai jumlah kartu. Misalnya ada dua kartu Reverse, maka urutan pemain tidak berubah.
- 5) Untuk kartu Wildcard, pemain tetap hanya dapat memilih satu warna.

## Menu Program Utama

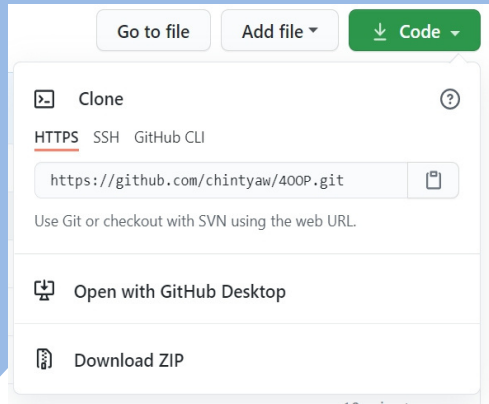
Berikut adalah menu program utama yang tersedia dalam permainan :

Menu	Command	Keterangan
Start Game	Start	Program memulai permainan dan menanyakan jumlah serta nama pemain. Masing-masing pemain akan memegang 7 buah kartu.

List Cards	CardList	Melihat kartu yang ada di tangan,
Discard	Discard	Mengeluarkan/memainkan kartu yang ada di tangan, dan giliran diakhiri.
Draw	Draw	Mengambil satu kartu dari deck.
Declare HIJI	HIJI	<ul style="list-style-type: none"> <li>• Apabila pemain memiliki sisa satu kartu di tangan, pemain harus melakukan Declare HIJI dalam waktu 3 detik. Jika tidak, pemain harus mengambil dua kartu dari deck.</li> <li>• Apabila pemain melakukan Declare HIJI pada saat kartu di tangan lebih dari satu, maka pemain wajib mengambil dua kartu dari deck.</li> </ul>
List Players	PlayerList	Melihat semua nama pemain, jumlah kartu di tangan, dan apakah sedang dalam giliran,
View Player in Turn	PlayerTurn	Melihat pemain yang sedang dalam giliran dan siapa pemain selanjutnya.
Help	Help	Melihat deskripsi aturan permainan.

# Cara Mengakses Game

## 1) Download File Game di repository Github 400P



## 2) Extract file di penyimpanan device kalian

## 3) Compile dan Run program di Command Line Interface atau Programming IDE sesuai petunjuk di file ReadMe.md

### Compile

```
javac Game/Main.java
```

### Run

```
java Game.Main
```



# Alur Bermain (Gameplay)

## Tampilan awal

```
Welcome to HIJI
List of Commands:
- Start
- CardList
- Discard
- Draw
- HIJI
- PlayerList
- PlayerTurn
- Help
Insert command: |
```

Permainan dimulai dengan tampilan awal yang berisi kalimat pembuka dan juga list dari command - command yang bisa mereka gunakan pada permainan. Setelah itu juga terdapat instruksi untuk memasukkan command yang ingin dimasukkan.

## Memulai Permainan

Permainan dimulai ketika *user* memanggil *command start*. Selanjutnya pengguna akan ditanya total pemain sekaligus namanya. Masing-masing pemain akan mendapatkan 7 kartu secara acak sekaligus mengambil satu kartu angka sebagai kartu acuan awal.

```
Insert command: Start
Insert number of players: 3
layer 1 name: Elsa
layer 2 name: Anna
layer 3 name: Olaf
List of Commands:
Start
CardList
Discard
Draw
HIJI
PlayerList
PlayerTurn
Help
Insert command: |
```

## Melihat giliran main

```
Insert command: PlayerTurn
Pemain giliran: Elsa
Pemain selanjutnya: Anna
List of Commands:
- Start
- CardList
- Discard
- Draw
```

Perintah *ViewPlayerTurn* berfungsi untuk melihat giliran pemain saat ini dan pemain yang memiliki giliran selanjutnya.

## Melihat kartu Pemain

Insert command: CardList

1. Skip Hijau
2. Reverse Merah
3. Reverse Biru
4. Draw 2 Kuning
5. 9 Hijau
6. Draw 2 Merah
7. Skip Merah

Perintah *CardList* berfungsi untuk melihat seluruh kartu milik masing-masing pemain. Pada gambar di samping, terlihat pemain memiliki 7 kartu.

