

# Implementasi Algoritma Runut Balik pada Permainan *Texas Hold'em Poker*

Yosef Ardhito Winatmoko / 13509052<sup>1</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

<sup>1</sup>13509052@std.stei.itb.ac.id

**Abstrak**—Permainan *Texas Hold'em Poker* adalah salah satu variasi dari permainan poker yang saat ini sedang banyak dimainkan. Pada dasarnya poker dianggap sebagai salah satu bentuk judi. Di luar konteks tersebut, ada berbagai model permasalahan menarik pada permainan poker yang dapat diselesaikan dengan cepat oleh komputer. Salah satu bentuk permasalahan adalah mengenai kemungkinan kombinasi kartu pada sebuah permainan poker yang menjadi salah satu dasar pemain dalam mengambil keputusan. Algoritma runut balik dapat digunakan untuk menelusuri setiap kemungkinan kombinasi kartu dengan baik.

**Kata Kunci**— Kartu, Kombinasi, Poker, Runut Balik

## I. PENDAHULUAN

Permainan poker bagi sebagian besar orang hanya dipandang sebelah mata karena permainan poker sudah di cap sebagai sebuah judi. Padahal permainan poker sendiri memiliki banyak sekali model masalah yang menarik untuk diselesaikan. Pada makalah ini, penulis tidak memperhatikan faktor judi dari permainan ini melainkan fokus kepada aturan yang terdapat pada poker sebagai batasan dalam menyelesaikan masalah. Variasi poker yang dipilih adalah *Texas Hold'em Poker* karena kini sedang berkembang terutama dimainkan di jejaring sosial seperti *facebook*, *tagged*, dan juga aplikasi *mobile* seperti pada OS Android dan iOS.

Fokus masalah yang ingin diselesaikan adalah salah satu masalah yang paling mendasar dari permainan poker. Permainan poker selalu berdasar kepada keputusan yang diambil setiap pemain sebagai reaksi atas kondisi yang sedang menimpa dirinya yaitu kondisi kartu di tangan, kondisi kartu di meja, dan keputusan yang diambil oleh pemain lain. Pada permainan poker, hal terpenting tidak hanya mengenai seberapa sering kartu kita lebih baik dari semua lawan kita tetapi juga bagaimana kita mengetahui bahwa kartu kita lebih buruk dari pemain lain.

Masalah yang diambil adalah mengenai kemungkinan kombinasi kartu pada setiap urutan babak di sebuah permainan poker. Dengan mengetahui persentase kombinasi kartu pada akhir babak, dipastikan keputusan

yang diambil juga lebih baik. Walaupun sebenarnya kejadian di akhir babak tidak selalu kombinasi yang memiliki persentase paling besar, pemain yang mengetahui bahwa persentase kombinasi kartunya sangat buruk akan lebih memilih untuk mundur dan menghindari dari kehilangan yang lebih besar.

Algoritma yang dipilih adalah runut balik. Sebenarnya runut balik adalah algoritma yang baik dipakai untuk melakukan pencarian. Disini penulis memanfaatkan kemampuan memotong pencarian yang dimiliki oleh algoritma runut balik untuk melakukan iterasi terhadap setiap kemungkinan kombinasi kartu, membuat perhitungan persentase jauh lebih cepat.

## II. TEXAS HOLD'EM POKER

### A. Deskripsi Umum

*Texas hold'em poker* atau biasa dikenal sebagai *hold'em* atau *holdem* adalah sebuah variasi permainan kartu poker. Permainan ini biasa dimainkan oleh dua atau lebih pemain, maksimal sembilan orang pemain. Setiap pemain akan mendapatkan dua kartu dan akan terdapat sejumlah kartu di meja, dibagikan oleh bandar. Pembagian dimulai dengan tiga kartu terlebih dahulu, biasa disebut *flop*, kemudian sebuah kartu yang disebut *turn*, terakhir sebuah kartu yang disebut *river*. Setiap putaran, pemain memiliki empat kemungkinan keputusan yaitu *check*, *bet*, *raise*, atau *fold*.

Objektif dari permainan *Texas hold'em* adalah mengumpulkan chip yang dimiliki oleh lawan hingga tidak ada lawan lagi yang memiliki *chip*. Inti dari permainan ini adalah betapa acaknya kartu yang dibagikan dan bagaimana pemain bisa menentukan kapan ia akan melanjutkan permainan, berarti mengharapkan hasil kombinasi kartu yang lebih baik dari lawan, atau keluar dari permainan jika sudah yakin bahwa lawan memiliki kartu yang lebih baik. Pada sekali permainan, biasanya hanya terdapat satu orang pemenang walaupun tidak menutup kemungkinan bahwa jumlah *chip* yang ada di meja akan dibagikan ke lebih dari satu pemain.

