

Penerapan Algoritma Branch & Bound dan Backtracking pada Game Flow

Rio Dwi Putra Perkasa (13515012)

Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung (ITB)
Bandung, Indonesia
13515012@std.stei.itb.ac.id
dwiputra.rio@gmail.com

Abstraksi – Algoritma Branch and Bound merupakan sebuah algoritma yang membentuk pohon ruang status secara BFS untuk mencari solusi. Untuk membedakan dengan BFS itu sendiri, pada algoritma B&B ini setiap cabang pohon diberi nilai(cost) yang digunakan untuk menentukan percabangan selanjutnya. Algoritma Backtracking merupakan sebuah algoritma yang akan kembali mengecek percabangan sebelumnya jika cabang yang dilewati tidak menemukan solusinya. Game Flow sendiri merupakan sebuah game bergenre puzzle yang misi utamanya adalah menggabungkan titik dengan membentuk sebuah jalur yang akan mengisi semua tempat yang tersedia.

Keywords – B&B, pohon, backtracking, game, flow

I. PENDAHULUAN

Game bergenre puzzle merupakan sebuah game yang sengaja dibuat untuk diselesaikan penggunaannya dengan sedikit pemikiran dan logika. Secara umum game puzzle dibawakan secara simple dan tidak terlalu berat juga ukurannya yang relative kecil jika dibandingkan game bergenre lainnya.

Game puzzle biasanya hanya mempunyai satu tujuan khusus untuk puzzle tersebut, hal ini berbeda dengan game bergenre lain yang mempunyai berapa tujuan. Tapi game seperti ini termasuk banyak peminatnya walau dibuat dengan simple.

Salah satu game yang akan saya bahas pada tulisan saya ini merupakan game bergenre puzzle yang berjudul Flow. Game ini merupakan game yang diproduksi oleh Big Duck Games LLC yang dapat diinstall di platform Android dan iOS. Game ini didesain dengan simple dan menarik sehingga game ini bisa dikatakan populer dengan jumlah download yang melebihi 100 juta download di Play Store.

Game Flow ini menantang pemainnya untuk menggabungkan dua titik dengan warna yang sama dengan cara menggeser salah satu titik membentuk sebuah jalur menuju titik lainnya. Jika misinya hanya menggabungkan titik dengan membuat jalan, pastinya akan mudah oleh karena itu di dalam game ini titik yang harus dipasangkan tidak hanya dua tetapi ada beberapa dwengian warna yang berbeda.

Dalam game Flow jalur dari titik yang berbeda warna tidak boleh untuk saling memotong. Game Flow ini juga terdapat ukuran tile atau tempat permainan, dalam game ini disediakan

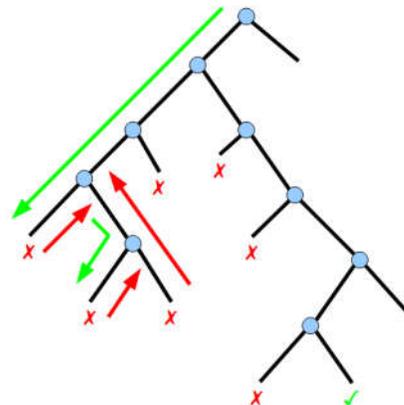
berbagai variasi ukuran tile yang tentunya dengan tingkat kesulitan masing-masing.

Apabila pemain ingin memenangkan setiap level maka pemain diwajibkan untuk menggabungkan semua titik dan mengisi semua tile dengan jalur yang di buat oleh titik tersebut.

II. DASAR TEORI

A. Algoritma Backtracking

Algoritma Runtut-balik atau biasa disebut dengan backtracking merupakan sebuah algoritma yang berbasis pada pencarian secara mendalam atau DFS. Pada algoritma backtracking hanya mempertimbangkan pencarian yang mengarah ke solusi saja. Sehingga algoritma ini juga bisa disebut perbaikan dari algoritma brute force, sehingga secara otomatis waktu yang dibutuhkan untuk menyelesaikan permasalahannya juga lebih cepat.