## Melihat Jumlah Kartu dan Status Giliran Pemain

perintah *Playerlist* adalah perintah yang digunakan untuk menampilkan nama pemain serta jumlah kartu yang dimiliki.

*PlayerList* juga menampilkan status giliran pemain.

Insert command: PlayerList

Pemain 1: Elsa  
Jumlah Kartu: 7  
Tidak sedang giliran  
Pemain 2: Anna  
Jumlah Kartu: 7  
Sedang giliran

## Mengeluarkan Kartu

Insert command: Discard

Kartu saat ini: 8 Merah  
Kartu yang anda miliki:

1. 4 Kuning
2. 2 Merah
3. Draw 4
4. 1 Merah
5. 2 Hijau

Masukkan kartu yang ingin di-discard: 2

Anda memiliki 1 Kartu 2 Merah

Berapa 2 Merah yang ingin di-discard: 1

Anda telah melakukan discard

Sekarang giliran Anna

Perintah *Discard* berfungsi untuk mengeluarkan kartu yang dimiliki pemain.

Kartu yang akan dapat *discard* ditentukan berdasarkan "Kartu saat ini"

Pemain dapat *discard* lebih dari satu kartu dalam sekali kesempatan bermain namun harus sama persis.

### Note

Pemain yang mendapat giliran pertama akan diacak oleh sistem. Untuk giliran selanjutnya, akan urut sesuai nomor pemain

"Kartu saat ini" di awal permainan juga diacak oleh sistem.



## Draw Kartu

```
Insert command: Draw
Mengambil 7 Biru
Kartu yang anda miliki:
1. 5 Hijau
2. 6 Hijau
3. Reverse Merah
4. 4 Hijau
5. Draw 2 Biru
6. Draw 2 Merah
7. Wildcard
8. 2 Merah
9. 7 Biru
```

Perintah *Draw* berfungsi untuk mengambil satu kartu secara acak dari tumpukan *deck* Kartu.

## Fitur Help

Perintah *Help* berfungsi untuk menyediakan bantuan - bantuan dalam game berupa informasi dari permainan.

Bantuan terdiri dari beberapa petunjuk yang terbagi menjadi 4 kategori. Petunjuk yang ditampilkan disesuaikan dengan pilihan *input user*

```
Insert command: Help

KATEGORI BANTUAN
1. Ketentuan Permainan
2. Command
3. Jenis Kartu yang ada
4. Peraturan Multiple Discard
Masukkan kategori bantuan yang ingin ditampilkan : 1
Hello HIJI Player!
Game HIJI adalah permainan yang diadaptasi dari perma

KETENTUAN PERMAINAN:
1) HIJI dimainkan oleh 2-6 pemain.
```

## Declare HIJI

```
List of Commands:
- Start
- CardList
- Discard
- Draw
- HIJI
- PlayerList
- PlayerTurn
- Help
Insert command: 3 detik telah terlewat dan Anda
belum declare HIJI.
Anda mendapat 2 kartu.
```

Perintah *HIJI* berfungsi untuk mencegah pemain yang kartunya tersisa 1 untuk mendapatkan 2 kartu tambahan. Perintah *HIJI* harus diketik oleh pemain maksimal 3 detik setelah kartunya tersisa 1 buah. Apabila pemain tidak mengetik *HIJI*, maka ia akan mendapatkan 2 kartu tambahan.

### Note

Permainan akan selesai apabila ada satu pemain yang kartunya habis atau Jumlah Kartu = 0



# Pembagian Tugas Kelompok



**18219031**

- Membantu mengerjakan fungsi Help
- Mendesain *booklet*
- Membantu mengisi *booklet*
- Membantu membuat Card class



**18219061**

- Membantu mengerjakan *Game* class
- Membantu mengerjakan *Class* dan *interface Help*
- Membantu mengisi *booklet*



**18219082**

- Mengerjakan class Deck, Player, Card
- Mengerjakan command CardList, PlayerList, Draw, PlayerTurn
- Membantu mengerjakan HIJI, Game, Start, dan Discard



**18219094**

- Mengerjakan class Game, Exception
- Mengerjakan command Discard, Declare Hiji
- Membantu mengerjakan Main