Probabilitas memiliki peran yang sangat besar pada permainan poker. Kemenangan sesungguhnya tidak selalu

berarti mendapatkan *chip* yang ada di meja, tetapi lebih kepada menentukan keputusan yang tepat pada setiap kombinasi kartu dan dalam melakukan prediksi terhadap kombinasi kartu tangan dengan kartu di meja. Dengan membuat keputusan - keputusan yang tepat, pemain dapat memaksimalkan pendapatan pada saat kartunya memang baik dan meminimalisasi pengeluaran sehingga berbuah pada kemenangan jangka panjang.

### B. Peraturan Dasar

Alur permainan *texas hold'em poker* ditentukan oleh sebuah token (*dealer button*) dan Ante (jumlah *chip* terkecil yang boleh dimainkan). *Dealer button* menentukan pemain yang bertindak sebagai *small blind* dan *big blind*. Dua buah token tersebut terus menerus berputar (searah jarum jam) untuk setiap pemain sehingga pemain yang mendapatkan *small blind* harus membayar ante, sedangkan pemain yang mendapatkan *big blind* harus membayar dua kali jumlah ante untuk bermain. Jika pemain memutuskan untuk tidak membayar, pemain tersebut dianggap *fold*, artinya keluar dari permainan kali itu dan harus mengembalikan dua buah kartu yang sudah dibagikan. Pada permainan yang hanya melibatkan dua orang pemain maka secara bergantian pemain akan bertindak sebagai *big blind* dan *small blind* dengan lawannya.

Salah satu tipe *Texas hold'em poker* yang paling terkenal adalah *no-limit*, dimana jumlah *chip* yang akan dimainkan oleh seorang pemain ke meja tidak memiliki batas maksimal, batasnya hanya sejumlah *chip* yang pemain tersebut miliki. Walaupun tidak ada batas maksimal dari jumlah yang diletakkan, pemain yang ingin melakukan *raise* atau *bet* tetap memiliki batas minimum yaitu sama dengan jumlah *bet* atau *raise* sebelumnya. Mereka yang melakukan *re-raise* harus meletakkan lebih dari atau sama dengan *raise* sebelumnya. *Raise* yang dilakukan pemain dengan meletakkan seluruh *chip* yang ia miliki saat itu diberi nama *All-in*. Saat ada pemain lain yang melakukan *call* atau *raise* terhadap *All-in*, pemain yang melakukan *All-in* akan dinyatakan keluar dari permainan jika kombinasi kartu yang ia miliki lebih jelek dari pemain yang melakukan *call* atau *raise*.

### C. Fase Permainan

Permainan *texas hold'em poker* dapat dibedakan menjadi empat fase utama yaitu : *preflop*, *flop*, *river*, dan *turn*. Tabel I menunjukkan jumlah kartu di player dan meja pada setiap fase. Pemain bisa memutuskan untuk berhenti bermain di fase apapun jika ia menganggap bahwa kartu yang ia miliki memiliki kombinasi yang lebih buruk daripada yang dimiliki setiap lawannya.

Tabel I. Kondisi pembagian kartu pada setiap fase

No	Fase	Player (Kartu)	Meja (Kartu)
1	Preflop	2	0
2	Flop	2	3
3	River	2	4
4	Turn	2	5

### D. Kombinasi Kartu

Kemenangan pada permainan poker ditentukan oleh kombinasi kartu tangan setiap pemain yang masih ikut bermain pada fase *river*, kecuali sebelum fase *river* seluruh pemain sudah memutuskan untuk mundur kecuali seorang pemain. Urutan peringkat kombinasi kartu dari yang paling tinggi derajatnya ditunjukkan oleh Tabel II. dan contoh kombinasinya ditunjukkan oleh Gambar 1.

Tabel II. Kondisi pembentuk kombinasi kartu

Kombinasi	Kondisi
<i>Straight Flush</i>	Terdapat lima kartu berurutan dengan daun ( <i>suit</i> ) yang sama.
<i>Four of a Kind</i>	Terdapat empat kartu dengan nilai yang sama.
<i>Full House</i>	Terdapat tiga kartu yang sama nilainya dengan dua buah kartu yang juga sama nilainya.
<i>Flush</i>	Terdapat lima kartu dengan daun yang sama tetapi tidak berurutan.
<i>Straight</i>	Terdapat lima kartu yang berurutan tetapi tidak kelimanya memiliki daun yang sama
<i>Two Pair</i>	Terdapat dua <i>pair</i> yang berbeda.
<i>Pair</i>	Terdapat dua kartu yang memiliki nilai yang sama.
<i>High Card</i>	Kombinasi yang tidak termasuk kategori apapun di atas.



