

### 3. DESAIN SISTEM

Bab ini berisi desain dalam implementasi yang meliputi *flowchart* dan rancangan desain *User Interface*.

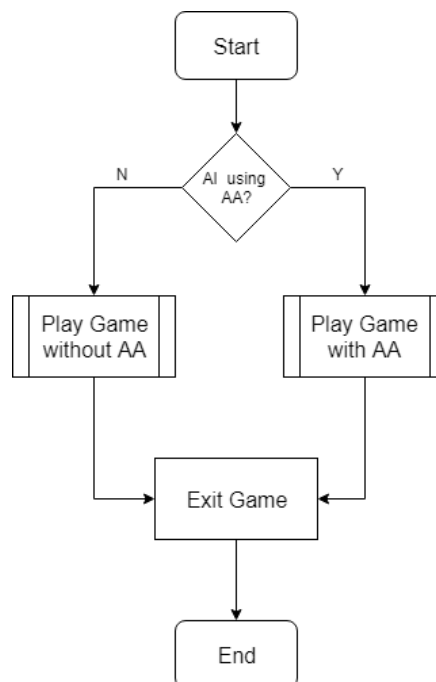
#### 3.1 Diagram *Flowchart*

Sub bab ini berisi diagram *Flowchart* tentang setiap proses yang berada dalam program yang dibuat.

##### 3.1.1 *Flowchart* Menu Utama

Bagian ini berisi *flowchart* yang menjelaskan fitur-fitur menu utama di dalam program seperti pada Gambar 3.1. Pada menu utama terdapat 3 pilihan yaitu *Play game without AA*, *Play game with AA*, dan *Exit Game*. Secara singkat penjelasan tiap opsi atau pilihan dapat dijabarkan sebagai berikut:

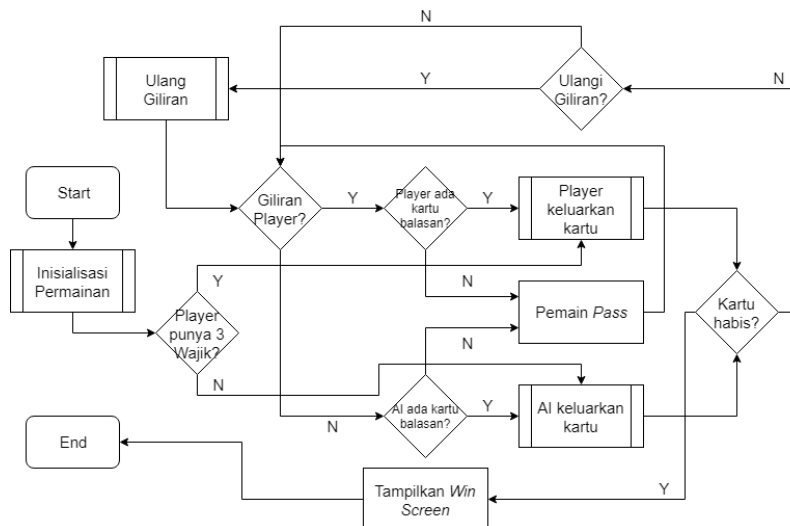
- *Play game without AA* akan memulai permainan capsa dengan pengaturan AI memakai algoritma biasa.
- *Play game with AA* akan memulai permainan capsa dengan pengaturan AI memakai *Alternative Algorithm*.
- *Exit Game* akan mengeluarkan pemain dari permainan



Gambar 3.1 *Flowchart* Menu Utama

### 3.1.2. Flowchart Play Game

Bagian ini berisi *flowchart* yang menjelaskan cara kerja permainan cepsa dimana pemain akan melawan 3 musuh dengan algoritma biasa maupun dengan *Alternative Algorithm* seperti pada Gambar 3.2. Pada awal permainan akan melakukan inisialisasi beberapa aset didalam permainan, membagikan kartu ke 4 pemain yang terdiri dari *player* dan 3 AI. Masing-masing pemain akan menerima 13 kartu yang sudah diacak pembagiannya. Bila pemain (*player*) menerima kartu 3 ♦ maka akan memulai permainan terlebih dahulu. Bila tidak maka musuh AI yang memiliki akan jalan terlebih dahulu. Kemudian apabila giliran *player* / musuh selesai maka akan dilanjutkan dengan giliran selanjutnya. Apabila pemain selanjutnya memiliki/ mau mengeluarkan *combo* kartu balasan maka dapat mengeluarkan *combo* tersebut. Bila tidak maka dapat melakukan *pass* dan berlanjut ke giliran pemain selanjutnya. Pada saat ini pemain dapat memilih untuk mengulangi giliran. Apabila salah satu pemain telah menghabiskan kartunya terlebih dahulu maka akan memunculkan tampilan *Win Screen* dan tulisan salah satu pemain yang berhasil memenangkan permainan.