Gambar 1 Algoritma Runtut-balik atau Backtracking
Sumber : <https://www.w3.org/2011/Talks/01-14-steven-phenotype/backtracking.png>

Diakses pada 18 Mei 2017 pukul 21:32

Algoritma backtracking pertama kali dikenalkan oleh D.H. Lehmer pada tahun 1950. Elanjutnya , R.J Walker, Golomb,

dan Baumert menyajikan uraian umum tentang runut-balik dan penerapannya pada berbagai persoalan [HOR78]. Algoritma backtracking lebih banyak penerapan pada permainan seperti *N Queen Problem*, *Knight Tour Problem*, *Rat in a Maze Puzzle*, dan *Word Matrix*.

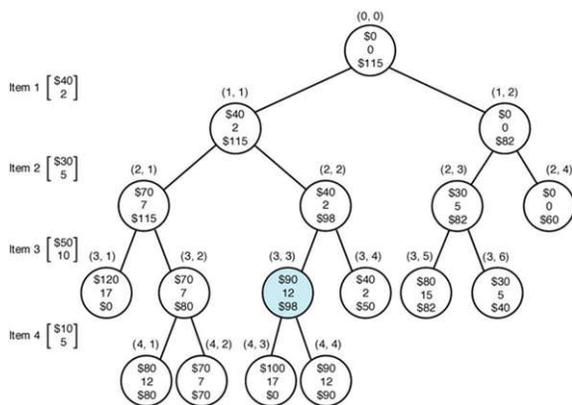
Seperti namanya, algoritma ini akan merunut balik pada percabangan terakhir apabila ekspansinya gagal. Setelah kembali ke simpul terakhir maka algoritma akan mencari di cabang lain.

Algoritma umum untuk backtracking adalah sebagai berikut

```
pilih titik mulai
while(solusi belum ditemukan)
  for setiap jalan dari titik mulai
    if jalan yang dipilih benar, maka
      dipilih
      buat rekursif untuk semua problem
      yang tersisa
    if rekursif mengembalikan true maka
      true
    else batalkan jalan yang dipilih dan
      kembalikan false.
  end for
  if tidak ada jalan yang benar,
    kembalikan false, NO SOLUTION
```

B. Algoritma Branch and Bound

Algoritma Branch and Bound atau biasa ditulis B&B merupakan sebuah metode pencarian di dalam ruang solusi secara sistematis, sehingga dalam pencarian solusinya B&B membentuk pohon ruang status. Pada B&B berbeda dengan backtracking pembentukan pohon ruang status dibangun dengan skema BFS. Untuk mempercepat pencarian solusi maka di setiap simpul yang diekspansi di beri nilai atau *cost* nilai inilah yang nantinya akan digunakan untuk menentukan simpul mana yang akan di ekspansi selanjutnya.



Gambar 2 Pohon Ruang Status dari algoritma B&B dalam penyelesaian *Knapsack Problem*

Sumber : <https://qph.ec.quoracdn.net/main-qimg-9dddb8feab89ee865658425f5383f9e5>

Diakses pada 18 Mei 2017 pukul 22:24

Algoritma B&B pertama kali dikenalkan oleh A.H. Land dan A.G Doig pada 1960 untuk pemrograman diskrit, dan menjadi algoritma yang sering digunakan untuk menyelesaikan masalah optimasi NP-Hard.

Simpul yang dibangkitkan pada algoritma B&B adalah simpul yang memiliki nilai terkecil diantara simpul hidup lainnya. Ongkos pada setiap simpul merupakan nilai taksiran yang didapatkan dari fungsi pembatas yang dibuat. Algoritma yang mirip dengan B&B adalah algoritma A*.

Prinsip pencarian algoritma B&B hamper mirip dengan BFS hanya saja ekspansi yang dilakukan tidak berdasarkan urutan pembangkitan tetapi dilihat dari nilai yang ada pada setiap simpul..