**18219114**

- Membantu mengerjakan *Game* class
- Membantu mengerjakan *Class* dan *interface Help*
- Membantu mengisi *booklet*



# Cerita Proses Pengembangan



## First Meeting

Dilakukan *1st meeting* untuk membahas tubes secara umum

NIM	Nama	Pembagian tugas
18219114	Parmaek Routther Siagian	Game class 2
18219031	Elisabeth Gunawan	Card class
18219094	Muhammad Ihsandro D Noor	Game class 3
18219061	Gian Denngan Benjamin	Game class 1
18219082	Chintya Wijaya	Deck class

Rancangan Awal

Menggunakan command line (for now), apabila memungkinkan akan menggunakan GUI.

Menggunakan class-class: Card, Deck, Game.

Bermain UNO:

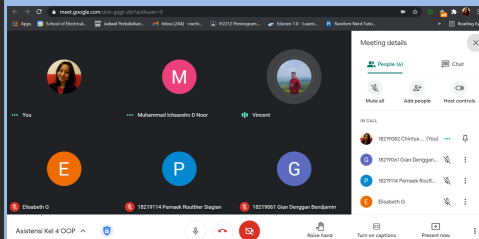
- Masukkan berapa pemain dan nama pemain
- Ada kartu sersisa, seluruh kartu dibagi ke setiap pemain dapat 7 kartu.
- 1 kartu diambil secara random buat di awal
- Ada aksi menentukan urutan reverse/forward, initialnya sorted dari terkecil-terbesar

## Milestone 1

Meeting pertama belum menghasilkan sesuatu yang berarti karena memang belum memiliki pengetahuan mendalam mengenai tubes.

Kami memutuskan untuk mencari referensi mengenai game "UNO".

Sehari setelahnya kami melakukan *2nd meeting* untuk membahas pembagian tugas dan *milestone 1*.



## Asistensi 1

Membahas hal-hal yang masih kurang paham, progress, dan to-do

## Milestone 2

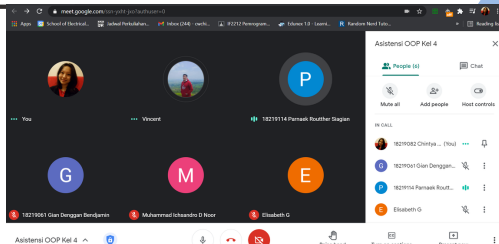
**Milestone 2** kami sudah melakukan implementasi pada *class* namun belum membuat program utama

*Class* yang dibuat adalah Card, Deck, Player, dan Game

Selanjutnya adalah membuat program utama yang menggabungkan *class* yang telah dibuat.

## Asistensi 2

Membahas hal-hal yang masih kurang paham, progress, dan to-do



## Second and Third Meeting

Saat *meeting* bersama yang kedua dan ketiga kalinya kami membahas bersama penyatuan *class* dan pembuatan fungsi-fungsi dalam *main program*.

Kami juga membahas bersama kesulitan-kesulitan yang dihadapi saat mengerjakan tugas yang dikerjakan individual.

# Rancangan Awal

## Rancangan Awal

Menggunakan command line (for now), apabila memungkinkan akan menggunakan GUI.

Menggunakan class-class: Card, Deck, Game.

### Bermain UNO:

- Masukkan berapa pemain dan nama pemain
- Ada kartu semesta, seluruh kartu dibagi ke setiap pemain dapat 7 kartu.
- 1 kartu diambil secara random buat di awal
- Ada state menentukan urutan reverse/tidak. Inisialnya sorted dari terkecil-terbesar (berdasarkan nomor pemain)
- Pemain pertama diacak.
- Masuk ke urutan, tiap pemain bisa mengeluarkan 0 atau lebih kartu. Pilihan (tidak mengeluarkan-ambil dari deck, list kartu yang dipunya). Kalau kartu tidak cocok, keluarin pesan error. Kalau cocok list kartu berkurang dan kartu deck menjadi kartu yang dikeluarkan. Pemain akan diminta untuk terus mengeluarkan kartu hingga memutuskan untuk 'tidak mengeluarkan'.

a) Pada setiap giliran, pemain boleh mengeluarkan satu atau lebih kartu yang dapat dimainkan pada giliran tersebut. Ketentuan kartu apa saja yang boleh dikeluarkan ada pada tabel Jenis Kartu.