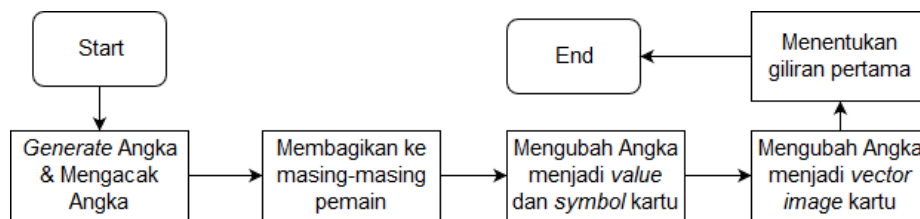


Gambar 3.2 Flowchart Play Game

### 3.1.3 Flowchart Inisialisasi Permainan

Bagian ini berisi *flowchart* yang menjelaskan cara kerja sistem dalam melakukan inisialisasi aset-aset yang diperlukan dalam permainan seperti pada Gambar 3.3. Awalnya sistem me-generate angka mulai dari 1 hingga 52 dan

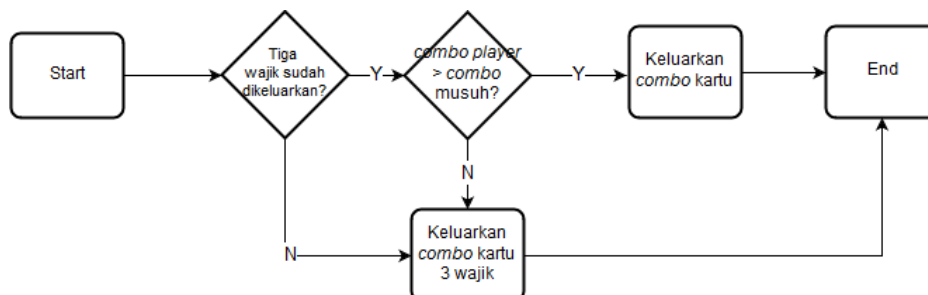
mengacak urutannya sehingga tidak ada pemain yang memiliki angka yang sama. Pengacakan yang dilakukan berbeda cara dengan permainan aslinya sehingga kartu yang didapatkan pemain lebih acak setiap permainan dimulai. Kemudian sistem membagikan 52 angka tersebut ke masing-masing pemain secara bergantian. Lalu sistem mengubah angka pada masing-masing pemain menjadi *value*, *symbol*, dan *vector image* kartu yang akan digunakan selama permainan berlangsung. Terakhir adalah menentukan giliran pemain yang akan memulai permainan dengan cara mencari pemain yang memegang kartu 3 ♦.



Gambar 3.3 *Flowchart* Inisialisasi Permainan

### 3.1.4 *Flowchart* Player keluarkan kartu

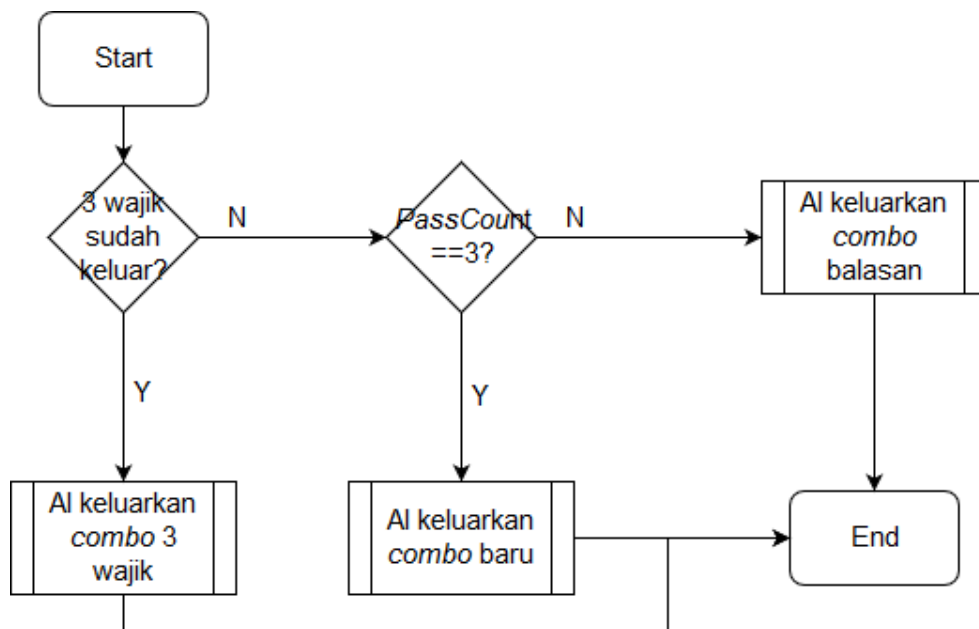
Bagian ini berisi *flowchart* yang menjelaskan cara kerja pemain (*player*) saat giliran dimulai seperti pada Gambar 3.4. Bila kartu 3 ♦ belum dikeluarkan maka pemain harus mengeluarkan *combo* kartu yang mengandung kartu 3 ♦ untuk memulai permainan. Bila tidak maka pemain dapat mengeluarkan kartu *combo* yang lebih besar daripada musuh. Bila tidak bisa/ mau maka pemain akan melakukan *pass* dan berlanjut ke pemain selanjutnya. Tidak ada ketentuan seberapa besar kartu *combo* balasan pemain ke kartu pemain sebelumnya sehingga pemain dapat mengeluarkan kartu *combo* yang menurutnya dapat menjadikan dirinya lebih unggul dari pemain yang lain.