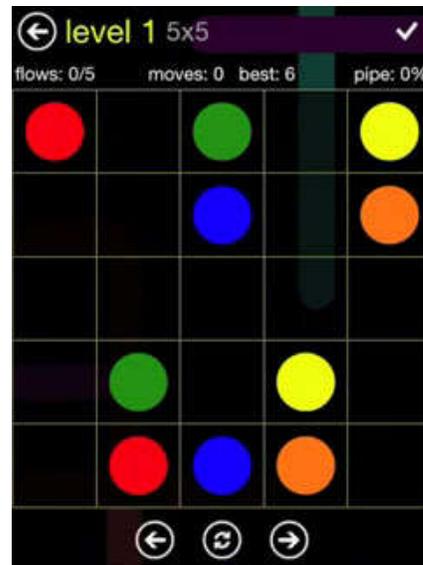
Untuk setiap simpul X, nilai batas ini dapat berupa :

- a. Jumlah simpul dalam subpohon X yang perlu dibangkitkan sebelum simpul solusi ditemukan.
- b. Panjang lintasan dari simpul X ke simpul solusi terdekat.

C. Permainan Flow

Permainan yang berjudul Flow ini merupakan salah satu permainan yang diproduksi oleh *Big Duck Games LLC*. Permainan Flow tersedia gratis di play store untuk android dan iOS. Permainan ini memiliki *gameplay* yang mudah dan desain yang minimalis. Permainan ini membutuhkan pilihan langkah untuk menyelesaikan misinya.

Dalam permainan ini disediakan kotak persegi untuk tempat bermain berukuran n x n (membesar sesuai level). Di dalam kotak terdapat beberapa pasang titik dengan warna yang berbeda tiap pasang. Setiap pasang dipisah dengan jarak tertentu.



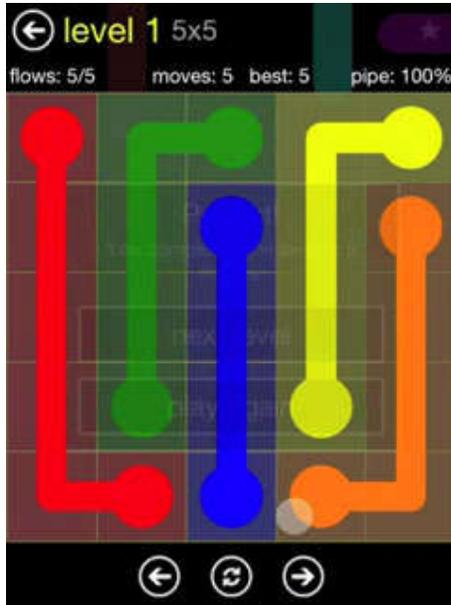
Gambar 3 Tampilan Permainan Flow, sebelum titik dipasang

Sumber :

<http://www.gamerevolution.com/images/misc/level51start.jpg>

Diakses pada 18 Mei 2017 pukul 23:08

Misi dari permainan ini adalah menyatukan seluruh titik dengan warna yang sama pada kotak yang telah disediakan. Jalur yang tercipta dari dua pasang titik yang berwarna sama tidak boleh berpotongan dengan jalur dari pasangan titik yang lain. Hal yang menantang pada permainan ini adalah menemukan jalan yang membuat semua jalur yang dibuat tidak berpotongan.



Gambar 4 Semua titik pada permainan flow sudah di pasangkan
Sumber :

<http://www.gamerevolution.com/images/misc/level51.jpg>
Diakses pada 18 Mei 2017 pukul 23:14

Dari gambar diatas dapat dilihat bahwa terdapat dua parameter keberhasilan dari permainan ini yaitu flow yang diciptakan dan grid yang telah disediakan harus diisi penuh. Apabila pemain dapat memenuhi kedua parameter itu maka pemain bisa dikatakan sudah menyelesaikan permainan tersebut dan akan berganti level ke level yang lebih sulit.

Selain kedua parameter tersebut, ketinggian nilai yang didapat juga dipengaruhi oleh jumlah langkah yang dilakukan untuk menyelesaikan level tersebut. Semakin sedikit langkah yang dilakukan, maka semakin tinggi score yang didapat.

III. PENERAPAN ALGORITMA BACKTRACKING DAN B&B PADA PERMAINAN FLOW

Seperti yang dijelaskan sebelumnya, bahwa permainan ini diperlukan langkah yang tepat untuk bisa menyelesaikan permainan dan memperoleh nilai yang besar dengan langkah yang seminimal mungkin. Juga tanpa memotong jalur lawan.

