Aplikasi Permainan Capsah Banting dengan Penerapan Algoritma *Minimax*

Hapnes Toba, Billy Kurniawan

Program Studi D3 Teknik Informatika
Fakultas Teknologi Informasi, Universitas Kristen Maranatha
. Prof. Drg. Suria Sumantri No. 65 Bandung 40164
Email: hapnes.toba@eng.maranatha.edu, lord_mephisto_82@yahoo.com

Abstract

Big-two card game, is one of well-known card climbing games which probably originated in coastal China around 1980. Soon it became very popular in Shanghai, Hong Kong, Taiwan and also in the Philippines and Singapore; it has also spread to some western countries. In climbing games, each player in turn must play a higher card (or combination of cards) than the previous player. A player who cannot or does not wish to beat the previous play can pass. This continues for as many rounds as necessary until someone makes a play which no one will beat. That player wins the "trick" and leads to another one. Because players can pass, their cards are not used at an equal rate and some will run out before others. This kind of strategy produces an imperfect information environment. In this article will be reported how the Big Two card game is implemented in computer with the application of Minimax Algorithm, to predict the information, i.e. the value of cards in hand, for the next player who has the turn to minimize the card "value" of the other players.

Keywords : Big Two, Card Games, MiniMax Algortihm, Macromedia Flash

1. Pendahuluan

Perkembangan teknologi saat ini kian berkembang pesat, komputerisasi pun merambah ke segala bidang. *Game* (aplikasi permainan) adalah sebagian kecil dari teknologi masa kini yang terus mengalami perkembangan. Banyak permainan klasik yang dibuat dalam versi komputer, contohnya permainan kartu dan kasino (*casino*). Komputer yang saat ini sudah menjadi barang kebutuhan pokok bagi masyarakat modern, menyebabkan banyaknya permainan kartu dan kasino yang diciptakan dalam versi komputer. Sebut saja *Hearts, Free Cell, Solitaire, Poker* dan *Black Jack*, *game-game* tersebut sudah cukup terkenal di masyarakat dan tersebar di berbagai sumber.

Capsah Banting adalah salah satu permainan kartu yang cukup terkenal di kalangan remaja maupun orang dewasa di Indonesia. Permainan kartu ini diciptakan di daratan Cina sekitar tahun 1980, dan menjadi sangat terkenal di Shanghai, Hongkong, Taiwan, Filipina dan Singapura dan juga menyebar ke daerah

barat. Permainan Capsah ini mempunyai beberapa nama, di China permainan ini dikenal dengan nama *Choi Dai Di* atau *Bu Bu Gao Sheng*. Di Amerika dan Australia dikenal dengan nama *Big Two, Big Deuce,* atau *Deuce.* Meskipun permainan kartu ini cukup terkenal, masih jarang ditemukan versi komputernya.

Keunggulan membuat *game* kartu di komputer adalah adanya tampilan yang atraktif, dan tidak membutuhkan orang lain sebagai musuh karena dapat diprogramkan menggunakan kecerdasan buatan yang dapat berperan sebagai musuh. Algoritma yang diterapkan pada aplikasi ini adalah *Minimax*.. Algoritma *Minimax* digunakan atas dasar pertimbangan bahwa algoritma ini terbukti cukup *hand*al dan telah diterapkan dengan baik untuk game catur dan *tic-tac-toe*.

2. Metodologi Pengembangan Aplikasi

2.1. Metode Analisis Sistem

Aplikasi ini dimodelkan dengan menggunakan *Unified Modelling Language* (UML), dengan kelangkapan diagram:

a) Diagram Aktivitas
Menggambarkan aliran kerja (*workflow*) dan operasional dari suatu sistem**.**

b) Diagram Kelas
Menggambarkan keterkaitan antar kelas dan struktur kelas yang digunakan dalam aplikasi.

2.2.Metode Evaluasi

a) Fungsional Test
Fungsional test bertujuan untuk mengevaluasi setiap fungsionalitas Aplikasi Permainan Capsah Banting. Dengan Fungsional Test ini, akan dapat diketahui apakah fungsi dari sebuah tombol sudah berjalan dengan baik, apakah informasi teks akan muncul dengan baik, dan apakah perpindahan modul berjalan seperti yang diharapkan.

b) Black Box
Pada tahap pengujian *Black box* akan dilakukan testing dengan cara menjalankan aplikasi Capsah Banting ini secara bertahap, selayaknya orang bermain kartu.

2.2.*Software* Yang Digunakan

Software yang digunakan dalam pembuatan aplikasi ini adalah :

a) Macromedia Flash 8
Macromedia Flash 8 dengan bahasa pemrograman *action script* 2.0 digunakan untuk pembuatan aplikasi permainan Capsah Banting ini. Pemilihan *software* ini dilandasi pertimbangan bahwa dengan Macromedia Flash tampilan akan lebih menarik dan memiliki kemampuan orientasi obyek. [Underdahl, 2002]

b) Corel draw X3

Digunakan untuk membuat objek gambar dan *background* yang akan digunakan dalam aplikasi.

c) **Photoshop CS 2**
Digunakan sebagai pengolah gambar yang telah dibuat dibuat dalam Corel Draw.