Gambar 3.4 *Flowchart* Player keluarkan kartu

### 3.1.5 Flowchart AI keluarkan kartu

Bagian ini berisi *flowchart* yang menjelaskan cara kerja pemain (AI) saat giliran dimulai seperti pada Gambar 3.5. Apabila pemain sebelumnya belum mengeluarkan *combo* kartu 3 ♦ maka AI akan mengeluarkan *combo* 3 ♦ untuk memulai permainan, tetapi apabila sudah maka AI akan memilih untuk mengeluarkan *combo* kartu balasan atau baru tergantung pada nilai variabel *PassCount* yang merupakan jumlah *pass* yang telah dilakukan masing-masing pemain. Apabila *PassCount* mencapai 3 maka AI akan mengeluarkan *combo* kartu baru untuk memulai giliran baru. Bila tidak maka AI akan mengeluarkan *combo* kartu balasan dari pemain sebelumnya.

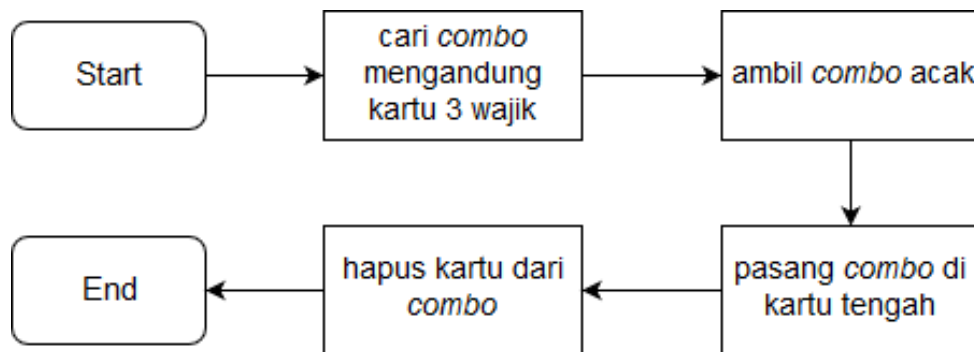


Gambar 3.5 Flowchart AI keluarkan kartu

### 3.1.6 Flowchart AI keluarkan 3 wajik algoritma biasa

Bagian ini berisi *flowchart* yang menjelaskan cara kerja AI untuk mengeluarkan *combo* kartu 3 ♦ dengan algoritma biasa seperti pada Gambar 3.6. Awalnya AI akan mencari *combo* kartu yang mengandung kartu 3 ♦ dari *list combo* AI. Kemudian AI akan mengambil *combo* acak untuk dikeluarkan. Kemudian AI memasang nilai dan simbol kartu di kartu tengah dan menghapus

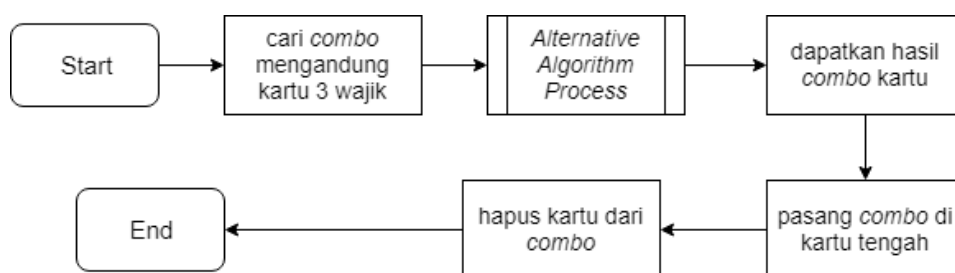
kartu yang dipakai dalam *combo* tersebut dan juga *combo* yang mengandung kartu-kartu yang dikeluarkan dari *list combo*.



Gambar 3.6 *Flowchart* AI keluarkan 3 wajik Algoritma biasa

### 3.1.7 *Flowchart* AI keluarkan 3 wajik *Alternative Algorithm*

Bagian ini berisi *flowchart* yang menjelaskan cara kerja AI untuk mengeluarkan *combo* kartu 3 ♦ dengan algoritma biasa seperti pada Gambar 3.7. Awalnya AI akan mencari *combo* kartu yang mengandung kartu 3 ♦ dari *list combo* AI. Kemudian AI melakukan proses kalkulasi dalam *Alternative Algorithm* untuk menentukan *combo* kartu yang akan dikeluarkan. Setelah mendapatkan hasil kemudian AI memasang nilai dan simbol kartu di kartu tengah dan menghapus kartu yang dipakai dalam *combo* tersebut dan juga *combo* yang mengandung kartu-kartu yang dikeluarkan dari *list combo*.

