

IF2211 Strategi Algoritma

Penerapan Algoritma Greedy pada Permainan *Riichi Mahjong*

Tadya Rahanady H/13509070
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13509070@std.stei.itb.ac.id

ABSTRAK

Makalah ini berisi pembahasan mengenai penerapan algoritma greedy pada salah satu jenis permainan mahjong, yaitu riichi mahjong. Permainan Riichi Mahjong merupakan variasi dari permainan Mahjong yang dimainkan di Jepang. Dalam makalah ini akan dijelaskan dua cara penggunaan algoritma greedy yang dapat digunakan di dalam permainan tersebut.

Kata kunci : algoritma greedy, riichi mahjong, tile, suit, wall, chii, pon, kan, dora, han, fu

I. PENDAHULUAN

Permainan *Riichi Mahjong* atau *Japanese Mahjong* adalah salah satu variasi dari permainan *Mahjong*. Dalam *Riichi Mahjong*, terdapat beberapa peraturan tambahan yang tidak terdapat pada permainan *Mahjong*. Permainan ini dibawa ke Jepang oleh seorang tentara yang bernama Saburo Hirayama pada tahun 1900-an. Dia membuka klub, sarana bermain, dan sekolah *Mahjong* di Tokyo. Pada tahun-tahun setelahnya, kepopuleran permainan ini meningkat secara drastis. Dalam prosesnya, permainan ini merupakan penyederhana dari versi aslinya. Lalu, terdapat beberapa peraturan baru yang ditambahkan untuk meningkatkan tingkat kesulitannya. Akan tetapi, orang Jepang berhasil mengubahnya tanpa benar-benar mengganti keseluruhan sifat-sifat dasarnya yang mengakibatkan permainan ini tetap menjadi salah satu permainan yang populer.

Berdasarkan pada tahun 2010, permainan ini merupakan *table game* paling populer di Jepang. Pada tahun 2008, terdapat kurang lebih 7.6 juta pemain dan sekitar 8900 sarana permainan yang menghasilkan 300 milyar yen. Terdapat juga beberapa *manga* dan *anime* yang menggunakan permainan ini sebagai fokus cerita yang ditampilkan. Permainan ini juga terdapat dalam *video arcades* yang dapat menghubungkan pemain-pemain yang ada melalui internet.

II. RIICHI MAHJONG

Dalam *Riichi Mahjong*, tujuan dari pemain adalah untuk mendapatkan poin tertinggi dalam ronde permainan yang telah ditentukan. Setiap ronde terdiri dari 4 putaran. Pada setiap putaran pemain harus berusaha mengumpulkan set tangan yang bernilai tinggi untuk menambah poin milik sendiri dan mengurangi poin yang dimiliki oleh pemain lainnya.

Riichi Mahjong dimainkan dengan menggunakan 136 buah *tiles*. Setiap *tiles* diacak dan disusun menjadi empat *wall* yang terdiri dari dua tingkat dan memuat 17 *tiles*. Pemain mengambil 26 tumpukan pertama dari *wall* untuk menjadi *tile* tangan, 7 urutan terakhir untuk menjadi *dead wall*, dan sisanya menjadi *playing wall*.

A. Tile Suit

Pada *Riichi Mahjong* terdapat 34 *tile* berbeda yang masing-masingnya berjumlah 4 buah. Berikut adalah jenis-jenis *tile* tersebut :

1. Pin (Lingkaran)



Gambar 1 Tile pin

2. Sou (Bambu)



Gambar 2 Tile sou

3. Wan (Karakter)



Gambar 3 Tile wan

4. Wind (Arah Mata Angin)



Gambar 4 Tile angin

5. Dragon (Naga)

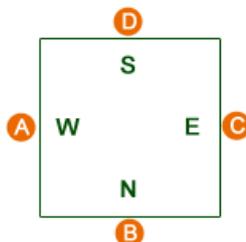


Gambar 5 Tile naga

Pada awal permainan tiap pemain masing-masing mendapatkan 13 buah *tile*. Setiap giliran pemain akan menarik satu *tile* baru dari *playing wall*, dan membuang satu *tile*, ke dalam *discard pile*. Seperti yang sudah disebutkan, pemain akan memenangkan putaran ketika dia berhasil membentuk sebuah set berjumlah 14 *tile* di tangannya. Set yang biasa dibuat adalah empat set yang terdiri dari set *chii* dan/atau *pon/kan*, dan satu set pasangan. Pemain juga harus memiliki minimal satu *han*. Setiap buangan dari pemain ditaruh terbuka, dan pemain lain berhak untuk mengambil *tile* yang telah dibuang tersebut.