Gambar 1. Urutan kombinasi kartu

### III. ALGORITMA RUNUT BALIK

Algoritma runut balik adalah sebuah algoritma yang merupakan pengembangan dari algoritma DFS (*Depth-first search*). Perbedaan antara DFS dengan runut balik adalah terdapat fungsi pemangkas yang memotong simpul yang tidak menuju simpul tujuan. Sama dengan DFS, pada dasarnya algoritma runut balik juga merupakan sebuah algoritma yang digunakan terutama untuk masalah pencarian.

Seperti halnya DFS, runut balik akan melakukan pemeriksaan sebuah simpul dan membangkitkan anak simpul tersebut sampai ditemukan solusi atau simpul anak sudah pernah dibangkitkan. Algoritma DFS tidak menjamin ditemukannya solusi yang optimal dan biasanya diimplementasikan secara rekursif.

Perbedaan dengan DFS, pada runut balik kita memiliki fungsi pemangkas (*pruning*). Untuk setiap simpul yang dibangkitkan, kita dapat memeriksa menggunakan fungsi pemangkas apakah simpul tersebut mengarah ke solusi atau tidak. Jika memang simpul tidak mengarah pada solusi, kita tidak perlu membangkitkan anak simpul tersebut. Urutan tindakan yang diambil oleh algoritma runut balik adalah sebagai berikut:

1. Mulai iterasi dari simpul utama
2. Untuk setiap simpul, jika simpul adalah solusi maka kembalikan simpul. Stop.
3. Jika simpul bukan solusi, periksa apakah simpul mengarah pada solusi, jika tidak maka runut balik.
4. Jika simpul adalah solusi, bangkitkan semua anak simpul.
5. Untuk setiap anak simpul yang belum pernah dibangkitkan, *push* ke sebuah *container*.
6. Mulai runut balik untuk setiap anggota *container*.

### IV. DESKRIPSI MASALAH

Pada permainan *texas hold'em poker*, salah satu dasar seorang pemain untuk mengambil keputusan adalah seberapa tinggi kombinasi kartu yang mungkin ia dapatkan dengan memanfaatkan dua buah kartu yang ada di tangannya. Seringkali seorang pemain memutuskan untuk mundur dari permainan ketika perbandingan antara kemungkinan kombinasi kartu tinggi yang bisa ia dapatkan jauh lebih rendah daripada jumlah *chip* yang bisa ia dapatkan.

Permasalahannya disini adalah jumlah seluruh kombinasi sangat banyak, terutama pada fase *preflop* dan *flop*, sehingga sangat sulit untuk menentukan seberapa besar kemungkinan kombinasi yang kita dapatkan berada pada urutan yang lebih tinggi. Umumnya pemain hanya akan menerka dengan memutuskan kartu apa yang akan membawa kombinasi yang tinggi baginya dan melanjutkan permainan tanpa mengetahui seberapa besar kemungkinan keluarnya kartu tersebut. Jumlah kombinasi kartu yang mungkin tanpa memperhitungkan kartu yang ada di tangan lawan ditunjukkan oleh Tabel III.

Karena kesalahan menerka, seringkali pemain melakukan kesalahan dengan melanjutkan permainan pada sebuah kondisi yang sebenarnya tidak layak untuk dilanjutkan. Misalnya ketika kartu di tangan kita dan di meja dapat membentuk kombinasi *flush* dengan kemungkinan keluar kartu yang dapat membentuk *flush* adalah 1 : 20. Jika untuk melanjutkan permainan kita harus membayar 1 *chip* padahal walaupun menang kita hanya akan mendapatkan 10 *chip* tambahan, pembayaran tidak layak dilakukan karena keuntungan yang diperoleh hanya 1 : 10, jauh lebih kecil daripada kemungkinan keluarnya kartu yang kita inginkan yaitu 1 : 20.

