



## II. DASAR TEORI

### A. Algoritma Greedy

Algoritma Greedy adalah algoritma dimana pilihan terbaik pada tahap ini (optimum lokal) selalu diambil, dengan harapan akan menuju ke solusi optimal (optimum global). Contoh permasalahan greedy adalah persoalan penukaran uang. Jika diberikan nilai uang dan beberapa uang lainnya yang berjumlah lebih kecil, berapa jumlah uang minimal yang dapat digunakan untuk mendapatkan nilai yang sama dengan nilai uang yang diminta?

Contoh: Diminta uang sejumlah 32. Berapa jumlah uang minimum yang dapat dipakai bila tersedia uang 1, 5, 10, dan 25?

Penyelesaian: Dengan algoritma Greedy, ambil uang paling besar yang dapat diambil (lebih kecil dari uang yang dibutuhkan). Ulangi hingga uang yang diambil sama dengan uang yang diminta.

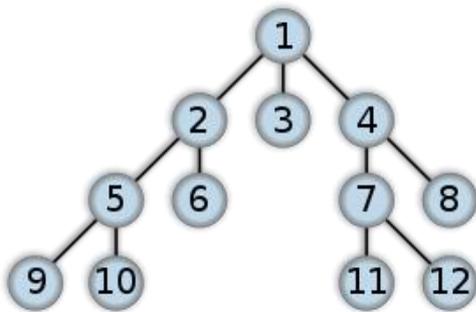
Solusi:  $25 + 5 + 1 + 1$ , ternyata solusi Greedy merupakan solusi optimal yaitu dibutuhkan 4 uang.

### B. Breadth First Search (BFS)

*Breadth First Search* atau BFS adalah algoritma dimana semua simpul yang telah diekspansi dari simpul *parent* akan diperiksa sebelum mengekspansi simpul dan menghasilkan simpul-simpul *child* baru. Algoritma BFS dilambangkan dengan penggunaan antrian/queue pada pemeriksaan simpul.

Tahapan algoritma BFS adalah sebagai berikut:

- Masukkan simpul akar ke antrian Q.
- Ambil simpul paling depan pada antrian. Jika simpul merupakan solusi, pencarian dihentikan.
- Jika simpul bukan merupakan solusi, masukkan simpul-simpul anak dari simpul tersebut kedalam antrian.
- Jika antrian kosong, berarti semua simpul sudah di cek dan tidak terdapat solusi.
- Ulangi dari langkah kedua.



Gambar 2.1 Urutan pencarian algoritma BFS

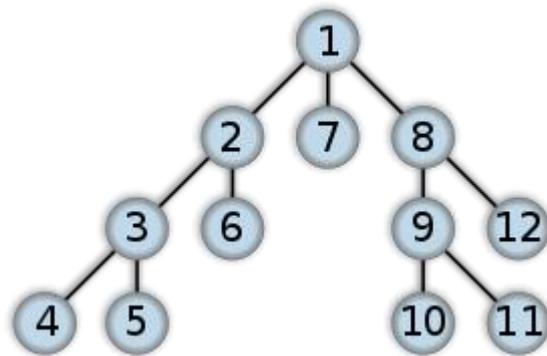
### C. Depth First Search

*Depth First Search* atau DFS adalah algoritma dimana setelah mengekspansi suatu simpul dan menghasilkan simpul-simpul *child* baru, algoritma akan mengekspansi simpul pertama dari himpunan simpul *child* dan menghasilkan simpul-simpul *child* baru. Aksi ini terus

dilakukan hingga solusi ditemukan maupun tidak. Bila solusi tidak ditemukan, algoritma akan menuju simpul *parent* dari simpul yang terakhir diperiksa dan mengekspansi simpul *child* berikutnya bila ada. Jika tidak, algoritma terus naik ke simpul *parent* selanjutnya. Algoritma DFS dilambangkan dengan penggunaan stack pada pemeriksaan simpul.

Tahapan algoritma DFS adalah sebagai berikut:

- Masukkan simpul akar ke stack S
- Ambil simpul teratas pada stack. Jika simpul merupakan solusi, maka pencarian dihentikan.
- Jika simpul bukan merupakan solusi, masukkan simpul-simpul anak dari simpul tersebut kedalam stack.
- Jika stack kosong, berarti semua simpul sudah di cek dan tidak terdapat solusi.
- Ulangi dari langkah kedua.



Gambar 2.2 Urutan pencarian algoritma DFS

### D. Mahjong Solitaire

Mahjong Solitaire adalah variasi dari permainan Mahjong. Menggunakan 144 keping yang sama dengan permainan Mahjong, terdapat keping-keping yang disusun secara tertentu dan pemain diharuskan untuk mengambil pasangan-pasangan keping yang sama untuk dihilangkan dari susunan tersebut. Permainan selesai saat semua keping pada susunan sudah diambil.

