

Penggunaan Algoritma *Brute Force*, DFS, BFS dan Perbandingannya dalam Permainan *Flow*

Mohammad Anugrah Sulaeman / 13510008

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

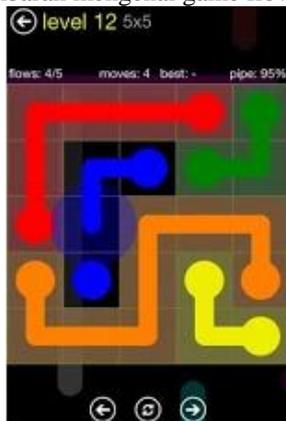
13510008@std.stei.itb.ac.id

Abstract—Flow adalah permainan game android untuk menyambungkan garis diantara titik yang berwarna. Permainan akan berakhir jika pemain dapat menyambungkan semua titik berwarna tanpa adanya garis yang bersinggungan. Permainan ini dapat diselesaikan dengan algoritma brute force, bfs, dan dfs.

Index Terms—Flow, brute force, bfs, dfs.

I. PENDAHULUAN

Flow adalah sebuah game android keluaran *Big Duck Games LLC* yang sedang populer belakangan ini. Permainan flow adalah permainan board sederhana yang memiliki tujuan untuk menghubungkan semua titik yang berwarna dengan garis vertikal dan horizontal sehingga semua garis saling berhubungan. Permainan akan selesai jika semua titik berhasil dihubungkan tanpa adanya garis yang saling berpotongan. Gambar di bawah ini dapat memberikan gambaran mengenai game flow.



Gambar 1 screenshot game flow

Pada makalah ini, untuk mempermudah pembahasan maka hanya akan digunakan game flow dengan ukuran 5x5 dan memiliki 5 warna titik yang berbeda.

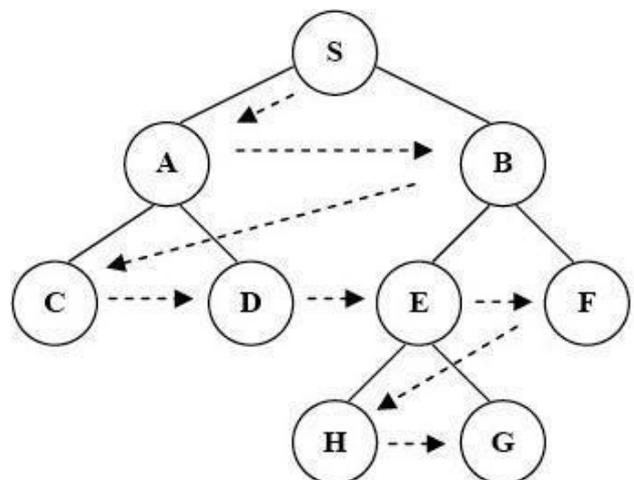
II. DASAR TEORI

A. Algoritma Brute Force

Algoritma *brute force* adalah algoritma yang dilakukan dengan pendekatan yang lempang (*straightforward*) untuk memecahkan suatu masalah. Algoritma *brute force* memecahkan masalah dengan sangat sederhana, langsung, dan jelas (*obvious way*).

B. Algoritma BFS

Algoritma Breadth-First Search (BFS) atau pencarian melebar merupakan salah satu metoda traversal di dalam graf yang mengunjungi simpul-simpul di dalam graf secara sistematis. Pada metode Breadth-First Search, semua node pada tingkat n akan dikunjungi terlebih dahulu sebelum mengunjungi node-node pada level $n+1$. Pencarian dimulai dari node akar terus ke level ke-1 dari kiri ke kanan, kemudian berpindah ke level berikutnya demikian pula dari kiri ke kanan sampai solusi persoalan yang direpresentasikan oleh graf tersebut ditemukan. Keuntungan BFS adalah tidak akan menemui jalan buntu. Namun, BFS membutuhkan banyak memori karena menyimpan semua node dalam satu pohon.



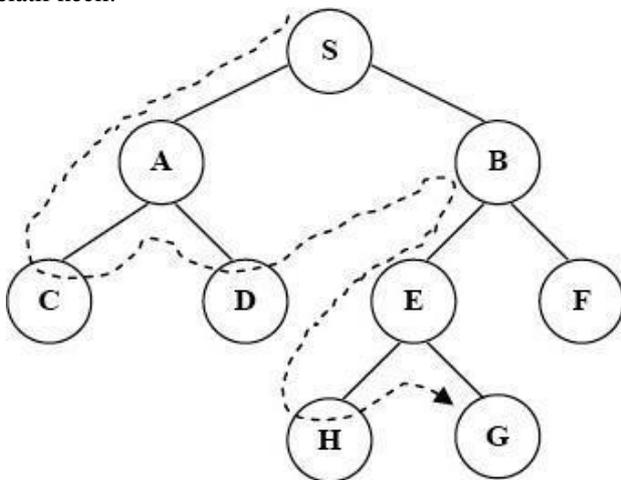
Gambar 2 langkah-langkah pengunjungan tree pada breadth-first search