**Tabel III. Jumlah kemungkinan kombinasi setiap fase**

Fase	Perhitungan Kemungkinan Kombinasi
<i>Preflop</i>	Kartu di meja = 0 kartu Kartu di deck = 50 kartu (52 total kartu dikurangi 2 kartu yang sudah dipegang pemain)
	Jumlah kemungkinan kombinasi = $50! / (50 - 5)!$ = $50! / 45!$ = $50 * 49 * 48 * 47 * 46$ = <b>254.251.200 kemungkinan</b>
<i>Flop</i>	Kartu di meja = 3 kartu Kartu di deck = 47 kartu (52 total kartu dikurangi 2 kartu yang sudah dipegang pemain dan 3 kartu di meja)
	Kemungkinan kombinasi = $47! / (47 - 2)!$ = $47! / 45!$ = $47 * 46$ = <b>2.162 kemungkinan</b>
<i>Turn</i>	Kartu di meja = 4 kartu Kartu di deck = 46 kartu (52 total kartu dikurangi 2 kartu yang sudah dipegang pemain dan 4 kartu di meja)
	Kemungkinan kombinasi = $46! / (46 - 1)!$ = <b>46 kemungkinan</b>
<i>River</i>	Kartu di meja = 5 kartu Kartu di deck = 45 kartu (52 total kartu dikurangi 2 kartu yang sudah dipegang pemain dan 4 kartu di meja)
	Kemungkinan kombinasi = $45! / (45 - 0)!$ = <b>1 kemungkinan</b>

Pada fase *preflop* dan *flop*, hampir mustahil pemain

dapat melakukan kalkulasi kemungkinan dengan cepat. Pada fase *turn*, jumlah kemungkinan lebih kecil dan memungkinkan pemain untuk menghitung, tetapi sulit dilakukan dalam waktu singkat. Dalam menentukan keputusan, waktu yang digunakan harus cepat. Sebuah poker dengan yang diberi waktu umumnya hanya diberikan lima sampai sepuluh detik untuk menentukan tindakan pada suatu fase.

Jadi, permasalahan yang ingin diselesaikan adalah *bagaimana melakukan kalkulasi kemungkinan kombinasi kartu berdasarkan kondisi kartu di tangan dan di meja*.

## V. METODE

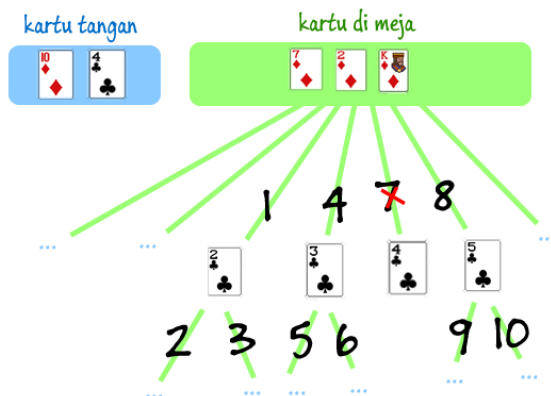
### A. Adaptasi Runut Balik untuk Poker

Untuk dapat menggunakan runut balik sebagai metode penyelesaian masalah, kita harus mengesampingkan kemungkinan penggunaan tabel yang sangat besar untuk menampung seluruh kemungkinan kombinasi kartu. Jika sudah tersedia tabel yang disebutkan tersebut, masalah ini cukup diselesaikan satu kali dan tidak perlu dibatasi waktu penyelesaiannya karena di lain waktu, pemain cukup mengacu kepada tabel tersebut.

Runut balik pada poker dapat digunakan dengan memanfaatkan kondisi kartu di meja sebagai ruang iterasi. Perbedaannya, runut balik tidak dimanfaatkan untuk menemukan sebuah solusi tunggal melainkan mengenumerasi setiap kombinasi yang mungkin dengan cepat melalui bantuan fungsi pemangkas. Perlu diperhatikan bahwa pada gambar yang menunjukkan contoh kasus, angka berwarna hitam hanya menunjukkan urutan kasar pembangkitan simpul berdasarkan algoritma runut balik.

#### Pemangkas 1: Kartu pada tangan pemain

Kartu yang sama juga tidak boleh muncul lebih dari satu kali, termasuk kartu yang sedang dipegang oleh pemain. Contoh implementasi runut balik pada fase *flop* ditunjukkan oleh Gambar 2.



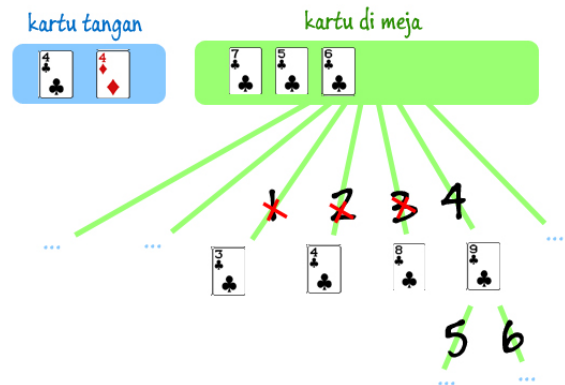
Gambar 2. Batasan kartu tangan pada fase flop

Pemangkasan ini dapat dilakukan dengan memanfaatkan daftar kartu yang terdapat pada *deck* kartu permainan.