B. Seat

Dalam *Riichi Mahjong* setiap posisi duduk memiliki Arah Mata Angin yang akan diperhitungkan untuk penjumlahan poin. Pemain dapat menentukan siapa *dealer* (pemain yang jalan pertama) dengan dadu. *Dealer* akan selalu memiliki arah mata angin Timur. Ketiga pemain lainnya mendapatkan Arah Mata Angin sesuai dengan gambar berikut :



Gambar 6 Arah Mata Angin

Permainan lalu akan dilakukan berlawanan dengan arah

jarum jam. Setiap putaran Mata Angin Timur akan berpindah berlawanan arah jarum jam. Putaran selesai saat Mata Angin Timur kembali pada pemain awal. Pada putaran selanjutnya, pemain pertama mendapatkan arah Mata Angin Selatan, dst.

C. Set

Pada *Riichi Mahjong* terdapat beragam jenis set yang dapat dibentuk untuk memainkan permainan. Set tersebut biasanya terdiri dari :

1. *Chii*



Gambar 7 Set chii

Chii merupakan set yang terdiri dari 3 *tile* berurutan. Pemain hanya boleh mengambil buangan dari pemain di sebelah kirinya untuk membentuk set ini.

2. *Pon*



Gambar 8 Set pon

Pon merupakan set yang terdiri dari 3 *tile* yang sama. Pemain boleh mengambil buangan dari pemain manapun untuk membentuk set ini. Set ini juga memiliki prioritas yang lebih tinggi daripada set *chii*. (e.g. Pemain A ingin men-*chii* sebuah *tile* yang dijatuhkan pemain B, tetapi pemain C akan mem-*pon* *tile* tersebut. Pada kasus tersebut, maka pemain C yang akan mendapatkan *tile* yang dibuang.)

3. *Kan*



Gambar 9 Set kan

Kan merupakan set yang terdiri dari 4 *tile* yang sama. Pada dasarnya *kan* sama dengan *pon*. Jika pemain memiliki satu atau lebih *kan*, jumlah *tile* yang dimiliki akan bertambah sebanyak *kan* yang dimiliki. Jika pemain melakukan *kan*, pemain mengambil *tile* baru dari *dead wall* dan *dora tiles* dari *dead wall* akan dibuka. Permainan juga akan diulang jika terdapat empat *kan* di dalam permainan. Jika keempat *kan* tersebut dimiliki oleh satu pemain, permainan akan diulang saat *kan* kelima dilakukan oleh siapapun.

4. Eye



Gambar 10 Set mata

Eye merupakan set yang terdiri dari 2 *tile* yang sama. Pemain tidak dapat membuat set ini dari buangan pemain lain kecuali saat pemain akan memenangkan permainan..

D. Dora tiles

Dora adalah *tile* special yang memberikan poin tambahan. Setiap *dora tile* yang digunakan akan menambah nilai *han*.

E. Han

Han adalah pengali yang melipatgandakan nilai dari set yang dibuat. Untuk memenangkan permainan, pemain harus memiliki minimal satu *han*. Jika pemain melakukan *chii* dan/atau *pon/kan* dari *discard* pemain lain, nilai *han* akan berkurang sebanyak 1.

F. Yaku

Yaku adalah kondisi yang menentukan jumlah *han* seorang pemain. Setiap *yaku* memiliki jumlah *han* yang berbeda, dan dapat juga digabungkan untuk mendapatkan *han* yang lebih tinggi. Berikut adalah daftar *yaku* yang terdapat dalam *Riichi Mahjong* :

1. Satu *han*
 - a. *Closed Tsumo*
Set tertutup dan menang dengan menarik *tile* dari *playing wall*.
 - b. *Colors*
Satu set *pon tile* naga. Dapat terbuka ataupun tertutup.
 - c. *Table Wind*
Pon dari Arah Mata Angin pada putaran yang bersangkutan. Dapat terbuka ataupun tertutup.
 - d. *Seat Wind*
Pon dari Arah Mata Angin masing-masing pemain, sama dengan *Table Wind* untuk *dealer*. Dapat terbuka ataupun tertutup.
 - e. *Ready Hand*
Kapanpun saat tangan pemain tertutup dan hanya membutuhkan satu *tile* untuk menang, pemain dapat melakukan *riichi*. *Riichi* hanya dapat dilakukan jika masih tersisa *tile* di dalam *playing wall*.
 - f. *Flat Hand*
Set yang terdiri empat *chii* dan *eye*. Tidak

boleh memiliki *pon* atau *kan*, naga, angin putaran, dan angin pemain. Hanya boleh tertutup.