Kriteria pasangan keping yang dapat diambil adalah:

- Mempunyai gambar yang sama, kecuali:
  - Keping bergambar bunga dapat dipasangkan dengan sesama bunga yang lain (walaupun angkanya berbeda)
  - Yang sama juga berlaku pada keping bergambarkan musim (semi, panas, gugur, dingin)
- Keping dapat digeser ke kiri/kanan tanpa mengganggu keping lainnya

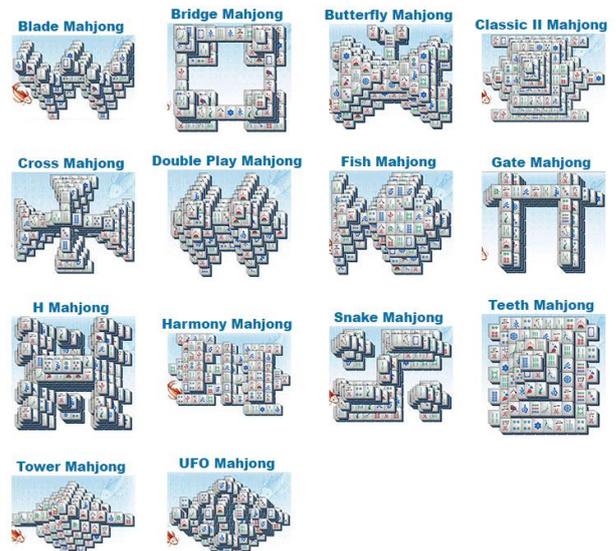
Yang dimaksud dengan keping lainnya adalah tidak boleh ada keping disebelah kiri atau kanan maupun diatas keping yang ingin diambil.

Bila pemain berhasil mengambil semua keping, maka permainan akan selesai. Waktu yang dibutuhkan oleh pemain akan ditunjukkan.



Gambar 2.3 Layar konfirmasi kemenangan pemain

Namun bila pemain memasuki *deadstate*, pemain dinyatakan gagal. Pemain dapat mengulang permainan dengan susunan keping yang berbeda atau mengacak keping-keping yang ada (*shuffle*) agar terdapat kemungkinan pasangan keping yang dapat diambil lagi.



Gambar 2.6 Variasi susunan keping Mahjong Solitaire

Jenis keping-keping pada permainan Mahjong maupun Mahjong Solitaire adalah sebagai berikut:



Gambar 2.4 Layar konfirmasi kekalahan pemain

Susunan keping-keping pada Mahjong Solitaire sangat beragam, dan yang paling terkenal adalah Turtle (Mahjong pada gambar 2.5).



Gambar 2.5 Variasi susunan keping Mahjong Solitaire

Bamboo Suit	Circle Suit	Character Suit
Honor Tiles	Flower Tiles	Season Tiles
Dragon Tile 1	Dragon Tile 2	Dragon Tile 3

### III. PEMBAHASAN

#### A. Penyelesaian dengan Algoritma Greedy

Tahapan penyelesaian permainan Mahjong Solitaire menggunakan algoritma greedy adalah sebagai berikut:

- Menghilangkan keping di tingkat tertinggi  $n$  dengan memasangkannya dengan keping lain yang sesuai.
- Bila keping yang sesuai tidak dapat diambil, hilangkan keping-keping lain disekitar keping yang sesuai dengan langkah paling sedikit.
- Bila keping yang sesuai tidak ada (tidak terlihat), ulangi langkah a pada tingkat  $n-1$ . Bila terdapat banyak kemungkinan, ambil pasangan yang tingkatnya paling tinggi.
- Ulangi langkah-langkah diatas hingga seluruh keping telah diambil.

Contoh penyelesaiannya adalah sebagai berikut.



Gambar 3.1 Susunan yang akan diselesaikan menggunakan algoritma Greedy

Tinjau keping pada tingkat paling atas, lalu cari apakah ada pasangan yang sesuai dengan keping tersebut.



Gambar 3.2 Pasangan keping optimum lokal (tahap 1)

Ternyata terdapat pasangan keping yang sesuai, jadi ambil pasangan keping tersebut lalu tinjau ulang pada tingkat paling tinggi (untuk kasus ini karena tingkat teratas hanya terdiri dari 1 keping, maka tingkat tertinggi adalah tingkat selanjutnya).



Gambar 3.3 Kemungkinan pasangan setelah pengambilan pasangan keping pertama

Dapat dilihat terdapat berbagai kemungkinan (warna hitam, merah dan biru). Ambil pasangan pada lingkaran berwarna hitam karena pasangan tersebut terletak lebih tinggi dari pasangan lainnya.



Gambar 3.4 Kemungkinan pasangan setelah pengambilan pasangan keping kedua

Karena tingkat kedua pasangan sama, pilihlah pasangan yang terletak lebih ke arah kiri-atas (ini adalah preferensi penulis, pada aplikasi sesungguhnya boleh dipilih pasangan dengan format tertentu asalkan konsisten).