#### Pemangkas 2: Batas atas kombinasi *straight flush*

Fungsi pemangkas batas *four of a kind* digunakan jika pada saat penelusuran sudah didapatkan minimal lima kartu yang membentuk kombinasi dimana syarat berikut berlaku:

- Sudah terdapat lima buah kartu dengan nilai berurutan dan daun yang sama



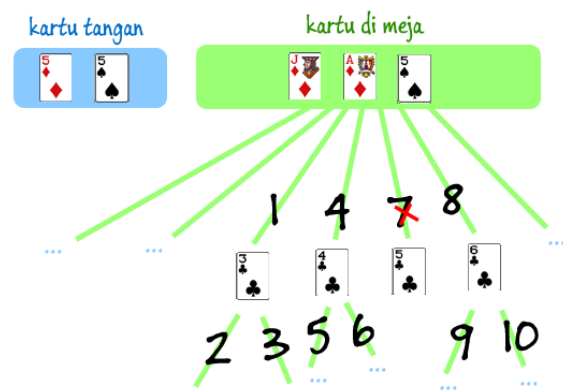
Gambar 3. Batasan *straight flush* pada fase flop

Pada kasus Gambar 3, untuk simpul kartu tiga keriting dan delapan keriting tidak perlu dilakukan pembangkitan anak karena sudah dapat dipastikan seluruh anaknya memiliki kasus terbaik yaitu *straight flush*. Kartu empat keriting dipangkas karena merupakan kartu yang sama dengan kartu tangan (pemangkas 1).

#### Pemangkas 3: Batas atas kombinasi *four of a kind*

Fungsi pemangkas batas *four of a kind* digunakan jika pada saat penelusuran sudah didapatkan minimal empat kartu yang membentuk kombinasi dimana syarat berikut berlaku:

- Sudah terdapat empat kartu dengan nilai yang sama



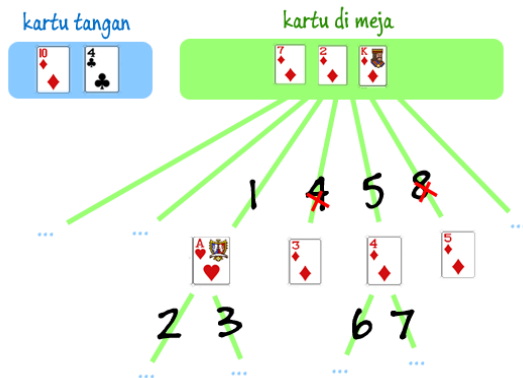
Gambar 4. Batas *four of a kind* pada fase flop

Dengan melihat pada simpul lima keriting maka dapat dipastikan seluruh anak simpulnya memiliki kasus terbaik *four of a kind*. Ini disebabkan kombinasi yang lebih baik dari *four of a kind*, yaitu *straight flush*, tidak mungkin lagi dicapai sementara kondisi *four of a kind* sudah dicapai dan merupakan kombinasi tertinggi yang dapat dicapai.

#### Pemangkas 4: Batas atas kombinasi *flush*

Fungsi pemangkas batas *flush* digunakan jika pada saat penelusuran sudah didapatkan minimal lima kartu yang membentuk kombinasi dimana syarat berikut berlaku:

- Terdapat lima kartu dengan daun yang sama
- Jika terdapat kartu dengan daun yang berbeda, tidak ada kartu dengan nilai yang sama dengan kartu tersebut.
- Sisa slot kartu tidak memungkinkan terbentuk lima kartu dengan nilai yang berurutan beserta daun yang sama (*Straight Flush*). Kemungkinan ini dapat dihitung dengan menentukan seluruh kombinasi kartu berdaun sama dengan selisih nilai maksimal lima.



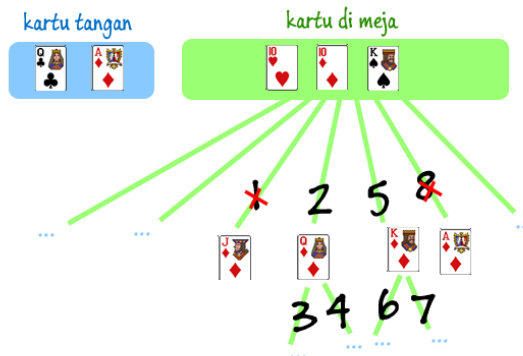
Gambar 5. Batas *flush* pada fase flop

Dengan melihat kemungkinan yang muncul pada simpul tiga wajah, maka dapat kita simpulkan bahwa simpul tersebut bersama seluruh simpul anaknya memiliki kasus kombinasi terbaik *flush* karena tidak mungkin membentuk *fullhouse*, *four of a kind*, maupun *straight flush*.