3. Analisis dan Desain Sistem

3.1.Peraturan Permainan

Untuk mendukung analisis kebutuhan sistem diperlukan penjelasan *rules* / peraturan yang harus diikuti dalam permainan Capsah Banting, [McLeod, 2007], sebagai berikut:

- Permainan kartu Capsah Banting ini menggunakan kartu remi yang berjumlah 52 buah.
- Urutan ranking-nya adalah 3,4,5,6,7,8,9,10,J,Q,K,A,2 (2 adalah nilai terbesar).
- Sedangkan urutan gambarnya / *suits* dari kecil ke besar adalah *diamond* ♦ (tahu), *club* ♣ (keriting), *heart* ♥ (hati), *spade* ♠ (daun).
- Kartu dibagikan kepada empat main secara acak. Pemain yang mendapatkan giliran pertama adalah pemain yang mempunyai kartu 3 *diamonds*. Kemudian pemain lain harus mengeluarkan kartu yang lebih besar. Pemain yang kartunya habis terlebih dahulu dialah yang menang.

Urutan nilai kartu dari yang terkecil hingga yang terbesar, adalah sebagai berikut:

- *One card hands* / *single* : kartu apapun yang berjumlah 1 buah
- *Two card hands* / *pair* : sepasang kartu yang memiliki ranking (angka) yang sama. Contoh: **K♥ - K♦**
- *Three card hands* / *tris* : tiga buah kartu dengan ranking yang sama. Contoh: **8♥ - 8♠ - 8♣**
- *Five cards hands in order* :
  - *Straight* : kartu yang memiliki ranking berurutan.
    Contoh : **K♥ - Q♠ -J♦ - 10♣ - 9♥**
  - *Flush* : Lima buah kartu dengan gambar yang sama.
    Contoh : **9♥ - 7♥ - 6♥ - 5♥ - 3♥**
  - *Full house* : 2 kartu dengan ranking yang sama dan 3 kartu lain dengan ranking yang sama pula.
    Contoh : **8♥ - 8♠ - 8♣ - K♥ - K♦**
  - *Four of a kind /four / boom* : 4 kartu dengan ranking yang sama.
    Contoh : **9♥ - 9♠ - 9♦ - 9♣**
  - *Straight flush* : 5 buah kartu yang rankingnya berurutan dan memiliki gambar yang sama.
    Contoh : **A♠ - K♠ - Q♠ - J♠ - 10♠**

Tujuan akhir (*goal*) dari *game* ini adalah memaksimalkan akumulasi skor sepanjang permainan. Seorang pemain dinyatakan menang jika kartu telah habis,

dan pemain tersebut mendapatkan nilai positif (+) hasil akumulasi dari sisa kartu pemain yang kalah. Sebaliknya pemain yang kalah akan mendapatkan nilai negatif (-) dari jumlah kartu yang tersisa.

Strategi yang baik dalam permainan Capsah Banting, yaitu: harus dapat memaksimalkan kesempatan pemain dalam mengeluarkan kartu sebanyak-banyaknya, sekaligus memaksa musuh meminimalkan jumlah kartu yang dapat dikeluarkan. Meskipun pada kenyataannya terkadang kartu yang diharapkan memegang kontrol ternyata dapat dikalahkan musuh, hal ini dikarenakan tidak dapat mengetahui kartu musuh dan tidak dapat menduga strategi musuh (*imperfect information*) [Russel, 2003].

3.2.Penerapan *Minimax*

Tujuan sebenarnya algoritma *Minimax* digunakan pada permainan capsah ini yaitu untuk memperkirakan kartu apakah yang dapat dikeluarkan pada 1 langkah ke depan. [Kusumadewi, 2003] dan [Russel, 2003].

Komponen-komponen yang membentuk *Minimax* yaitu langkah pemain utama (max), langkah lawan (min), dan fungsi evaluasi. Langkah lawan dalam permainan Capsah Banting ini dilakukan oleh *virtual hand*, yaitu tumpukan kartu simulasi yang digunakan oleh komputer dan berisi semua kartu dikurangi kartu yang telah dibuang, kemudian dikurangi oleh kartu pemain tersebut. Sedangkan untuk dapat menghitung nilai evaluasi diperlukan beberapa rumus, yaitu rumus untuk penilaian kartu, penilaian kombinasi kartu, dan total nilai evaluasi.

Langkah utama dalam permainan Capsah Banting ini dibagi menjadi 2 yaitu, *first turn* (giliran pertama) dan *second turn / not first turn* (bukan giliran pertama). Kondisi *first turn* dan *second turn* memiliki perbedaan besar, dalam *first turn* pemain bebas mengeluarkan kartu apa saja (asalkan memenuhi syarat), sedangkan dalam *second turn* pemain harus mengeluarkan kategori kartu seseuai kartu yang sudah berada di meja (kartu yang telah dikeluarkan lawan).