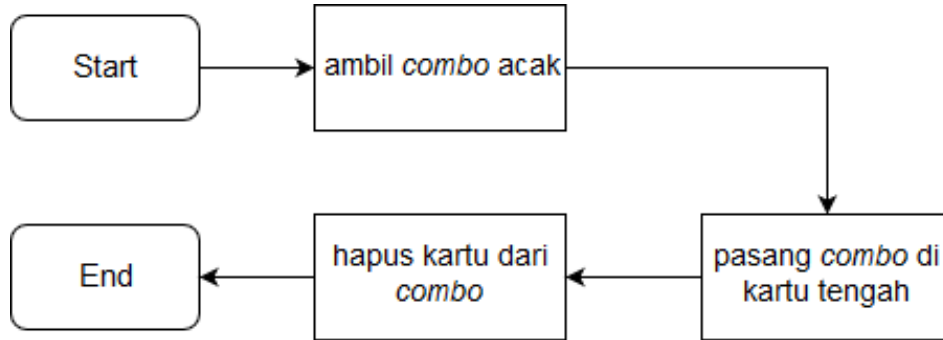


Gambar 3.7 *Flowchart* AI keluarkan 3 wajik *Alternative Algorithm*

### 3.1.8 *Flowchart* AI keluarkan *combo* baru algoritma biasa

Bagian ini berisi *flowchart* yang menjelaskan cara kerja AI untuk mengeluarkan *combo* kartu baru dengan algoritma biasa seperti pada Gambar 3.8.

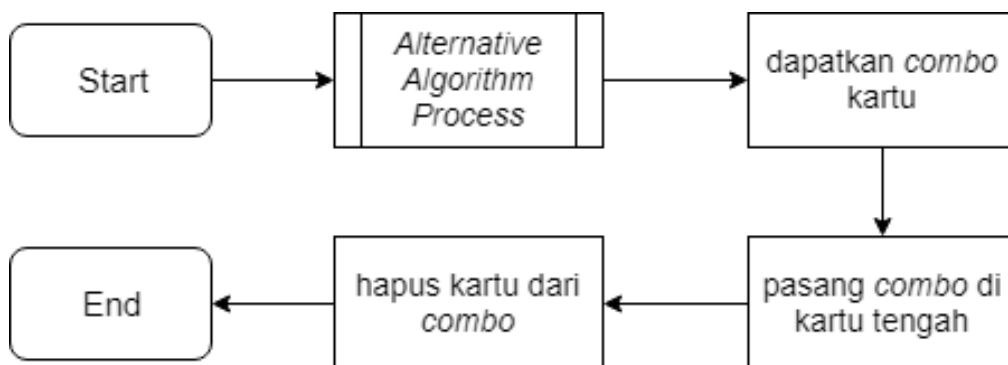
Awalnya AI mengambil *combo* acak untuk dikeluarkan. Kemudian AI memasang nilai dan simbol kartu di kartu tengah dan menghapus kartu yang dipakai dalam *combo* tersebut dan juga *combo* yang mengandung kartu-kartu yang dikeluarkan dari *list combo*.



Gambar 3.8. *Flowchart* AI keluarkan *combo* baru Algoritma biasa

### 3.1.9 *Flowchart* AI keluarkan *combo* baru *Alternative Algorithm*

Bagian ini berisi *flowchart* yang menjelaskan cara kerja AI untuk mengeluarkan *combo* kartu baru dengan algoritma biasa seperti pada Gambar 3.9. Awalnya AI akan memulai proses kalkulasi *Alternative Algorithm* untuk menentukan *combo* kartu untuk dikeluarkan. Kemudian AI memasang nilai dan simbol kartu di kartu tengah dan menghapus kartu yang dipakai dalam *combo* tersebut dan juga *combo* yang mengandung kartu-kartu yang dikeluarkan dari *list combo*.



Gambar 3.9 *Flowchart* AI keluarkan *combo* baru *Alternative Algorithm*

### 3.1.10 *Flowchart* AI keluarkan *combo* balasan algoritma biasa

Bagian ini berisi *flowchart* yang menjelaskan cara kerja AI untuk mengeluarkan *combo* kartu balasan dengan algoritma biasa seperti pada Gambar 3.10. Awalnya AI mencari *combo* kartu yang nilainya lebih besar dari *combo* kartu sebelumnya dan menyimpannya kedalam *list* baru. Lalu AI mengambil *combo* dengan indeks pertama dalam *list* baru yang telah dibuat sebelumnya. Kemudian AI memasang nilai dan simbol kartu di kartu tengah dan menghapus kartu yang dipakai dalam *combo* tersebut dan juga *combo* yang mengandung kartu-kartu yang dikeluarkan dari *list combo*.



Gambar 3.10. *Flowchart* AI keluarkan *combo* balasan Algoritma biasa

### 3.1.11 *Flowchart* AI keluarkan *combo* balasan *Alternative Algorithm*

Bagian ini berisi *flowchart* yang menjelaskan cara kerja AI untuk mengeluarkan *combo* kartu balasan dengan *Alternative Algorithm* seperti pada Gambar 3.11. Awalnya AI mencari *combo* kartu yang nilainya lebih besar dari *combo* kartu sebelumnya dan menyimpannya kedalam *list* baru. Lalu AI melakukan proses kalkulasi *Alternative Algorithm* untuk menentukan *combo* kartu yang akan dikeluarkan berdasarkan dari *list* baru yang telah dibuat sebelumnya. Setelah mendapatkan hasil *combo* kartu terbaik kemudian AI memasang nilai dan simbol kartu di kartu tengah dan menghapus kartu yang dipakai dalam *combo* tersebut dan juga *combo* yang mengandung kartu-kartu yang dikeluarkan dari *list combo*.