Persoalan ini akan diselesaikan dengan algoritma backtracking dan B&B dimana akan ditentukan dulu nilai taksiran dengan fungsi yang nanti akan dijelaskan.

Sebelum melangkah ke pembahasan algoritma, saya akan menjelaskan terlebih dahulu *prerequisite* yang harus dipenuhi yaitu, program sudah mengetahui urutan pasangan titik yang memiliki jarak terpendek ke yang terjauh. Syarat ini diperlukan untuk urutan pengaplikasian algoritma untuk setiap pasang titik. Jadi pasangan titik terjauh akan mendapat giliran terakhir.

Untuk mencari titik terdekat bisa menggunakan algoritma lain seperti Brute Force, Divide and Conquer dan yang lainnya. Saya tidak akan membahas algoritma untuk mencari titik terdekat.

Untuk selanjutnya saya akan menentukan pattern yang akan saya gunakan untuk menganalisis adalah sebagai berikut

Red	Blue		Blue	
			Green	
Green			Yellow	
Yellow		Red		

Gambar 5 pattern

Dari tabel disamping dapat diketahui bahwa warna biru merupakan pasangan titik yang terdekat, kemudian dilanjutkan dengan hijau dan kuning, lalu merah

A. Penerapan dengan Algoritma Backtracking

Penerapan dengan Algoritma Backtracking dilakukan terlebih dahulu pada warna biru. Penentuan penarikan garis ditentukan yaitu berawal dari titik yang memiliki absis dan ordinat terbesar. Contoh pada pattern diatas maka biru yang kanan akan menjadi awal garis.

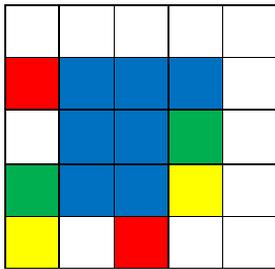
Prioritas pembangkitan simpul selanjutnya adalah sebagai berikut

	3	
2	Red	4
	1	

Gambar 6 urutan prioritas

Dapat diketahui dari ilustrasi disamping bahwa prioritas menurun searah jarum jam. Apabila terdapat titik lain di jalan, maka jalan tersebut dimatikan.

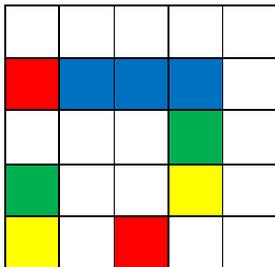
Pertama-tama titik biru akan menuju bagian kiri. Saat sudah melangkah satu kotak, maka langkah selanjutnya adalah ke bawah sampai tiga kali hingga akhirnya bertemu dengan titik merah dan penampakkannya akan menjadi seperti berikut sampai titik biru terhubung.



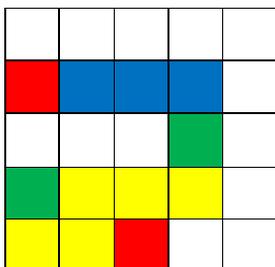
Sampai disini maka akan dapat diprediksi bahwa merah akan bergabung lewat jalan paling luar.

Gambar 7 titik biru sudah tersambung dengan algoritma Backtracking

Tetapi terdapat kendala, yaitu warna kuning dan hijau tidak bisa menemukan jalan untuk bergabung, sehingga hal ini di perlukan backtracking untuk titik yang sudah digabungkan dipisah kembali, kemudian untuk setiap titik mencoba jalan yang lain.

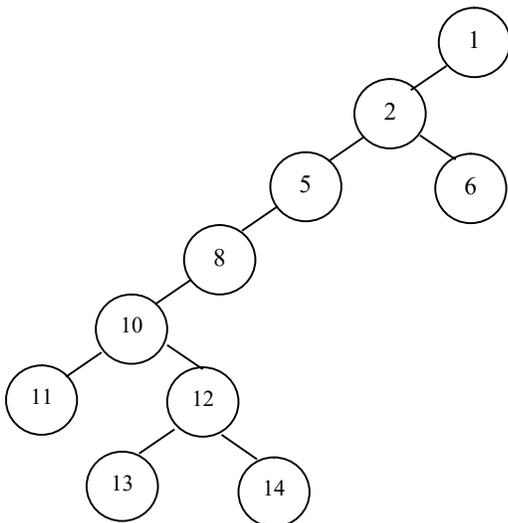


Gambar 8 Solusi penggabungan titik Biru



Gambar 9 Titik kuning disatukan

Hal ini dilakukan secara berulang sampai semua bisa bergabung. Pohon Status yang tercipta untuk pemasangan titik biru adalah