b) Apabila pemain tidak mengeluarkan kartu, pemain wajib mengambil satu kartu dari deck.

c) Apabila kartu yang baru diambil tersebut bisa dikeluarkan, pemain boleh mengeluarkan kartu tersebut (tidak wajib).

d) Apabila kartu tersebut tidak dapat dimainkan, maka giliran diselesaikan tanpa mengeluarkan kartu

Kartunya abstract, lalu didefinisikan kartunya apa

Inheritance: Card (kartu biasa (0-9), kartu lain(skip, reverse, dll))

### Class:

1. Card  
Atribut: value, color  
Method: constructor, getter (getcolor, getvalue), setter (setcolor, setvalue)
2. Deck  
Atribut: list of card  
Method: reset (buat inisiasi kartu yang ada di deck apa-apa aja +randomin), draw (ambil n kartu teratas, dibutuhkan saat +2 dan +4), shuffle(?)
3. Game  
Atribut: idplayer, direction,  
Method: start (keluarin kartu pertama: valid kalau bukan kartu skip, +2, +4, wild), gettop (ambil kartu paling atas), getcurrentplayer (pemain saat ini), getnextplayer (pemain selanjutnya), getplayers (pemain ada siapa aja), getplayerhand (menampilkan kartu yang dimiliki oleh pemain), discard, declare, isgameover
4. Main

### Exception

- Ada beberapa exception -> inheritance
- List beberapa exception

### Pembagian

1. Card (1) + help
2. Deck (1):
3. Game (3): 3.1(start, gettop, getplayerhand), 3.2(getcurrentplayer, getnextplayer, getplayers), 3.3(discard, declare, isgameover)
4. Exception
5. Main

# Perubahan dari Rancangan Awal

## 1) Menambahkan *class Player*

Alasan : Penumpukan fungsi yang terlalu banyak di *class Game* menjadi alasan terutama ide dibalik penambahan *class Player*. Selain itu, *Player* merupakan sebuah objek yang memiliki method tersendiri untuk setiap pemain sehingga lebih baik untuk dijadikan *class*. *Class Player* sendiri memiliki fungsi-fungsi sebagai berikut *getName*, *getCard*, *addCard*, *removeCard*, *getJumlah*, *ListCards*, dan *ShuffleSeven*. Fungsi ini awalnya diproyeksi diinisialisasi di *class Game*, namun setelah pertimbangan di atas, akhirnya dipindahkan ke *class Player*.

## 2) Menambah *Class Help* dan penggunaan *Interface*

Alasan: Bantuan pada permainan terlalu banyak untuk sekaligus ditampilkan dalam sekali pemanggilan. Oleh karena itu kami menggunakan *class khusus* bernama *Help* serta penggunaan *interface* untuk menyiasatinya. Pada *interface*, akan diinisialisasikan fungsi sesuai jenis bantuan yang dipilih. Selanjutnya, pemain dapat mengakses informasi tersebut dengan mengetikkan angka sesuai dengan informasi yang ingin mereka akses.

## 3) Awalnya kartu ingin dibuat abstrak, pada implementasinya *class Card* menerapkan *Inheritance* dengan *Card* sebagai *superclass* dan jenis-jenis kartu sebagai *subclass*

Alasan: Penerapan *inheritance* dengan membuat jenis-jenis kartu sebagai *subclass* membuat atribut dan *method* dari masing-masing jenis kartu bisa terkelompokkan. Selain itu, ketika membuat objek kartu, tidak perlu mendefinisikan ulang atribut dan *method*-nya.

## 4) Atribut *class Card* memiliki *number* dan *color* pada beberapa *child*

Alasan: Atribut *number* dimiliki semua jenis termasuk *non number* yang bernilai -1, sedangkan *color* dimiliki oleh semua jenis kecuali *Wildcard* dan *Draw Four*.

## 5) Atribut *class Game* menjadi *nama*, *players*, *jumlahTwo*, *skipcount*, *warnaKartu*, *angkaKartu*, *deck*, *numplayers*, *giliran*, *clockwise*.

Alasan: Atribut-atribut tersebut dibutuhkan untuk mempermudah akses variabel-variabel yang dibutuhkan untuk *implementasi* fungsi.



KELOMPOK 4

400P