Ketika pertama kali pemain menerima kartu, atau ketika hendak mengeluarkan kartu pada kasus *first turn* (giliran pertama), hal pertama yang harus dilakukan adalah mengurutkan kartu kemudian menganalisisnya. Setiap kartu atau kombinasi kartu mempunyai nilai tersendiri yang mengukur seberapa berharganya kartu tersebut. Hal ini mengakibatkan adanya prioritas dalam urutan pengeluaran kartu.

1. Virtual Hand

Strategi permainan akan berjalan dengan baik jika seorang pemain dapat menganalisa kartu yang dimilikinya (*hand*) dengan baik dan dapat memperkirakan kartu musuh. Seorang pemain agar dapat memperkirakan kemungkinan kartu musuh, idealnya mengingat semua kartu yang telah dikeluarkan oleh musuhnya. Untuk itu diperlukan suatu *hand* lagi, yang berisi semua kartu yang telah dibuang. Bila total 52 kartu dikurangi 13 kartu pemain (*user*), kemudian dikurangi kartu yang dibuang oleh lawan, akan didapatkan kondisi kartu lawan saat ini yang

selanjutnya disebut *virtual hand.* Beranjak dari sini, model matematika dari *virtual hand* adalah:

***Virtual Hand* = (T - Mh) – Pc**
T = total kartu dalam *deck*
Mh = kartu pemain (*user*) pada keadaan awal
Pc = kartu yang dikeluarkan oleh lawan dalam permainan

2. Nilai Kombinasi Kartu

Untuk mendapatkan nilai yang unik (tidak ada yang berulang) dan terurut dengan selisih 1 angka, setelah dilakukan observasi pada saat pengurutan kartu dalam permainan yang sebenarnya, dihasilkan rumus :

C = ((v + 1) * 4) + s
C = nilai sebuah kartu pada permainan Capsah
v = *value,* nilai angka dari sebuah kartu (3, 4, 5, dst)
s = *suits,* rangking dari kartu tersebut (*diamonds, clubs, hearts, spades*)

Angka 4 diperoleh dari jumlah *suits* yang ada, *value* ditambahkan dengan angka 1 untuk mencegah agar (v * 4) tidak bernilai 0, dan terakhir *value* ditambahkan dengan nilai s (♦ = 0, ♣ = 1, ♥ = 2, ♠ = 3) agar nilai kartu bertambah sebanyak 1 angka. Hasil penilaian kartu dapat dilihat pada Tabel 1.

Tabel 1 Penilaian Kartu

Kartu	V	S	((v+1)*4) + s
3♦	0	0	4
3♣	0	1	5
3♥	0	2	6
3♠	0	3	7
4♦	1	0	8
…	…	…	…
A♠	11	3	51
2♦	12	0	52
2♣	12	1	53
2♥	12	2	54
2♠	12	3	55

Aturan dalam permainan capsah banting memungkinkan seorang pemain untuk mengeluarkan lebih dari satu buah kartu, seperti *pair*, *tris*, *flush*, dsb. Tiap kombinasi kartu terbut harus memiliki nilai kelayakan sendiri. Hal lain yang menjadi perhitungan seorang pemain ketika hendak mengeluarkan kartu adalah jumlah kartu, karena tidak lain tujuan akhir dari permainan ini adalah membuang kartu sebanyak mungkin. Jadi dalam 1 langkah pemain harus bisa mencari

kombinasi kartu terbaik sekaligus jumlah kartu terbanyak yang dapat dikeluarkan dalam 1 langkah (maksimal 5 kartu dalam 1 langkah).

Model matematika untuk menghitung nilai kombinasi kartu adalah:

K = ((C1 + C2 + ... + Cn)/ n) + n
K = nilai kombinasi kartu
C = nilai kartu = ((v + 1) * 4) + s
n = jumlah kartu dalam kombinasi

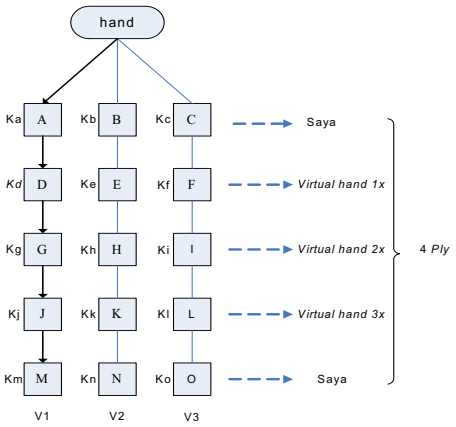
Sebagai contoh apabila dalam hand Max terdapat kartu 3♣, 3♥ kemungkinan kombinasi kartu yang dapat dikeluarkan adalah *single* 3♣, *single* 3♥, dan *pair* 3♣ 3♥. Berdasarkan rumus yang telah dibuat langkah yang dipilih Max adalah *pair* 3♣ 3♥. karena memiliki nilai terbesar yaitu -3,5, sedangkan nilai untuk *single* 3♣ adalah -4 dan *single* 3♥ adalah -5.