#### Pemangkas 5: Batas atas kombinasi *straight*

Fungsi pemangkas batas *straight* digunakan jika pada saat penelusuran sudah didapatkan minimal lima kartu atau lebih untuk membentuk kombinasi dimana syarat berikut berlaku:

- Terdapat lima kartu dengan nilai yang berurutan
- Dalam lima kartu yang berurutan terdapat empat kartu dengan daun yang masing – masing berbeda.



Gambar 6. Batas *straight* pada fase flop

Mirip dengan pemangkas – pemangkas sebelumnya, kasus Jack wajah membuat kombinasi terbaik yang dapat dibentuk oleh simpul ini dan anak – anaknya adalah *straight*.

#### B. Pseudocode Runut Balik

Dari algoritma dan sebuah fungsi pemangkas yang telah diuraikan maka dapat *pseudocode* untuk prosedur runut balik adalah sebagai berikut:

```

procedure inHand (input/output tableStack :
  stack of Card, input currentDeck : Deck,
  input/output result : array[0..9] of
  integer)
{ prosedur INHAND merupakan implementasi
  algoritma runut balik untuk perhitungan
  kemungkinan kombinasi terbaik setiap
  iterasi kartu pada permainan texas hold'em
  poker }

```

#### Kamus Lokal

```

tempStack : stack of Card
outDeck : Deck
needCheck : boolean

```

```

for i = 0 to currentDeck.numberOfCard do
  outDeck <- currentDeck
  tempStack <- tableStack
  tempStack.push(currentDeck.card[i])
  removeFromDeck(currentDeck, i)
  needCheck <- isPangkas(tempStack,
    result)

  if (not needCheck or
    tempStack.size() = 7) then
    inHand(tempStack, outDeck,
      result)

```

## VI. HASIL UJI DAN ANALISIS

### A. Hasil Uji

Implementasi dari *pseudocode* dan fungsi pemangkas didapatkan hasil uji dalam kondisi acak sebagai berikut :

#### Pengujian 1

Pada pengujian 1, secara acak pemain yang akan diperiksa mendapatkan kartu tangan sembilan keriting (*nine of club*) dan jack wajah (*Jack of diamond*).

#### Preflop

```
[COMMAND] Preflop$status
#####POKER HAND STRENGTH METER#####
[NOP] 2
[DECK] Card Left : 50
[TABLE] [None?][None?][None?][None?][None?]
[PLAYER] [9][C][J][D]
#####
[STATUS]
[COMMAND] Preflop$count
COMBINATION | POSSIBILITIES | %
Straight Flush | 36240 | 0.0143
Four of a Kind | 320160 | 0.126
Full House | 5225760 | 2.06
Flush | 4977120 | 1.96
Straight | 1192560 | 0.469
Three of a Kind | 11754720 | 4.62
Two Pair | 53301600 | 21
Pair | 117945840 | 46.4
High Card | 59497200 | 23.4
-----
waktu : 10 menit, 26 detik, 978 milisekon, 446 mikrosekond.
```

Gambar 7. Pengujian 1 fase preflop

```
[COMMAND] Stop$continue
[CONTINUE]
[COMMAND] Preflop$status
#####POKER HAND STRENGTH METER#####
[NOP] 2
[DECK] Card Left : 50
[TABLE] [None?][None?][None?][None?][None?]
[PLAYER] [Q][H][5][C]
#####
[STATUS]
[COMMAND] Preflop$count
COMBINATION | POSSIBILITIES | %
Straight Flush | 36240 | 0.0143
Four of a Kind | 320160 | 0.126
Full House | 5225760 | 2.06
Flush | 4977120 | 1.96
Straight | 1192560 | 0.469
Three of a Kind | 11754720 | 4.62
Two Pair | 53301600 | 21
Pair | 117945840 | 46.4
High Card | 59497200 | 23.4
-----
waktu : 10 menit, 30 detik, 334 milisekon, 178 mikrosekond.
```