Algoritma BFS:

1. Kunjungi simpul n,
2. Kunjungi semua simpul yang bertetangga dengan simpul n terlebih dahulu.
3. Kunjungi simpul yang belum dikunjungi dan bertetangga dengan simpul-simpul yang tadi dikunjungi, demikian seterusnya sampai solusi ditemukan.

B. Algoritma DFS

Algoritma Depth-first search (DFS) atau pencarian mendalam merupakan salah satu metoda traversal di dalam graf yang mengunjungi simpul-simpul di dalam graf secara sistematis. Pada depth-first search, proses pencarian akan dilakukan pada anaknya sebelum pencarian ke tetangga yang setingkat dengan simpul tersebut dilakukan. Pencarian dimulai dari node akar ke level yang lebih tinggi. Proses ini diulangi terus hingga ditemukan solusi. DFS hanya akan mendapatkan 1 solusi pada setiap pencarian, namun membutuhkan memori yang relatif kecil.



Gambar 3 langkah-langkah pengunjungan tree pada depth-first search

Algoritma DFS:

1. Kunjungi simpul n,
2. Kunjungi simpul w yang bertetangga dengan simpul n.
3. Ulangi DFS mulai dari simpul w.

III. METODE PENCARIAN SOLUSI GAME

A. Menggunakan Algoritma Brute Force

Algoritma brute force dapat digunakan untuk menyelesaikan game *flow* ini. Langkah pencarian solusi dilakukan dengan metode khas brute force, yaitu dengan mencoba seluruh kemungkinan susunan warna pada board. Algoritma ini memiliki jumlah maksimum pencarian sebesar 25!.

Pada Algoritma ini program memberikan satu warna pada satu grid, pemberian warna dilakukan terus menerus sampai semua grid terisi. Proses pemberian warna ini diulang terus hingga semua kemungkinan sudah dilakukan. Hasil dari seluruh kemungkinan ini kemudian diberikan pada suatu fungsi `isSolusi()` yang mengecek

apakah susunan warna pada board sudah memenuhi solusi atau belum. Jika ada susunan warna yang memenuhi solusi, maka algoritma ini akan mengembalikan susunan board tersebut.

Algoritma ini membutuhkan waktu yang cukup lama tetapi menjamin bahwa solusi akan ditemukan. Jika ukuran board bertambah besar, misalnya menjadi 9x9, maka penggunaan algoritma brute force sangatlah tidak efisien.

A. Menggunakan Algoritma Breadth-First Search

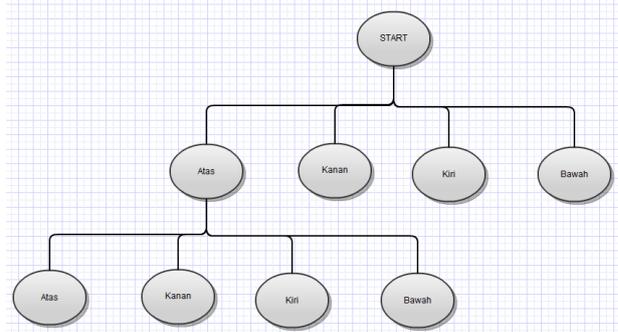
Algoritma Breadth-First Search juga dapat dilakukan untuk mencari solusi dari game *flow*. Berbeda dengan algoritma brute force yang mengisi warna sembarang mulai dari kotak kosong pertama, algoritma BFS mulai mengisi warna mulai dari tetangga sebuah titik yang sudah diketahui letaknya.



Gambar 4 State pada permainan game flow

Misalkan pada suatu state game seperti gambar di atas, untuk menghubungkan titik kuning, maka algoritma BFS akan mulai mengisi warna dari tetangga kuning yang belum terisi warna lain. Algoritma akan bercabang ke arah atas, kiri, dan bawah kuning. Kemudian dari setiap cabangnya akan bercabang lagi ke tetangganya. Proses ini dilakukan recursive sampai ditemukan path untuk kuning. Proses yang sama dilakukan untuk titik berwarna biru muda.