Fungsi evaluasi kombinasi kartu (K), akan bernilai negatif (-) bagi Max dan akan bernilai positif (+) bagi Min. Apabila Max tidak mengeluarkan kartu (*pass*) maka nilai K akan diisi dengan -55 $(K \leftarrow -\infty)$, dan bila Min tidak mengeluarkan kartu maka nilai K akan diisi dengan 0 $(K \leftarrow \infty)$.

3. Fungsi Evaluasi

Tujuan permainan capsah banting adalah menghabiskan kartu secepat mungkin. Tujuan ini akan diterapkan dalam algoritma *Minimax* dengan cara menjumlahkan nilai kombinasi kartu yang dikeluarkan pemain pada giliran pertama ditambah nilai kombinasi kartu pada langkah berikutnya. Dari hasil penjumlahan ini akan didapat nilai maksimal dari sebuah langkah. Selain itu perlu ditambahkan pula nilai efektifitas kartu. Nilai efektifitas didapat dengan menghitung selisih antara kartu lawan dengan kartu pemain.

Secara umum gambaran *Minimax* pada Capsah Banting dapat dilihat pada Gambar 1. Rangkaian proses yang dilakukan adalah pertama-tama pemain ("saya") mengeluarkan kartu, kemudian lawan (*virtual hand*) akan bereaksi dan mengeluarkan kartu, demikian seterusnya hingga *virtual hand* melakukan langkah sebanyak 3 kali. Kemudian pemain kembali mengeluarkan kartu, dan langkah terakhir adalah melakukan fungsi evaluasi dengan cara menjumlahkan nilai keseluruhan dari tiap *node*. Istilah *node* digunakan untuk menunjukkan setiap nilai dalam kombinasi kemungkinan dari setiap *virtual hand* pada setiap giliran melangkah.

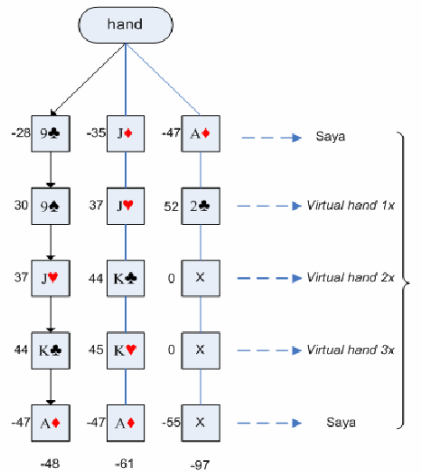


Gambar 1. Penerapan *Minimax*

V1 = Ka + (Ka – Kd) + (Ka – Kg) + (Ka – Kj) + Km
V2 = Kb + (Ka – Ke) + (Ka – Kh) + (Ka – Kk) + Kn
V3 = Kc + (Ka – Kf) + (Ka – Ki) + (Ka – Kl) + Ko

Dari hasil perhitungan fungsi evaluasi, pemain ("saya"), harus mengambil nilai *node* yang terbesar.

Sebagai contoh apabila *hand* seorang pemain adalah 9♣ J♦ A♦, dan lawan / *virtual hand* adalah 4♠ 7♣ 8♠ 9♠ J♥ K♣ K♥ K♠ 2♣. Maka pohon *Minimax* yang terjadi akan terlihat seperti pada Gambar 2.



Gambar 2. Penerapan *Minimax* dengan

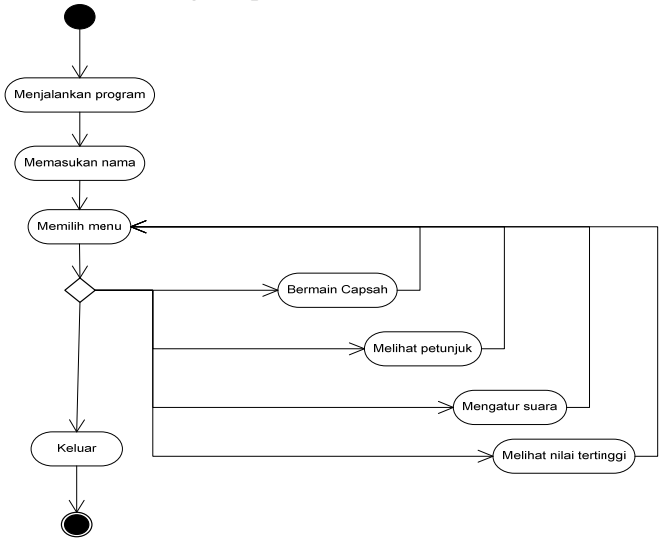
Dari contoh Gambar 2, maka pemain ("saya"), akan mengambil jalur permainan dengan nilai fungsi evaluasi –48, yaitu membuang 9♣.

4.2.Perancangan dan Implementasi

4.2.1.Diagram Aktivitas

Pada saat user menjalankan aplikasi, akan muncul halaman bergambar (*cover*), jika user mengklik tombol "mulai" maka user akan diminta untuk memasukkan nama, kemudian akan masuk ke halaman / tampilan utama. terpisah, yaitu modul permainan, modul pengaturan, modul petunjuk, dan modul skor / nilai tertinggi (lihat Gambar 3).