Gambar 10 pohon pembangkitan hasil algoritma backtracking

Karena solusi pada simpul 14 tidak bisa menyelesaikan permainan, jadi titik biru melakukan backtracking sampai menemukan solusi kembali. Dan Akhirnya menemukan pada simpul ke enam

B. Penerapan dengan Algoritma Branch and Bound

Pada pengaplikasian Branch and Bound pohon ruang status yang terbentuk menggunakan skema BFS, jadi disini untuk menyimpan simpul hidup dibutuhkan sebuah queue.

Penarikan garis berlaku sama seperti backtracking yaitu titik mulai adalah titik yang mempunyai jarak atau jumlah kotak yang paling banyak, apabila sama, maka dipilih yang memiliki Prioritas yang berlaku untuk pembangkitan simpul baru pada penerapan algoritma ini sama dengan yang digunakan pada backtracking yaitu searah jarum jam.

Selaian itu algoritma B&B juga memerlukan sebuah fungsi pembatas (bound) dalam kasus ini adalah

$$c(i) = f(i) + g(i)$$

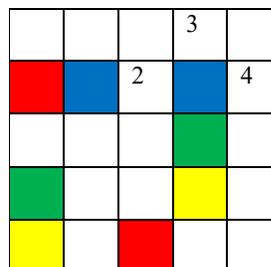
dimana $c(i)$ adalah nilai yang akan diberikan pada setiap simpul yang dibangkitkan.

$f(i)$ adalah jumlah kotak yang telah dilewati dari akar.

$g(i)$ adalah jumlah kotak paling sedikit dari tempat simpul sampai simpul solusi, tetapi tidak melewati tempat yang ada titiknya

Sama seperti contoh pada algoritma backtracking disini saya akan memberikan contoh penerapan algoritma pada permainan flow. Saat pertama kali permainan dimulai maka pada list antrian akan terdapat node start kemudian titik biru di sebelah kanan akan mengekspansi menjadi titik-titik yang ada di sekitarnya.

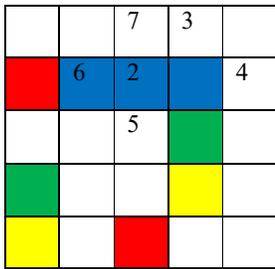
Simpul	C(2)	C(3)	C(4)
1	2	3	5



Gambar 11 pembangkitan simpul dengan algoritma B&B

Dengan ilustrasi diatas maka simpul ke-3 yang dibangkitkan akan diekspansi lagi

Simpul	C(5)	C(6)	C(7)
2	4	2	4



Gambar 12 titik biru digabungkan dengan B&B

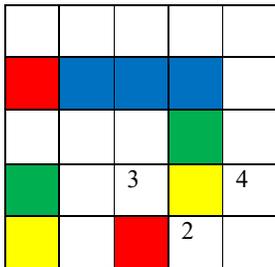
Apabila diteruskan maka simpul 6 akan diekspansi tetapi simpul 6 adalah solusi untuk pasangan titik biru.

Apabila menggunakan algoritma ini jalur yang dipilih relative pendek disbanding dengan algoritma backtracking.

Pada suatu kasus pasti terdapat keadaan dimana jalur yang dibuat menggunakan algoritma ini menghalangi jalur titik lain, jika hal ini terjadi maka diperlukan backtracking. Backtracking yang dilakukan sama dengan algoritma sebelumnya, yaitu menggagalkan langkah sebelumnya.