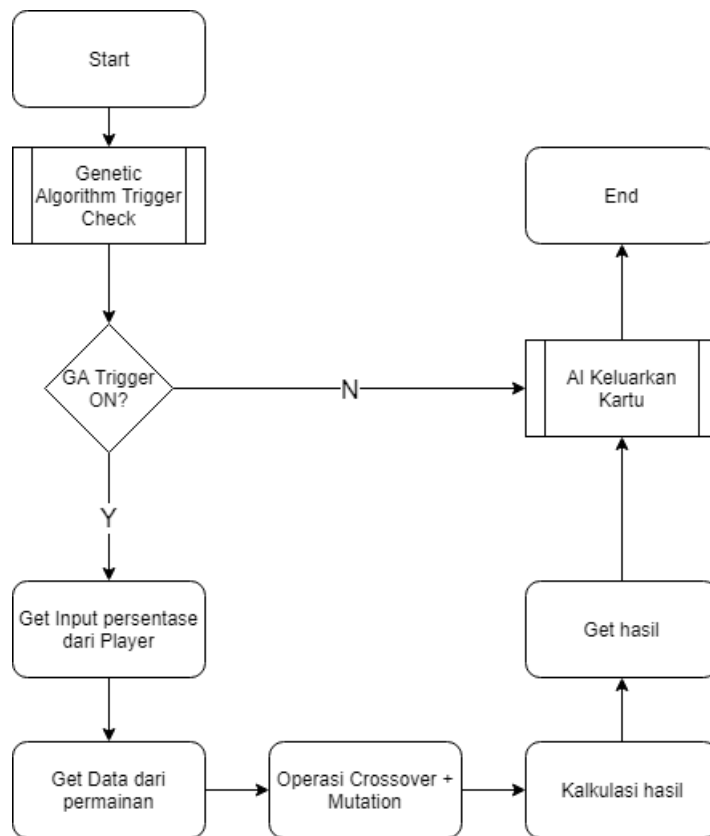


Gambar 3.11 *Flowchart* AI keluaran *combo* balasan *Alternative Algorithm*

### 3.1.12 *Flowchart Alternative Algorithm*

Bagian ini berisi *flowchart* yang menjelaskan cara kerja *Alternative Algorithm* saat AI melakukan berbagai fungsi untuk mengeluarkan *combo* kartu seperti mengeluarkan *combo* kartu 3 ♦, *combo* kartu baru, dan *combo* kartu balasan dengan cara seperti pada Gambar 3.12. Awalnya AI akan melakukan inialisasi untuk parameter yang terdapat dalam *Alternative Algorithm* seperti kromosom-kromosom dari *list combo* yang akan dipakai, fungsi *random* untuk mengambil *combo* kartu acak dari *list*, *fitness function*, dan *elitism*. Selanjutnya algoritma akan membandingkan nilai *fitness* dari *combo* kartu sekarang dan *combo* kartu yang sebelumnya, apabila lebih baik maka akan disimpan untuk iterasi selanjutnya. Setelah iterasi pemilihan *combo* kartu selesai maka dilakukan operasi *crossover* dan *mutation* untuk menghasilkan anak dari kromosom induk yang telah terseleksi. Hasil kromosom terbaik akan disimpan untuk dipanggil pada akhir iterasi. Setelah semua iterasi selesai maka algoritma akan mengembalikan hasil berupa nama *combo* kartu akan dikeluarkan oleh AI.

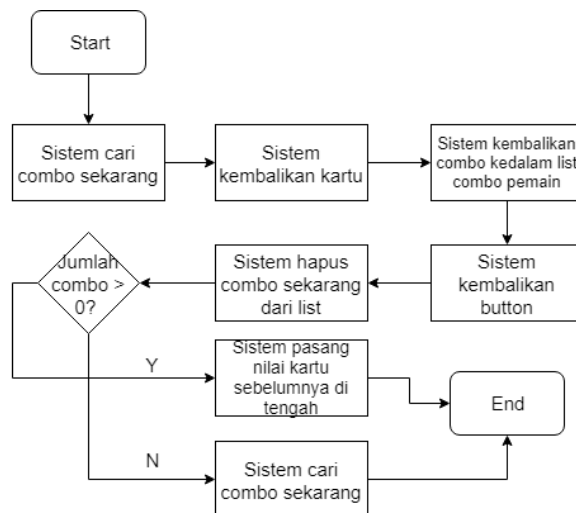
Kromosom-kromosom yang terdapat dalam *list combo* telah ditentukan sesuai dengan berbagai fungsi sebelum proses *Alternative Algorithm* dipanggil oleh sistem. Contohnya apabila fungsi yang dilakukan sebelumnya adalah menentukan *combo* kartu balasan maka isi *list combo* yang akan digunakan dalam proses *Alternative Algorithm* akan berisi kumpulan *combo* dengan jumlah yang sama dengan *combo* sebelumnya dan memiliki nilai lebih besar.