Ketika user memilih "mulai" dari halaman utama maka user akan masuk ke tampilan permainan (lihat Gambar 3). Ketika permainan dimulai, hal pertama yang dilakukan adalah melakukan proses pengocokan kartu, kemudian membagikan kartu ke setiap pemain hingga kartu habis (tiap pemain memiliki 13 kartu), dan mengurutkan kartu dari yang terkecil hingga terbesar. Permainan ini terdiri dari 4 orang pemain (3 komputer dan 1 user), setiap pemain akan memperoleh giliran untuk mengeluarkan kartu pada saat yang ditentukan. Dalam hal ini permainan berputar dari pemain 1 (user), pemain 2, pemain 3, pemain 4 dan kembali lagi ke pemain 1.



Gambar 3. Diagram Aktivitas Utama

Setelah itu akan terjadi pengulangan permainan (*looping*), dengan penerapan algoritma *Minimax* hingga kartu salah satu pemain habis, dan pemain yang kartunya habis terlebih dahulu dinyatakan sebagai pemenang, kemudian skor tiap pemain akan dihitung. Jika pemain memilih tombol "mulai" maka proses akan berulang seperti sebelumnya, jika tidak maka akan keluar ke tampilan utama dan nilai pemain (*user*) akan dicatat.

4.2.2.Diagram Kelas

Untuk memudahkan dalam pengerjaannya aplikasi permainan capsah ini dibuat menjadi beberapa kelas sesuai dengan karakteristik dan kebutuhannya. Kelas-kelas tersebut adalah *Card* (kartu), *CardDeck* (tumpukan kartu), *Hand* (kartu pemain), dan *VirtualHand* (kartu simulasi yang digunakan oleh

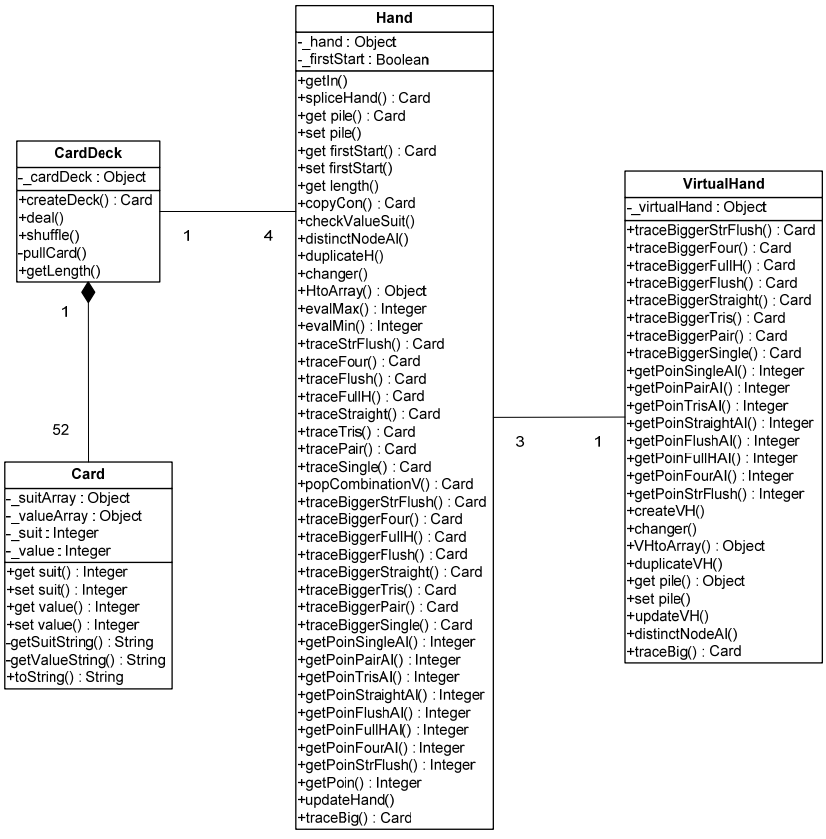
komputer), kemudian kelas-kelas tersebut akan dipanggil ke dalam program utama (Gambar 4).

Kelas *Card* merupakan struktur dasar dalam *game* Capsah Banting, dimana elemen dasar pembentuk sebuah *game* kartu tentu saja adalah sebuah kartu. Kelas *Card* mempunyai properties berupa *suit* (gambar) dan *value* (nilai).

Kelas CardDeck dibentuk dari Kelas Card. CardDeck merupakan sekumpulan objek (kartu) yang berjumlah 52 buah, dan siap dimainkan.

Kelas *hand* pada dasarnya sama dengan kelas CardDeck, hanya saja kelas *Hand* ini merupakan kartu yang telah dibagikan dari cardDeck ke setiap pemain. Dalam permainan Capsah Banting terdapat 4 *hand* yang mewakili 4 pemain.

Kelas *virtual hand* adalah *hand* yang sebenarnya tidak ada, tetapi digunakan untuk memperkirakan kartu yang akan dikeluarkan oleh lawan.