Apabila ditinjau dari titik yang kuning maka titik kuning yang sebelah kanan akan menjadi acuan yang langkah awal daftar antrian akan sebagai berikut

Simpul	C(2)	C(3)	C(4)
1	0	4	0



Gambar 13 pembangkitan simpul pada titik kuning

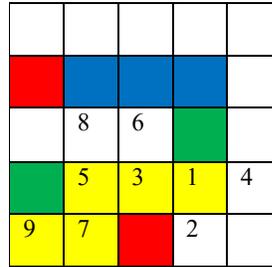
Dari ilustrasi diatas hanya simpul 3 yang bisa di ekspansi, karena simpul yang lain tidak ada jalur yang bisa menuju simpul solusi. Sehingga apabila diteruskan maka antrian akan menjadi

Simpul	C(5)	C(6)
3	4	6

Simpul	C(7)	C(8)
5	4	6

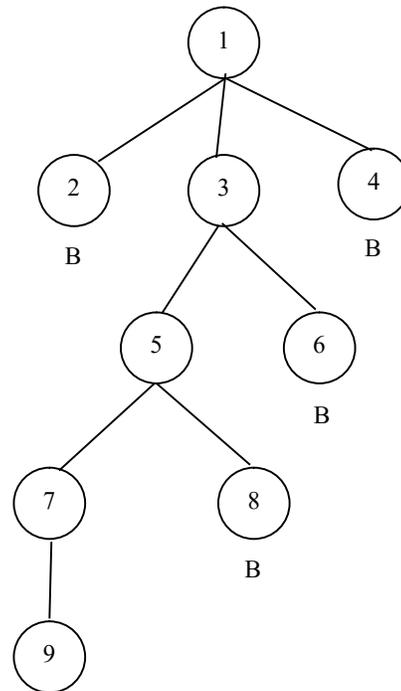
Simpul	C(9)
7	4

Simpul	C(Solusi)
9	Solusi



Gambar 14 Simpul kuning disatukan dan semua simpul

Pohon status yang dihasilkan untuk titik yang kuning adalah sebagai berikut



Gambar 15 Pohon pembangkitan dari algoritma B&B

IV. KESIMPULAN

Algoritma Backtracking dan Branch&Bound dapat diterapkan di berbagai kasus, terutama kasus penentuan rute, langkah, dan jalan yang mangkus dan murah. Sedangkan untuk algoritma Backtracking lebih tepat pada penerapan pencarian jalan keluar atau labirin.

Penerapan Algoritma backtracking dan B&B pada permainan Flow cukup mangkus karena bisa menyelesaikan masalah dengan lebih cepat. Terutama pada algoritma B&B dengan menggunakan fungsi pembatas

$$c(i) = f(i) + g(i)$$

dimana $c(i)$ merupakan nilai pembatas yang didapatkan dari jumlah kotak dari simpul ke simpul akar ($f(i)$) dan jumlah kotak dari simpul ke simpul solusi ($g(i)$).

Kelemahan pada algoritma backtracking adalah jika simpul solusi terletak selalu di akhir dari simpul ekspansi yang juga terakhir kali diekspansi. Sedangkan untuk algoritma B&B mempunyai kelemahan yaitu pada permainan Flow akan menutup jalur lawan, karena yang dipilih selalu jalur terpendek.

UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Allah, karena dengan karunia-Nya, saya dapat menyelesaikan makalah ini dengan baik. Terima kasih saya ucapkan kepada Ibu Nur Ulfa Maulidevi, Ibu Masayu Leylia Khodra, dan Bapak Rinaldi Munir selaku dosen pengajar mata kuliah IF2211 Strategi Algoritma.

REFERENSI

- [1] Munir, Rinaldi, *Diktat Kuliah IF2211, Strategi Algoritma*, Program Studi Teknik Informatika, STEI, ITB, 2009
- [2] <http://algorithms.tutorialhorizon.com/introduction-to-backtracking-programming/> diakses pada 18/05/2017 pukul 20:57

- [3] <http://www.geeksforgeeks.org/branch-and-bound-set-1-introduction-with-01-knapsack/> diakses pada 18/05/2017 pukul 21:39
- [4] <http://ieeexplore.ieee.org/document/5483286/> diakses pada 18/05/2017 pukul 21:30

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 2012



Rio Dwi Putra Perkasa
13515012