Gambar 3.5 Pasangan keping optimum lokal (tahap 3)



Gambar 3.6 Kemungkinan pasangan pada tingkat n+1

Karena keping pada tingkat tertinggi tidak terlihat pasangannya, maka pemeriksaan turun ke tingkat selanjutnya. Terlihat bahwa kemungkinan pasangan yang ada menjadi banyak karena pada tingkat sekarang keping yang ada juga lebih banyak. Ulangi langkah-langkah diatas hingga semua pasangan pada papan telah diambil.

### B. Penyelesaian dengan Algoritma BFS

Tahapan penyelesaian dengan algoritma BFS adalah sebagai berikut:

- Mencatat semua pasangan keping yang dapat dihilangkan pada kedalaman  $n$ .
- Menghilangkan pasangan keping pertama pada kedalaman  $n$
- Ulangi langkah b hingga semua pasangan keping pada kedalaman  $n$  telah diambil
- Ulangi langkah a pada kedalaman  $n+1$

Contoh penyelesaiannya adalah sebagai berikut.



Gambar 3.7 Kemungkinan pasangan keping pada penyelesaian dengan algoritma BFS

Ambil pasangan keping dari yang cenderung terletak di kiri-atas (ini adalah preferensi penulis, pada aplikasi sesungguhnya boleh dipilih pasangan dengan format tertentu asalkan konsisten). Untuk kasus ini penulis memilih pasangan pada lingkaran pink.



Gambar 3.8 Kondisi susunan keping setelah pasangan keping yang ditandai lingkaran pink diambil

Teruskan mengambil semua pasangan keping yang telah diketahui pada gambar 3.7 .



Gambar 3.9 Kemungkinan pasangan keping pada kedalaman  $n+1$  pohon BFS

Ulangi langkah-langkah diatas hingga semua pasangan keping telah diambil.

### C. Penyelesaian dengan Algoritma DFS

Tahapan penyelesaian dengan algoritma DFS adalah sebagai berikut:

- Mencatat semua pasangan keping yang dapat dihilangkan pada kedalaman  $n$ .
- Menghilangkan pasangan keping pertama pada kedalaman  $n$
- Meninjau apakah terdapat pasangan keping yang terbuka karena pengambilan pasangan keping sebelumnya
- Bila ada, catat semua pasangan keping yang dapat dihilangkan dan ulangi langkah b pada kedalaman  $n+1$ .
- Bila tidak ada, lanjutkan ke pasangan keping selanjutnya (kedalaman  $n$ ).

Contoh penyelesaiannya adalah sebagai berikut.



Gambar 3.7 Kemungkinan pasangan keping pada penyelesaian dengan algoritma DFS

Ambil pasangan keping dari yang cenderung terletak di kiri-atas (ini adalah preferensi penulis, pada aplikasi sesungguhnya boleh dipilih pasangan dengan format tertentu asalkan konsisten). Untuk kasus ini penulis memilih pasangan pada lingkaran hitam.



Gambar 3.8 Kemungkinan pasangan keping yang bersinggungan dengan pasangan keping yang baru saja diambil

Periksa apakah ada pasangan keping yang dapat diambil (“terbuka”) dikarenakan pengambilan pasangan keping sebelumnya. Ternyata ada pasangan keping yang dapat diambil ditandakan dengan lingkaran kuning.



Gambar 3.9 Tidak terdapat pasangan keping yang dapat diambil lagi, simpul dimatikan dan berpindah ke pasangan keping lainnya

Terlihat bahwa tidak ada pasangan keping yang bisa diambil lagi (ditandakan dengan lingkaran hitam), oleh karena itu simpul ini pada pohon DFS dimatikan dan pemeriksaan berpindah ke pasangan keping lainnya.

#### IV. KESIMPULAN

Permainan Mahjong Solitaire dapat diselesaikan dengan algoritma Greedy dan BFS. Penggunaan algoritma tersebut dapat mengurangi terjadinya *deadstate* dan membantu pemain dalam menyelesaikan permainan.

#### REFERENSI

- [1] Munir, Rinaldi, Strategi Algoritma. Bandung : Penerbit Informatika, Palasari
- [2] <http://www.247mahjong.com/> Diakses pada 07 Mei 2016, Pukul 13.40
- [3] <http://home.halden.net/vkp/vkp/history.html> Diakses pada 07 Mei 2016, Pukul 14.02
- [4] <http://www.ics.uci.edu/~epstein/cgt/hard.html> Diakses pada 07 Mei 2016 Pukul 14.25

- [5] <http://www.math.ru.nl/~debondt/mjsolver.html> Diakses pada 07 Mei 2016 Pukul 14.53

#### 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, 08 Mei 2016

Resa Kemal Saharso 13514109