Gambar 4. *Class* Diagram

4.2.3. Implementasi Permainan

Pada modul permainan, terdapat empat tombol yang dapat diakses oleh pemain ("saya"). Pertama adalah tombol "kirim", tombol ini berguna untuk mengirim kartu ke meja permainan apabila pemain ("saya") telah menyeleksi kartu dengan cara mengklik pada kartu tersebut. Tombol "*pass"* digunakan apabila *v* berniat tidak akan mengeluarkan kartu, dan gilirannya akan dilewat. Tombol "mulai" berguna untuk memulai permainan. Tombol "mulai" hanya akan dapat diakses apabila tidak sedang dalam permainan. Dan yang terakhir adalah tombol keluar (berupa tanda panah) yang berguna untuk pindah ke halaman utama.

Modul permainan berisi kelas utama untuk keseluruhan permainan, seluruh *class* lainnya akan diimpor ke dalam modul ini.

```
import com.actionscript.games.CardDeck;
import com.actionscript.games.Card;
import com.actionscript.games.Hand;
import com.actionscript.games.VirtualHand;
```

Pada modul permainan ini, objek yang telah didefinisikan dalam kelas dideklarasikan.

```
var cd:CardDeck = new CardDeck();
cd.createDecks();
cd.shuffle();
var player1:Hand = new Hand();
var player2:Hand = new Hand();
var player3:Hand = new Hand();
var player4:Hand = new Hand();
var virtu2:VirtualHand = new VirtualHand;
virtu2.createVH(player1,player3,player4);
var virtu3:VirtualHand = new VirtualHand;
virtu3.createVH(player1,player2,player4);
var virtu4:VirtualHand = new VirtualHand;
virtu4.createVH(player1,player2,player3);
```

Komponen terpenting dalam permainan Capsah Banting ini terletak pada implementasi algoritma *Minimax* yang terdapat pada bagian permainan. Fungsi *Minimax* dibagi menjadi 2 yaitu *first turn* dan *second turn*. Jumlah *node* yang digunakan pada *first turn* adalah 5 buah, hal ini dikarenakan pencarian yang dilakukan dibatasi sampai giliran kembali lagi pada pemain.

Langkah yang dilakukan pada algoritma *Minimax first turn* adalah melacak semua kemungkinan kartu yang dapat dikeluarkan oleh pemain dan dimasukkan kedalam node1, setelah itu *virtual hand* melakukan pelacakan kartu yang dapat keluar sebanyak tiga kali (dari setiap lawan), dan disimpan dalam node2, node3, node4. Kemudian dari node4 lacak kembali semua kartu yang dapat dikeluarkan oleh pemain dan disimpan dalam node5. Kemudian Langkah terakhir adalah melakukan fungsi evaluasi terhadap kesemua *node* tersebut, dan mengambil nilai yang terbesar.

Pada kondisi *second turn* seorang pemain tidak memiliki kebebasan untuk mengeluarkan kartu. Kartu yang dapat dikeluarkan pada *second turn* haruslah kartu yang memiliki nilai lebih besar dari kartu lawan dan memiliki kategori kartu yang sama dengan kartu lawan tersebut. Seperti pada kondisi *first turn* pencarian yang dilakukan dibatasi sampai giliran kembali lagi pada pemain, hanya saja jumlah *node* yang digunakan pada *second turn* adalah 6 buah , hal ini dikarenakan node1diisi oleh kartu lawan yang telah dikeluarkan / telah berada dimeja (pada kondisi *first turn* keadaan meja adalah kosong), sehingga diperlukan 1 node lebih banyak untuk sampai kembali pada giliran pemain. Langkah pertama pada algoritma *Minimax second turn* adalah memasukkan kartu lawan kedalam node1, kemudian lacak semua kartu yang lebih besar dari kartu pada node 1 tersebut dan masukkan kedalam node2, setelah itu *virtual hand* melakukan pelacakan kartu yang dapat keluar sebanyak tiga kali dan disimpan dalam node3, node4 , node5. Kemudian dari node5 lacak kembali semua kartu yang dapat dikeluarkan oleh pemain dan dimasukkan ke dalam node6. Langkah terakhir adalah melakukan fungsi evaluasi terhadap ke-6 node tersebut.