- g. *Inside Hand*
Set tanpa *terminal* (Angka 1 dan 9) dan naga.
 - h. *Double Run*
Set dengan dua *chii* yang sama. Hanya boleh tertutup.
 - i. *Instant*
Instant adalah bonus 1 *han* saat pemain melakukan *riichi* dan memenangkan permainan sebelum pemain membuang *tile* pada gilirannya kembali.
 - j. *Last Tile Ron*
Menang dengan menggunakan *tile* terakhir.
 - k. *Out on Extra Tile Draw*
Menang dengan *tile* tambahan setelah melakukan *kan*.
 - l. *Late Kan Ron*
Menang dengan menambahkan *tile* pada *pon* terbuka untuk membentuk *kan*.
2. Dua *han*
 - a. *Triple Run*
Tiga *chii* pada masing-masing *suit* dengan urutan yang sama. Jika terbuka, hanya mendapatkan satu *han*.
 - b. *Straight*
Tiga *chii* dengan *tile* 1-9 pada *suit* yang sama. Jika terbuka, hanya mendapatkan satu *han*.
 - c. *All Sets*
Empat *pon*.
 - d. *Half Outside*
Setiap set harus mengandung *terminal* dan angin/naga. Jika terbuka, hanya mendapatkan satu *han*.
 - e. *Triple Pon*
Tiga *pon* dengan angka sama pada setiap *suit*.
 - f. *3-Closed Sets*
Tiga *pon* tertutup. Menang dengan melakukan *tsumo*(menarik *tile* dari *playing wall*).
 - g. *All Pairs*
Set spesial yang dibentuk dengan tujuh *eye*. Tidak dapat menggunakan *eye* yang sama.
 - h. *Double Reach*
Riichi yang dilakukan pada awal permainan.
 - i. *Three Quads*
Set dengan tiga *kan*.
 3. Tiga *han*
 - a. *Half Flush*
Set dengan salah satu *suit* dan angin/naga. Jika terbuka, hanya mendapatkan dua *han*.
 - b. *Full Outside*
Setiap set mengandung *terminal* dengan minimal satu *chii*. Jika terbuka, hanya mendapatkan dua *han*.

- c. *Double-Double Run*
Set dengan dua *Double Run*.

- 4. Empat *han*
 - a. *Lesser Colors*
Set dengan dua *pon* naga dan *eye* naga ketiga.
 - b. *All Ends*
Set hanya berisi *terminal* dan angin/naga.
 - c. *Quick Out*
Dapat terjadi ketika pemain menang dengan buangan pertama pemain lain sebelum ada *chii*, *pon*, dan *kan*.
- 5. Enam *han*
 - a. *Full Flush*
Set yang hanya terdiri dari satu *suit*. Jika terbuka, hanya mendapatkan lima *han*.
- 6. *Limit* (13+ *han*)
 - a. *Four Closed Sets*
Set dengan empat *pon* tertutup.
 - b. *All Crap*
Set yang terdiri dari 1 dan 9 dari semua *suit*, empat angin yang berbeda, tiga naga yang berbeda, dan satu pasangan dari ketigabelas *tile* tersebut. Hanya boleh tertutup.
 - c. *Greater Colors*
Tiga *pon* atau *kan* dari semua *tile* naga.
 - d. *Four Winds*
Tiga *pon* angin dan *eye* angin keempat, atau empat *Pon* angin.
 - e. *All Green*
Set yang terdiri dari 2, 3, 4, 6, 8 bambu dan *tile* naga hijau.
 - f. *All Honors*
Set yang hanya terdiri dari angin dan naga.
 - g. *All Terminals*
Set yang hanya terdiri dari *terminal*(1 dan 9).
 - h. *Out From Start*
Non-dealer menang dengan *discard* pertama *dealer*.
 - i. *Nine Treasures*
Set dengan 1, 1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 9, 9 pada salah satu *suit*. Dapat menang dengan *tile* 1, 2, 3, 4, 5, 6, 7, 8, atau 9. Hanya boleh tertutup.
 - j. *Four Quads*
Set dengan empat *kan*.