Gambar 3.12 *Flowchart Alternative Algorithm Process*

### 3.1.13 *Flowchart Ulangi Giliran*

Bagian ini berisi *flowchart* yang menjelaskan cara kerja sistem untuk kembali pada giliran sebelumnya seperti pada Gambar 3.13. Awalnya sistem mencari *combo* kartu sekarang dari *list* penyimpanan, kemudian sistem mengembalikan kartu yang dipakai dalam *combo* sekarang. Dilanjutkan dengan mengembalikan semua kombinasi *combo* yang memiliki nilai kartu dalam *combo* sekarang kedalam *list combo* pemain dan mengatur *button* yang akan dipakai oleh pemain sebelumnya. Sistem kemudian menghapus semua kombinasi *combo* yang memiliki nilai kartu dalam *combo* sekarang dari *list* penyimpanan dan mengecek apabila masih ada *combo* tersisa didalam *list* penyimpanan. Bila ada maka akan diambil *combo* baru sebelum *combo* sekarang dan dipasang pada kartu tengah. Bila tidak maka kartu tengah dibiarkan dalam kondisi tertutup.



Gambar 3.13 *Flowchart* Ulangi Giliran

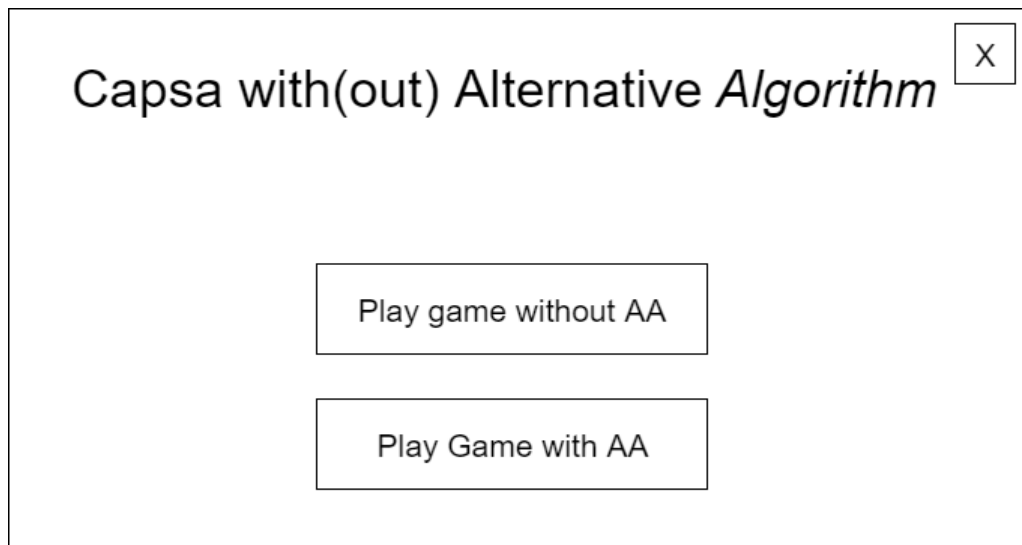
### 3.2 Desain *Interface* Permainan

Sub bab ini berisi tentang desain yang memaparkan rancangan tampilan dari setiap menu yang ada di dalam program setelah pemain masuk dan menjalankan program. Berbagai desain yang dibuat dalam program ini terinspirasi oleh berbagai desain permainan yang serupa seperti permainan kartu yang berbasis aplikasi dengan berbagai *game engine* dan bahasa pemrograman.

#### 3.2.1 Desain *Interface* Menu Utama

Bagian ini berisi desain yang menjelaskan tampilan setelah pemain masuk ke permainan seperti pada Gambar 3.13. Warna latar tampilan dominan hitam dengan tulisan dan tombol yang berwarna terang dan beberapa gambar kartu remi dan simbolnya agar mata tidak perih saat melihat tampilan.

Dalam tampilan menu utama akan ada 3 tombol yang terdiri dari *Play game without AA*, *Play game with AA* dan *exit*. Tombol *exit* disimbolkan dengan gambar palang (X) digunakan untuk keluar dari permainan. Tombol *Play game without AA* digunakan untuk memulai permainan tanpa *Alternative Algorithm*, sedangkan tombol *Play game with AA* digunakan untuk memulai permainan dengan *Alternative Algorithm*.



Gambar 3.13 Desain *Interface* Menu Utama

### 3.2.2 Desain *Interface Start Game*

Bagian ini berisi desain yang menjelaskan tampilan setelah pemain memilih salah satu tombol pada tampilan awal permainan seperti pada Gambar 3.14. Penentuan masing-masing pemain dimulai dari *player* yang berada di paling bawah layar dilanjutkan oleh musuh 1 yang berada di samping kiri *player*, musuh 2 yang berada di depan *player*, dan musuh 3 yang berada di samping kanan *player*.