5. Evaluasi Sistem

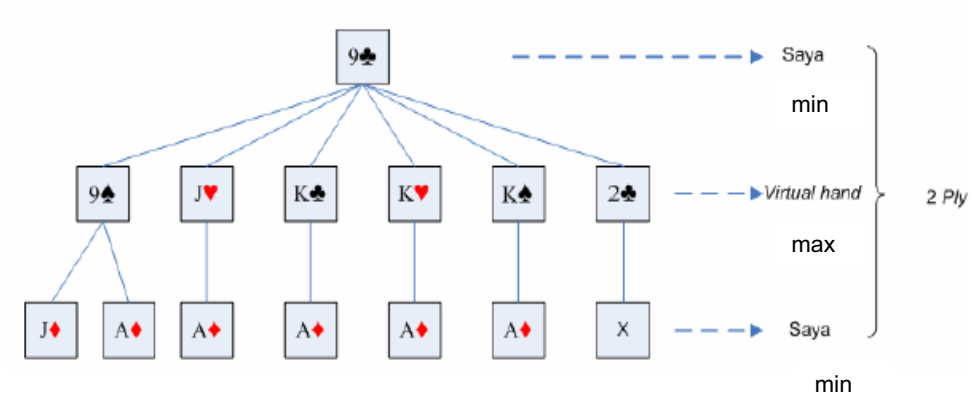
Algoritma *Minimax* dapat diimplementasikan dengan baik, hanya saja pada saat proses pelacakan pohon *Minimax* membutuhkan waktu yang cukup lama. Untuk melacak pohon pada kondisi *hand* pemain masih penuh, waktu pelacakan memakan waktu yang cukup lama. Hal ini dikarenakan, setiap aplikasi harus menghitung setiap kemungkinan langkah dari setiap pemain, dan mengevaluasi kombinasi kartu yang mungkin dikeluarkan oleh setiap pemain.

Contoh pelacakan yang terbentuk pada satu kali giliran permainan dapat dilihat dalam ilustrasi-ilustrasi berikut.

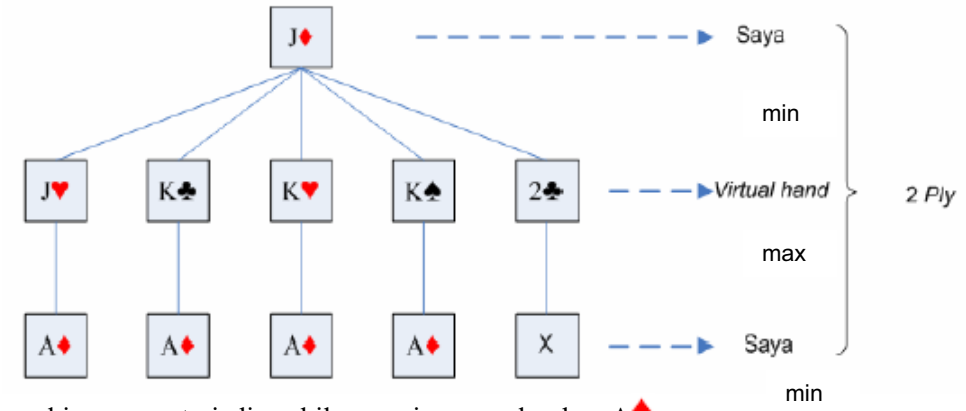
Diasumsikan kondisi permainan:

Player 1	9♣	J♦	A♦
Player 2	8♠	K♣	K♥
Player 3	4♠	7♣	J♥
Player 4	9♠	K♠	2♣

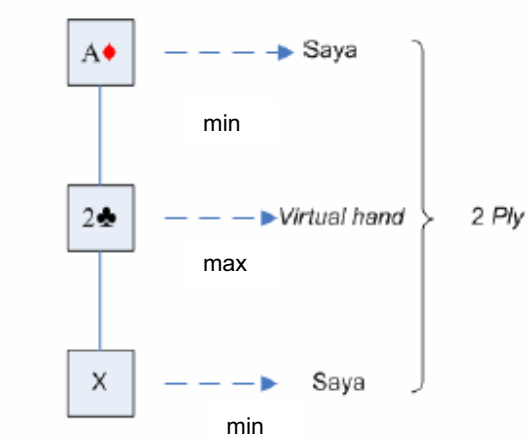
Asumsi pemain ("saya"), sebagai *player 1*, mendapatkan giliran pertama, kemungkinan yang terjadi apabila mengeluarkan 9♣ .



Kemungkinan yang terjadi apabila pemain mengeluarkan J♦:



Kemungkinan yang terjadi apabila pemain mengeluarkan A♦:



Dari hasil simulasi diatas langkah *player1* yang memungkinkan adalah J♦, A♦dan *pass.* Dalam pengujian, *player1* mengeluarkan 9♣, komputer dengan penerapan *Minimax*, melalui *player2* mengeluarkan K♠, *player* 3 *pass*, *player4* akan mengeluarkan K♠. Kemudian pada saat kembali giliran *player1*, *player1* hanya dapat mengeluarkan A♦.

Penerapan algoritma *Minimax* pada permainan Capsah Banting ini belum dapat membuat suatu sistem kecerdasan yang benar-benar handal. Hal ini dikarenakan terbatasnya kedalaman pencarian pada algoritma *Minimax*, yang hanya untuk satu putaran.