G. *Fu*

Fu adalah nilai dari set dan *tile* yang ditunggu. *Fu* digunakan dalam perhitungan poin

H. *Scoring*.

Perhitungan poin pada *Riichi Mahjong* adalah sebagai

berikut :

$$basic\ points = fu \times 2^{(2+han)}$$

1. Jika *non-dealer* menang dengan *tsumo*, *dealer* membayar pemenang 2 x *basic point*, pemain lain membayar 1 x *basic point*.
2. Jika *non-dealer* menang dengan *ron*(menang dengan *discard* pemain lain), pemain tersebut membayar pemenang 4 x *basic point*.
3. Jika *dealer* menang dengan *tsumo*, pemain lain membayar 2 x *basic point*.
4. Jika *dealer* menang dengan *ron*, pemain tersebut membayar 6 x *basic point*.

Akan tetapi, jika *basic point* yang didapatkan sudah jelas lebih dari 2000, akan dibatasi menjadi 2000 dan disebut *mangan*. Jenis-jenis *mangan* adalah :

1. *Mangan*
Jika set memiliki 3 *han* dan 70 *fu* atau lebih, 4 *han* dan 40 *fu* atau lebih, atau 5 *han*. *Basic point* 2000.
2. *Haneman*
Jika set memiliki 6 atau 7 *han*. *Basic point* 3000.
3. *Baiman*
Jika set memiliki 8 -10 *han*. *Basic point* 4000.
4. *Sanbaiman*
Jika set memiliki 11 atau 12 *han*. *Basic point* 6000.
5. *Kazoe-yakuman*
Jika set memiliki 13 atau lebih *han*. *Basic point* 8000.
6. *Yakuman*
Limit. *Basic point* 8000.
7. *Multiple Yakuman*
Jika set memiliki lebih dari satu *Limit*, *basic point* dikali dengan jumlah *Yakuman* yang terbentuk.

III. ALGORITMA GREEDY

Algoritma Greedy adalah algoritma pemecahan masalah yang membentuk solusi langkah per langkah, dan setiap langkahnya akan dipilih nilai maksimum lokal atau minimum lokal yang dapat dipilih. Algoritma ini merupakan metode yang paling populer untuk memecahkan persoalan optimasi.

Sesuai dengan namanya yang berarti rakus atau tamak, prinsip dasar dari greedy adalah “take what you can get now! ”. Dalam banyak kasus strategi greedy tidak dapat menghasilkan solusi yang optimal, tetapi greedy secara heuristic dapat menghasilkan nilai maksimum atau minimum lokal yang mendekati solusi global dengan waktu yang efisien.

A. Elemen-Elemen Algoritma Greedy

Algoritma greedy terdiri dari elemen-elemen berikut:

1. Himpunan kandidat.
Himpunan yang berisi elemen-elemen yang dapat menjadi himpunan pembentuk solusi.
2. Himpunan solusi.
Himpunan yang berisi kandidat-kandidat yang terpilih sebagai solusi persoalan.
3. Fungsi seleksi.
Fungsi yang digunakan untuk memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.
4. Fungsi kelayakan.
Fungsi yang memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala yang ada. Kandidat yang layak akan dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak akan dibuang dan tidak pernah dipertimbangkan lagi.
5. Fungsi objektif.
Fungsi yang memaksimalkan atau meminimumkan nilai solusi.

IV. IMPLEMENTASI

Pada permainan ini, terdapat dua cara untuk dapat memenangkan permainan, yaitu dengan memenangkan setiap putaran dengan cepat atau berusaha mengumpulkan tangan dengan nilai yang tinggi. Berikut akan diberikan dua strategi algoritma yang berbeda untuk memenangkan permainan.

A. Greedy Menang Cepat

Algoritma *Greedy Menang Cepat* merupakan algoritma yang bertujuan untuk memenangkan permainan dengan cepat tanpa mempedulikan nilai kartu yang berada di tangan. Dengan algoritma ini, poin pemain diharapkan tidak berkurang dan dapat tetap berada pada posisi tertinggi.