Terdapat 6 tombol yang terdiri dari *Pass*, *Out*, *3 Wajik*, *Next Turn*, *Exit Game*, dan *Retry*. Tombol *Pass* digunakan untuk mengakhiri giliran tanpa mengeluarkan kartu pemain bila pemain tidak memiliki *combo* kartu yang lebih besar atau memilih untuk tidak mengeluarkan kartu balasan. Tombol *Out* digunakan untuk membalas kartu *combo* dari giliran musuh sebelum pemain. Kartu pemain (*player*) dapat di tekan atau *klik* untuk melakukan *selection* pada kartu yang ingin dikeluarkan. Kartu yang telah diseleksi oleh *player* akan ditandai dengan majunya kartu dibanding kartu yang tidak diseleksi oleh *player*. Tombol *3 Wajik* hanya dapat sekali digunakan bila salah satu musuh yang memiliki kartu tiga wajik. Tombol ini berguna untuk mengeluarkan *combo* kartu tiga wajik yang dimiliki oleh musuh sekaligus untuk memulai permainan. Terakhir tombol *Next*

*Turn* digunakan untuk memulai giliran musuh selanjutnya. Tombol *Exit Game* yang diberi tanda X berguna untuk kembali ke menu utama dari permainan sehingga pemain dapat memilih untuk melanjutkan permainan dengan metode biasa atau metode alternatif. Terakhir tombol *Retry* digunakan untuk kembali ke giliran pemain sebelumnya.

Terdapat 13 kartu yang dimiliki oleh masing-masing pemain dengan angka dan gambar yang sudah diacak sebelum permainan dimulai tetapi hanya kartu pemain (*player*) yang menampilkan angka dan gambar tersebut. Seluruh kartu musuh hanya akan menampilkan bagian bawah kartu saja. Terdapat juga 5 kartu ditengah-tengah pemain untuk menunjukkan *combo* yang sedang dimainkan sekarang. Awalnya 5 kartu ditengah tersebut akan menampilkan bagian bawah kartu tetapi setelah salah satu pemain mengeluarkan *combo* kartu maka 5 atau sebagian dari kartu tengah tersebut akan berganti tergantung pada besar *combo* kartu yang dimainkan (misal musuh mengeluarkan *combo pair\_3* maka 2 kartu tengah akan berubah sesuai dengan *combo* kartu yang dipilih oleh musuh, sedangkan 3 kartu tengah sebelahnya akan tetap memakai tampilan kartu bagian bawah).

Warna papan adalah hijau agar menyesuaikan dengan tema permainan kartu pada umumnya. Sedangkan warna keempat tombol dibuat berbeda seperti tombol *pass* dengan warna merah, tombol *out* dengan warna biru muda, tombol *3 wajik* dengan warna hijau muda, dan tombol *next turn* dengan warna kuning agar memudahkan *player* dalam mengenali tiap tombol dalam permainan. Tombol *Exit Game* yang ditandai dengan X diberi warna biru muda agar terlihat mencolok pada permainan. Terakhir tombol *Retry* diberi warna orange agar pemain lebih terdugah untuk mencoba tombol fitur dari tombol tersebut.

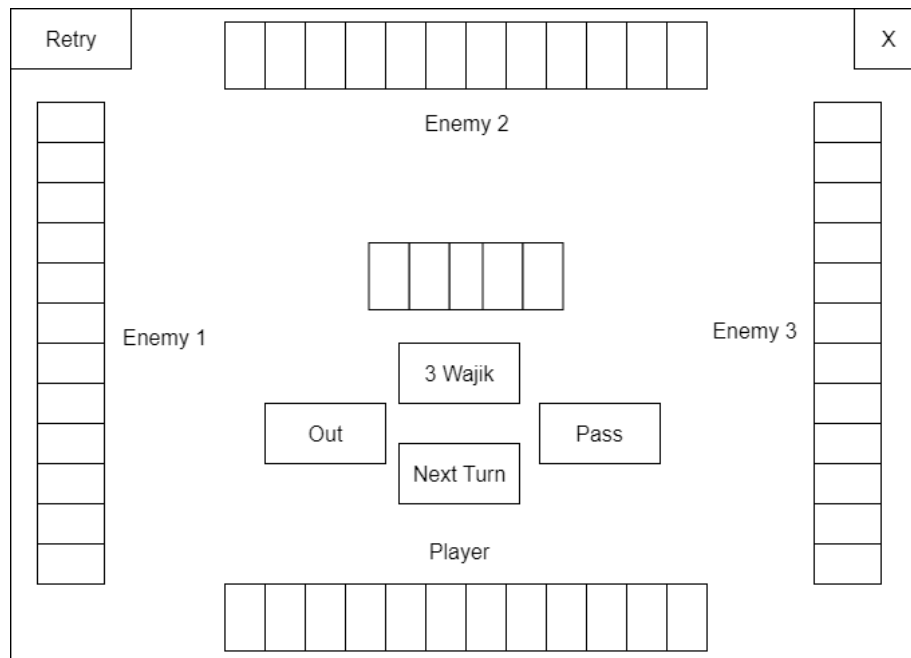
Untuk kapan setiap tombol dapat ditekan juga telah diatur agar dapat mengikuti dengan jalurnya permainan. Untuk tombol *3 Wajik* hanya akan menyala apabila pemain tidak mendapatkan kartu 3 ♦. Sedangkan untuk tombol *Out* dan *Pass* hanya akan menyala apabila giliran *player* telah dimulai. Selanjutnya untuk tombol *Next Turn* hanya akan menyala saat giliran musuh dimulai.