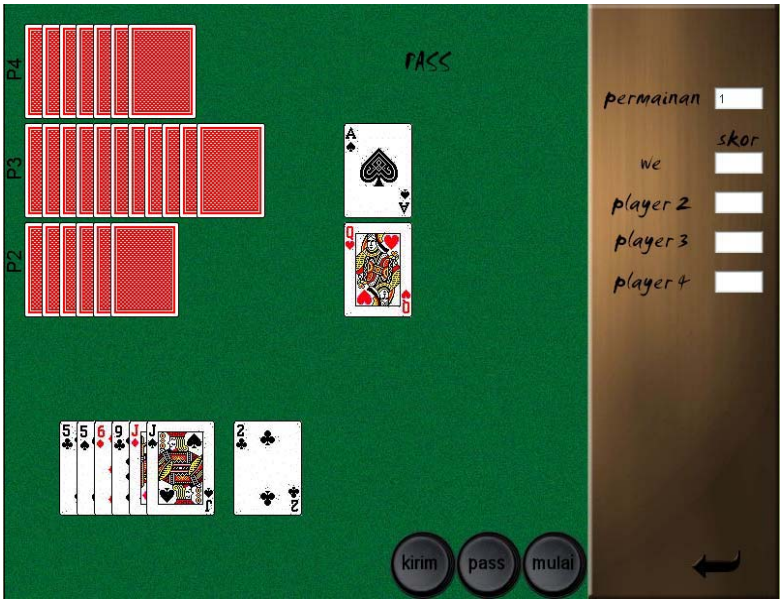
Contoh lain sebuah keputusan yang diambil oleh algoritma *Minimax* dalam sebuah permainan (Gambar 5) :

Player1: memulai denga 2 Club.
Player2:

```
Kartu Player 2:  8 Hearts, 9 Diamonds, 9 Spades, Queen Hearts, Queen Spades,
King Diamonds, 2 Spades
        Kombinasi kartu dan poin : Queen Hearts, 151
        Kombinasi kartu dan poin : Queen Spades, 147
        Kombinasi kartu dan poin : King Diamonds, 143
        Kombinasi kartu dan poin : 2 Spades, 30
        Kombinasi kartu dan poin : pass,  -19
        Hasil yang dipilih : Queen Hearts
```

Player3:

```
Kartu Player 3: 3 Hearts, 4 Spades, 5 Hearts, 6 Clubs, 7 Spades, 8 Clubs, 8
Spades, Jack Clubs, Jack Hearts, As Spades, 2 Diamonds, 2 Hearts
        Kombinasi kartu dan poin : As Spades, 132
        Kombinasi kartu dan poin : 2 Diamonds, 129
        Kombinasi kartu dan poin : 2 Hearts, 79
        Kombinasi kartu dan poin : pass, -34
        Hasil yang dipilih : As Spades
```



Gambar 5. Contoh Permainan

6. Kesimpulan

Dari proses pembuatan aplikasi permainan Capsah Banting ini dapat diambil beberapa kesimpulan sebagai berikut :

1. Algoritma *Minimax* dapat diterapkan pada permainan Capsah Banting, hanya saja kekurangannya terletak pada kedalaman pencarian pada pohon, yang hanya untuk satu putaran. Di samping itu, waktu pelacakan dirasakan cukup lama terutama ketika kombinasi yang dapat dikeluarkan sangat banyak.
2. Capsah Banting merupakan permainan kartu yang melibatkan faktor peluang. Faktor peluang yang dimaksud disini adalah kesempatan seorang pemain mendapatkan kartu bernilai tinggi atau rendah ketika kartu dibagikan pada awal permainan. Karena itu sebaik apapun algoritma yang digunakan tetap memiliki peluang kekalahan yang besar jika kartu yang dimiliki bernilai rendah.
3. Permainan Capsah Banting merupakan permainan dengan *imperfect information*, artinya pemain tidak memiliki informasi keadaan lawan secara lengkap. Karena factor ini pelacakan pada algoritma *Minimax* tidak bekerja seakurat jika diterapkan pada permainan catur atau tic-tac-toe yang kategori informasinya berupa *perfect information*.
4. Kehandalan algoritma *Minimax* dalam mencari solusi sangat bergantung pada fungsí evaluasi yang digunakan.
5. Macromedia Flash 8 dengan bahasa pemrograman *Actionscript 2.0* dapat digunakan untuk membuat aplikasi dengan basis *Object Oriented Programing.*

Daftar Pustaka

[Underdahl, 2002] Underdahl, Brian. (2002). *The Complete Reference Macromedia Flash MX*, Mc Graw Hill, Osborne.

[Lott, 2005] Lott, Joey and Reinhardt, Robert (2005) *Flash 8 Action Script Bible*, Wiley Publishing.Inc, Canada.

[McLeod, 2007] McLeod, John (2007) *Rules of Card Games: Big Two*, Access from: http://www.pagat.com/climbing/bigtwo.html, Sept 2007

[Russel, 2003] Russel, Stuart & Norvig, Peter (2003). *Artificial Intelligence A Modern Approach,* Prentice Hall, new Jersey.

[Schmuller, 2004] Schmuller, Joseph (2004). Sams *Teach Yourself UML in 24 Hours*, Third Edition, Sams Publishing, United States of America.

[Kusumadewi, 2003] *Artificial Intelligence (Teknik dan Aplikasinya)*, Graha Ilmu, Yogyakarta.