Pada algoritma *greedy* ini, pemain yang mendapat giliran harus mengumpulkan *tile* yang dapat membentuk sebuah *set* di tangannya. Dalam algoritma ini *tile* di tangan pemain akan dibagi ke dalam dua himpunan. Himpunan pertama adalah himpunan *tile* yang sudah membentuk sebuah *set*, dan himpunan kedua adalah himpunan *tile* yang tidak membentuk *set*. Pengelompokan ini bertujuan untuk memudahkan pengecekan *tile* pada setiap gilirannya.

Pada setiap gilirannya akan dilakukan pengecekan *tile* yang ditarik oleh pemain dan dibandingkan dengan *tile* dari himpunan yang tidak membentuk *set*. Jika *tile* yang

ditarik dapat membentuk sebuah *set*, pemain akan membuang salah satu *tile* dari himpunan kedua. Pada saat membuang *tile*, akan dilakukan pengecekan pada *discard pile* dan *tile* yang sudah banyak terbuang akan dibuang oleh pemain. Pada setiap giliran lawan akan dilakukan pengecekan yang sama terhadap *tile* yang dibuang oleh pemain lawan.

Algoritma ini memiliki kondisi yang mewajibkan pemain untuk mengumpulkan *set-set* tertentu sebagai prasyarat agar tangan yang dimiliki pemain bersifat *valid*. Syarat tersebut adalah pemain memiliki setidaknya satu *han* jika memiliki *set* yang terbuka. Dengan prasyarat tersebut, pemilihan *tile* yang dibuang akan menghindari *tile* naga, angin putaran, dan angin pemain.

Elemen-elemen dari algoritma greedy ini antara lain:

1. Himpunan kandidat (C)
Himpunan semua *tile* yang berada di tangan pemain.
2. Himpunan solusi (S)
Himpunan *tile* di tangan pemain yang telah membentuk sebuah *set*.
3. Fungsi seleksi
Fungsi yang melakukan pengecekan pada *tile* yang berada di tangan dan membuang *tile* yang mati atau sukar didapatkan.
4. Fungsi kelayakan
Fungsi ini memeriksa apakah *tile* yang ditarik atau terakhir dibuang oleh pemain lawan dapat membentuk sebuah *set*. Fungsi ini juga memeriksa apakah ada *set* yang hampir terbentuk di tangan.
5. Fungsi objektif
Fungsi objektif bertujuan agar pemain selalu memiliki *han* dan memutuskan apakah akan membuat *set* terbuka atau tidak.

Berikut adalah beberapa contoh dari penerapan algoritma di atas :



Gambar 11 Contoh 1 tile awal

Himpunan solusi yang terbentuk adalah sepasang tujuh karakter, sepasang tujuh lingkaran, dan sepasang satu bambu. Tarikan pertama adalah angin timur yang merupakan angin putaran. Fungsi kelayakan mendapatkan *tile* sembilan karakter, tiga lingkaran, lima lingkaran, dan sembilan lingkaran dapat membentuk sebuah *set*. Karena angin pemain adalah barat, maka fungsi seleksi akan membuang *tile* sembilan bambu atau naga hijau. Akan tetapi karena fungsi objektif bertujuan agar pemain mendapatkan *han*, maka *tile* yang akan dibuang adalah sembilan bambu.



Gambar 12 Contoh 2 tile awal

Himpunan solusi yang terbentuk adalah *chi* enam-delapan bambu dan sepasang delapan bambu yang merupakan tarikan pertama. Fungsi kelayakan mendapatkan *tile* satu dan tiga karakter, serta delapan dan sembilan karakter dapat membentuk sebuah *set*. Karena angin pemain adalah utara dan angin putaran adalah selatan, maka fungsi seleksi akan membuang *tile* dua bambu, angin timur, naga putih, atau naga merah. Akan tetapi karena fungsi objektif bertujuan agar pemain mendapatkan *han*, maka *tile* yang akan dibuang adalah dua bambu.

B. Greedy Menang Besar

Algoritma *Greedy Menang Besar* merupakan algoritma yang bertujuan untuk memenangkan permainan dengan mempertimbangkan nilai kartu yang berada di tangan. Dengan algoritma ini, pemain diharapkan mendapatkan poin yang tinggi pada setiap putarannya.