Contohnya apabila sekarang adalah giliran musuh 2 dan kartu 3 ♦ telah dikeluarkan maka yang dapat ditekan oleh *player* hanya tombol *Next Turn*

sedangkan tombol lainnya seperti *Out*, *Pass*, dan *3 Wajik* tidak dapat ditekan oleh *player*.

Untuk tombol *Exit Game* yang ditandai dengan X pada kanan atas layar dapat selalu dipilih apabila pemain ingin mengubah mode dari permainan tersebut. Terakhir tombol *Retry* yang terletak pada kiri atas layar hanya dapat dipilih ketika pemain yang memiliki *combo* kartu 3 ♦ telah mengeluarkan *combo* tersebut. Selama ada *combo* kartu yang telah dikeluarkan maka tombol *Retry* dapat terus digunakan.

Contohnya apabila *list* penyimpanan sekarang memiliki *combo* 3 *diamonds*, 5 *hearts*, 9 *spades*, dan *q\_diamonds* maka saat tombol *Retry* dipilih pemain maka sistem akan mengembalikan kartu *q\_diamonds* kepada pemain yang memilikinya dan memasang 9 *spades* pada kartu tengah. Kemudian menghapus *combo* yang memiliki *q\_diamonds* pada *list* sementara. Tetapi apabila *list* penyimpanan sekarang hanya tersisa 3 *diamonds* maka sistem akan menghapus *combo* 3 *diamonds* dari *list* penyimpanan dan mengubah kartu tengah menjadi kartu tertutup sehingga menandakan *combo* kartu 3 *diamonds* belum dikeluarkan.



Gambar 3.14 Desain *Interface* saat mulai permainan

### 3.2.3 Desain *Interface Win Screen*

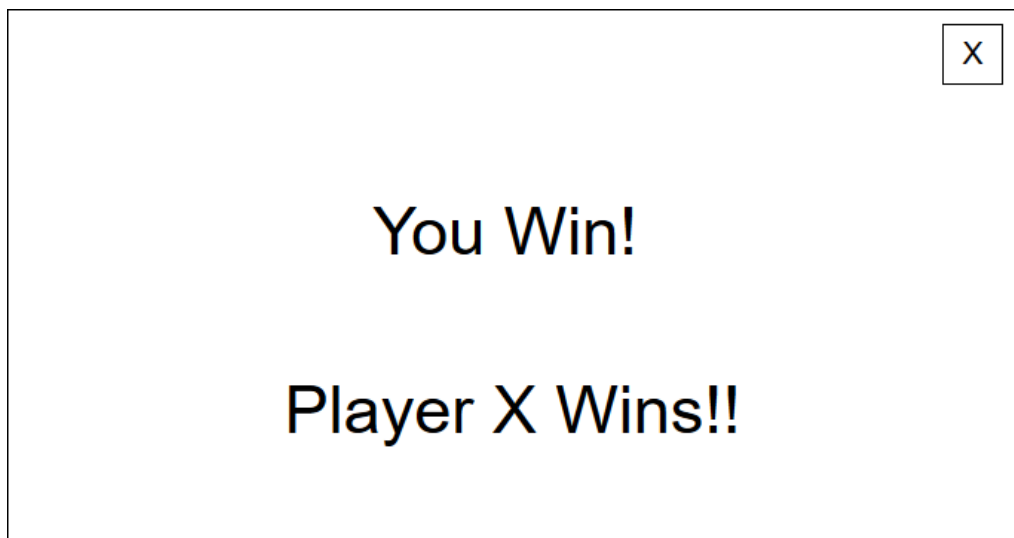
Bagian ini berisi desain yang menjelaskan tampilan setelah salah satu pemain berhasil menghabiskan kartu terlebih dahulu seperti pada Gambar 3.15.

Hanya ada 1 tombol yaitu *Exit Game* yang dilambangkan dengan tombol X pada kanan atas layar untuk keluar dari permainan. Pada bagian tengah layar terdapat tulisan besar *You Win!* dan tulisan lebih kecil *Player X wins!!* Kata X pada tulisan *Player X wins!!* akan digantikan oleh angka pemain pertama yang telah menghabiskan kartu.

Contohnya apabila pemain 3 (musuh ke-2) telah menghabiskan semua kartunya terlebih dahulu maka pada tampilan *winScreen* kata X akan digantikan oleh 3 sehingga menjadi *Player 3 Wins!!* Yang menunjukkan bahwa pemain 3 telah berhasil memenangkan permainan.

Semua tulisan berwarna putih hijau dan putih dan warna latar hitam agar tidak membuat mata pemain lelah dengan warna cerah pada tema warna tampilan sebelumnya.

Setelah pemain menekan tombol X pada kanan atas permainan maka permainan akan berakhir dan pemain dapat memulai kembali permainan dengan memilih *file* permainan yang berekstensi *.exe* pada *folder* permainan.



Gambar 3.15 Desain *Interface Win Screen*