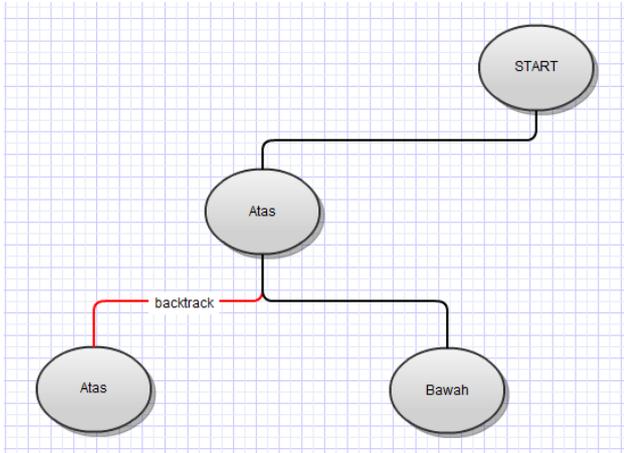
Penggambaran tree untuk algoritma BFS kurang lebih diberikan di bawah ini.



Gambar 5 Langkah penyelesaian dengan BFS

A. Menggunakan Algoritma DFS

Penggunaan algoritma DFS pada umumnya tidak jauh berbeda dengan BFS. Proses yang membedakannya hanyalah pada saat mengisi titik, algoritma DFS akan berjalan terus ke kedalaman berikutnya tanpa memproses tetangga yang lain terlebih dahulu. Perbedaan proses dengan BFS terlihat pada gambar di bawah ini.



Gambar 6 Langkah penyelesaian dengan DFS

IV. PERBANDINGAN ALGORITMA BRUTE FORCE, BFS, DAN DFS

Algoritma brute force adalah algoritma yang paling buruk untuk mencari solusi permainan game flow. Algoritma ini membuang banyak sekali memory dan waktu karena harus men-generate seluruh kemungkinan warna pada board. Algoritma ini memiliki kompleksitas mencapai $O(n!)$.

Algoritma BFS merupakan algoritma perbaikan dari brute force. Algoritma ini mulai pencarian dari titik yang sudah diketahui warnanya. Algoritma ini sangat baik untuk mencari solusi. Memiliki kompleksitas sebesar $O(\text{jumlah warna titik}^n)$.

Algoritma DFS tidak jauh berbeda dengan algoritma BFS. Algoritma ini tidak membutuhkan state sebanyak yang dibutuhkan oleh algoritma DFS. Memiliki kompleksitas waktu yang sama dengan algoritma BFS. Algoritma ini baik untuk board berukuran kecil, tetapi jauh lebih buruk dari algoritma BFS jika ukuran board bertambah besar. Hal ini dikarenakan DFS melakukan pencarian kedalam, padahal hal ini buruk jika ukuran board besar.

Dari ketiga algoritma di atas, setiap algoritma menjamin bahwa solusi akan ditemukan. Pemilihan algoritma yang akan digunakan lebih baik menggunakan algoritma BFS. Algoritma BFS berjalan baik dan cepat dan tidak memiliki perbedaan waktu yang besar untuk menyelesaikan permainan dengan ukuran board yang besar.

V. KESIMPULAN

Permainan game flow dapat diselesaikan dengan tiga

algoritma yang berbeda, yaitu algoritma brute force, algoritma BFS, dan algoritma DFS.

Algoritma brute force dilakukan dengan pendekatan yang lempang, yaitu dengan mengisi titik dari kotak kosong pertama sampai terakhir, dan mencari semua kemungkinan susunan board yang ada, kemudian memilih solusi dari susunan tersebut. Algoritma ini adalah algoritma yang paling lambat dari ketiga algoritma yang digunakan, dan memiliki kompleksitas waktu $O(n!)$.

Algoritma BFS dilakukan dengan pendekatan mencari tetangga dari sebuah titik awal warna yang memang sudah diketahui. Proses pengisian warna tetangga ini kemudian dilakukan secara recursive sampai semua titik warna tersambung. Algoritma ini merupakan algoritma yang paling baik dan memiliki kompleksitas $O(\text{jumlah warna titik}^n)$.

Algoritma DFS pada dasarnya tidak jauh berbeda dengan algoritma BFS. Pencarian dilakukan ke tetangga yang berarah sama sampai proses backtrack dilakukan. Kompleksitas algoritma sama seperti BFS tetapi pada implementasinya akan menghasilkan waktu eksekusi yang lebih buruk dari BFS jika ukuran board besar, misal 9×9 .

REFERENCES

- [1] <https://play.google.com/store/apps/details?id=com.bigduckgames.flow>, tanggal terakhir akses : 20 Desember 2012
- [2] Diktat Kuliah Strategi Algoritma 2012, Rinaldi Munir.
- [3] Slide kuliah 2012

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, 20 Desember 2012

Mohammad Anugrah Sulaeman
13510008