Gambar 10. Pengujian 2 fase preflop

**Flop**

```
[COMMAND] Flop$status
#####POKER HAND STRENGTH METER#####
[NOP] 2
[DECK] Card Left : 47
[TABLE] [3][H][6][C][6][S][None?][None?]
[PLAYER] [9][C][J][D]
#####
[STATUS]
[COMMAND] Flop$count
COMBINATION | POSSIBILITIES | %
Straight Flush | 0 | 0
Four of a Kind | 2 | 0.0925
Full House | 54 | 2.5
Flush | 0 | 0
Straight | 0 | 0
Three of a Kind | 144 | 6.66
Two Pair | 798 | 36.9
Pair | 1164 | 53.8
High Card | 0 | 0
-----
waktu : 0 menit, 0 detik, 6 milisekon, 394 mikrosekond.
```

Gambar 8. Pengujian 1 fase flop

**Flop**

```
#####POKER HAND STRENGTH METER#####
[NOP] 2
[DECK] Card Left : 47
[TABLE] [A][D][T][C][9][C][None?][None?]
[PLAYER] [Q][H][5][C]
#####
[STATUS]
[COMMAND] Flop$count
COMBINATION | POSSIBILITIES | %
Straight Flush | 2 | 0.0925
Four of a Kind | 0 | 0
Full House | 0 | 0
Flush | 88 | 4.07
Straight | 30 | 1.39
Three of a Kind | 30 | 1.39
Two Pair | 178 | 8.23
Pair | 1012 | 46.8
High Card | 822 | 38
-----
waktu : 0 menit, 0 detik, 7 milisekon, 800 mikrosekond.
```

Gambar 11. Pengujian 2 fase flop

**Turn**

```
[COMMAND] Flop$continue
[CONTINUE]
[COMMAND] River$status
#####POKER HAND STRENGTH METER#####
[NOP] 2
[DECK] Card Left : 46
[TABLE] [3][H][6][C][6][S][J][H][None?]
[PLAYER] [9][C][J][D]
#####
[STATUS]
[COMMAND] River$count
COMBINATION | POSSIBILITIES | %
Straight Flush | 0 | 0
Four of a Kind | 0 | 0
Full House | 4 | 8.7
Flush | 0 | 0
Straight | 0 | 0
Three of a Kind | 0 | 0
Two Pair | 42 | 91.3
Pair | 0 | 0
High Card | 0 | 0
-----
waktu : 0 menit, 0 detik, 0 milisekon, 141 mikrosekond.
```

Gambar 9. Pengujian 1 fase turn

**Turn**

```
[COMMAND] River$status
#####POKER HAND STRENGTH METER#####
[NOP] 2
[DECK] Card Left : 46
[TABLE] [A][D][T][C][9][C][J][S][None?]
[PLAYER] [Q][H][5][C]
#####
[STATUS]
[COMMAND] River$count
COMBINATION | POSSIBILITIES | %
Straight Flush | 0 | 0
Four of a Kind | 0 | 0
Full House | 0 | 0
Flush | 0 | 0
Straight | 4 | 8.7
Three of a Kind | 0 | 0
Two Pair | 0 | 0
Pair | 18 | 39.1
High Card | 24 | 52.2
-----
waktu : 0 menit, 0 detik, 0 milisekon, 169 mikrosekond.
```

Gambar 12. Pengujian 2 fase turn

**Pengujian 2**

Pada pengujian 2, secara acak pemain mendapatkan kartu tangan ratu hati(*queen of heart*) dan lima keriting(*five of diamond*).

**Preflop**

**Pengujian 3**

Pada pengujian 3, secara acak pemain mendapatkan kartu tangan enam keriting(*six of club*) dan tiga wajik(*three of diamond*).

**Preflop**

```
[NEWGAME]
[COMMAND]      Stop$continue
[CONTINUE]
[COMMAND]      Preflop$status
#####POKER HAND STRENGTH METER#####
[NOP]          2
[DECK]         Card Left : 50
[TABLE]        [None?][None?][None?][None?]
[PLAYER]       [6C][3D]
#####
[STATUS]
[COMMAND]      Preflop$count
COMBINATION    POSSIBILITIES      %
Straight Flush | 4080      0.0016
Four of a Kind | 320160    0.126
Full House     | 5225760   2.06
Flush         | 5009280   1.97
Straight      | 118800    0.0467
Three of a Kind | 11754720  4.62
Two Pair      | 53301600  21
Pair          | 118306800 46.5
High Card     | 60210000  23.7
-----
waktu : 10 menit, 0 detik, 847 milisekon, 455 mikrosekond
```

Gambar 13. Pengujian 3 fase preflop

```
Flop
#####POKER HAND STRENGTH METER#####
[NOP]          2
[DECK]         Card Left : 47
[TABLE]        [Q][S][J][H][9][S][None?][None?]
[PLAYER]       [6C][3D]
#####
[STATUS]
[COMMAND]      Flop$count
COMBINATION    POSSIBILITIES      %
Straight Flush | 0          0
Four of a Kind | 0          0
Full House     | 0          0
Flush         | 0          0
Straight      | 0          0
Three of a Kind | 30        1.39
Two Pair      | 180       8.33
Pair          | 1044      48.3
High Card     | 908       42
-----
waktu : 0 menit, 0 detik, 7 milisekon, 849 mikrosekond
```