Pada algoritma *greedy* ini, pemain yang mendapat giliran harus mengumpulkan *tile* yang dapat membentuk sebuah *set* di tangannya dengan nilai yang terbesar. Dalam algoritma ini *tile* di tangan pemain akan diperiksa dan dilakukan pencocokan dengan *set* berdasarkan nilai tertinggi. Setelah pencocokan selesai, *tile* dibagi ke dalam dua himpunan. Himpunan pertama adalah himpunan *tile* yang dapat membentuk *set* yang telah dipilih, dan himpunan kedua adalah himpunan *tile* yang tidak dapat membentuk *set* terpilih. Pengelompokan ini bertujuan untuk memudahkan pengecekan *tile* pada setiap gilirannya.

Pada setiap gilirannya akan dilakukan pengecekan *tile* yang ditarik oleh pemain dan dibandingkan dengan *set* yang telah tersimpan. Jika *tile* yang ditarik dapat membentuk *set* tersebut, pemain akan membuang salah satu *tile* dari himpunan kedua. Pada saat membuang *tile*, akan dilakukan pengecekan pada *discard pile* dan *tile* yang sudah banyak terbuang akan dibuang oleh pemain. Pada setiap giliran lawan akan dilakukan pengecekan yang sama terhadap *tile* yang dibuang oleh pemain lawan. Pengecekan ini mempertimbangkan apakah *set* tersimpan memperbolehkan pemain memiliki *set* terbuka atau tidak.

Algoritma ini memiliki kondisi yang mengizinkan pemain untuk memiliki *set* terbuka atau tidak. Syarat tersebut adalah syarat-syarat yang dimiliki oleh setiap *set yaku*.

Elemen-elemen dari algoritma *greedy* ini antara lain:

1. Himpunan kandidat (C)
Himpunan semua *tile* yang berada di tangan pemain.
2. Himpunan solusi (S)
Himpunan *tile* di tangan pemain yang telah sesuai

dengan *set* solusi.

3. Fungsi seleksi
Fungsi yang melakukan pengecekan pada *tile* yang berada di tangan dan membuang *tile* yang tidak membentuk *set* solusi.
4. Fungsi kelayakan
Fungsi ini memeriksa apakah *tile* yang ditarik atau terakhir dibuang oleh pemain lawan dapat membentuk *set* solusi.
5. Fungsi objektif
Fungsi objektif melakukan pengecekan pada awal permainan dan memberikan sebuah *set* solusi yang harus dibentuk dan memutuskan apakah pemain akan membuat *set* terbuka atau tidak.

Berikut adalah beberapa contoh dari penerapan algoritma di atas :



Gambar 13 Contoh 3 tile awal

Dengan menggunakan contoh tangan awal pada penggunaan algoritma sebelumnya didapatkan hasil yang berbeda. Berdasarkan fungsi objektif, *set* dengan nilai tertinggi yang paling cepat dapat dibentuk adalah *All Crap* dengan tujuh *tile* yang sudah memenuhi solusi. Himpunan solusi yang terbentuk adalah satu karakter, sembilan karakter, angin timur, angin selatan, angin utara, naga putih dan naga merah. Fungsi seleksi akan membuang tiga karakter, delapan karakter, dua bambu, enam bambu, tujuh bambu, dan sepasang delapan bambu. *Tile* yang akan dibuang adalah *tile* yang berada paling kiri dan bukan merupakan himpunan solusi, yaitu tiga karakter.



Gambar 14 Contoh 4 tile awal

Dalam contoh ini pemain tidak mendapatkan giliran jalan pertama. Fungsi objektif melakukan pengecekan dan mendapatkan bahwa *set* tertinggi yang paling cepat dapat terbentuk adalah *Half Flush* dengan sepuluh *tile* yang memenuhi himpunan solusi. Pada saat pemain lawan membuang *tile*, fungsi kelayakan memeriksa apakah *tile* tersebut dapat membentuk himpunan solusi. Jika ya, fungsi objektif akan memutuskan apakah *tile* tersebut akan diambil atau tidak. Pada contoh ini *tile* angin barat yang dibuang oleh pemain lawan akan di-*pon* dan ditambahkan