Gambar 14. Pengujian 3 fase flop

```
Turn
-----
[COMMAND]      Flop$continue
[CONTINUE]
[COMMAND]      River$status
#####POKER HAND STRENGTH METER#####
[NOP]          2
[DECK]         Card Left : 46
[TABLE]        [Q][S][J][H][9][S][3][S][None?]
[PLAYER]       [6C][3D]
#####
[STATUS]
[COMMAND]      River$count
COMBINATION    POSSIBILITIES      %
Straight Flush | 0          0
Four of a Kind | 0          0
Full House     | 0          0
Flush         | 0          0
Straight      | 0          0
Three of a Kind | 2          4.35
Two Pair      | 12         26.1
Pair          | 32         69.6
High Card     | 0          0
-----
waktu : 0 menit, 0 detik, 0 milisekon, 165 mikrosekond
```

Gambar 15. Pengujian 3 fase turn

### B. Analisis Hasil Uji

Dengan menggunakan *bruteforce*, kompleksitas algoritmanya merupakan  $O(n!)$  karena persoalan ini merupakan persoalan permutasi yang harus melakukan penelusuran ke semua kemungkinan.

Dari hasil pengujian terhadap permasalahan kombinasi ini *pseudocode* yang dihasilkan memiliki kompleksitas  $O(n!/(n-m)!)$  dimana  $n$  adalah jumlah kartu pada deck untuk setiap waktu yang kosong. sementara  $m$  adalah jumlah slot kosong yang dicari. Kompleksitas yang didapatkan itu sendiri merupakan kemungkinan terburuk

jika tidak terjadi pemangkasan sama sekali. Dari hasil percobaan ternyata ditemukan walaupun algoritma pemangkasan dapat dilakukan, tetapi pemangkasan yang terjadi bukan pada bagian yang memang berjumlah banyak. Sebagian besar simpul adalah simpul yang bukan hasil pemangkasan yaitu simpul dengan kombinasi lebih rendah dari *straight*. Bahkan jika dilihat kecenderungan simpul *full house* juga cukup besar sementara tidak ada pemangkasan akan simpul tersebut.

Bagaimanapun dengan runut balik dapat dipastikan akan lebih cepat daripada tanpa melakukan pemangkasan karena  $O(n!/(n-m)!)$  pada runut balik merupakan kasus terburuk. Waktu yang dibutuhkan untuk perhitungan juga tergolong lama dan masih belum bisa diterima untuk permainan poker. Hal ini juga disebabkan oleh fungsi pemangkasan yang tidak tepat sasaran sehingga memakan waktu memeriksa simpul – simpul dalam jumlah banyak dengan kasus mirip.

## VII. KESIMPULAN

Algoritma runut balik pada dasarnya digunakan untuk masalah pencarian dan tidak cukup baik untuk digunakan sebagai metode penyelesaian masalah penelusuran seluruh kemungkinan. Dalam memecahkan masalah penjumlahan seluruh kemungkinan kombinasi kartu dengan runut balik didapatkan kompleksitas  $O(n!/(n-m)!)$  yang tidak jauh berbeda jika dengan menggunakan *bruteforce*.

Ada dua cara untuk mempercepat penelusuran yaitu mencari kesempatan pemangkasan yang lebih tepat sasaran dan menggunakan fungsi pemangkasan yang memanfaatkan sifat kombinasi dimana urutan tidak membedakan satu kasus dengan kasus lainnya. Untuk solusi kedua, pencarian hanya akan lebih cepat jika sifat kombinasi yang dimanfaatkan cukup mangkus juga.

## DAFTAR REFERENSI

- [1] <http://en.wikipedia.org/wiki/Backtracking> diakses tanggal 6 Desember 2011.
- [2] <http://www.roomreview.net/winning-secrets-of-online-texas-holdem/> diakses tanggal 7 Desember 2011
- [3] [http://www.tightpoker.com/poker\\_odds.html](http://www.tightpoker.com/poker_odds.html) diakses tanggal 8 Desember 2011
- [4] Munir, Rinaldi, "Diktat Kuliah IF3051 Strategi Algoritmik", Program Studi Teknik Informatika, ITB, 2009.

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 29 April 2010

ttd

Yosef Ardhito W. / 13509052