

# Algoritma Backtracking Pada Logic Game : Family Crisis (Game Penyebrangan)

Muhammad Husain Jakfari 13512067  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
<sup>1</sup>emha.husain104@gmail.com

**Abstract**—Banyak Sekali Game-game yang dapat mengasah otak dan nalar kita. Salah satu game tersebut ialah game yang saya temukan di internet. Game ini tentang sebuah keluarga yang ingin menyebrang sebuah jurang menggunakan kayu yang hampir patah, mereka hanya punya waktu tertentu untuk menyebrang, sedangkan tiap-tiap anggota keluarga mempunyai kecepatan masing-masing, ada yang lambat dan ada yang cepat. Dan kayu tersebut hanya bisa digunakan oleh maksimal dua orang tiap satu kali penyebrangan. Goal dari game ini ialah semua anggota keluarga dapat menyebrang dalam waktu yang ditentukan tersebut.

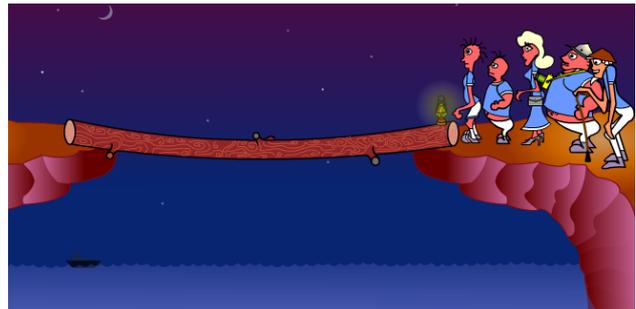
**Index Terms**—Backtracking, Waktu.

## I. PENDAHULUAN

Game adalah salah satu sarana bagi manusia untuk mengembalikan *mood* atau suasana hati kita yang semula buruk akibat terlalu banyak tekanan dari luar maupun dalam seperti tugas yang melimpah, ujian yang bertubi-tubi, atau masalah lain yang membuat kita *down*. Dengan bermain game diharapkan kita menjadi “tersegarkan” kembali dan siap menghadapi tantangan selanjutnya. Dewasa ini banyak sekali jenis-jenis game yang ada, baik itu dalam bentuk elektronik maupun benda nyata. Game-game itupun sekarang sudah dikelompok-kelompokkan berdasarkan banyak hal mulai dari manfaatnya sampai media permainan. Game yang menjadi Perhatian bagi penulis dalam hal ini ialah kategori game yang mengasah otak, atau bisa kita sebut sebagai logic game. Game jenis ini sangat unik, karena selain kita menyegarkan diri kita juga dapat mengasah kemampuan kita dalam hal logika, nalar, dan pengetahuan-pengetahuan umum lainnya. Ada banyak sekali bentuk-bentuk game yang mengasah otak mulai dari puzzle sampai ke bentuk yang abstrak.

Salah satu game yang saya ingin teliti ialah, game flash asah otak tentang menyebrang. Game ini ada tiga versi, masing-masing versi memiliki tingkat kesusahan yang berbeda. Versi satu menceritakan pengembala domba, versi 2 kanibal dan manusia, dan yang ketiga ialah sebuah keluarga yang ingin menyebrang jurang. Penulis akan membahas game pada versipada tingkat kesusahan 3, yaitu dimana sebuah keluarga akan menyebrang sebuah

jurang yang dapat dilihat pada gambar berikut.



Pada gambar tersebut, terdapat 5 anggota keluarga yang ingin menyebrang. Namun harus memenuhi ketentuan yang berlaku. Ketentuannya ialah waktu mereka semua sampai kesana maksimal 30 detik dengan kriteria masing masing 5 orang yang akan menyeberangi jembatan kayu tersebut, adalah si kurus, si kekar, si tante, si gendut, dan si kakek. dan mereka butuh waktu menyeberang yang berbeda-beda, yaitu 1 detik, 3 detik, 6 detik, 8 detik, 12 detik.

Dan ini gambar ketika sudah menang



Penulis ingin mengetahui keefektifan dari algoritma runut balik atau biasa disebut dengan backtracking ini dalam menyelesaikan persoalan permainan tersebut. Mengapa memilih algoritma backtracking? Karena penulis berhipotesa bahwa karakteristik algoritma backtracking yang menggunakan fungsi pembatas sangat cocok dengan persoalan yang terdapat dalam permainan ini. Oleh karena itu penulis ingin membuktikan

keefektifan algoritma runut balik atau backtracking ini dengan algoritma yang lainnya.

## II. DASAR TEORI

### A. Algoritma Backtracking

Algoritma Backtracking merupakan algoritma yang berdasar dari algoritma DFS atau Depth First Search.[2] Jika algoritma DFS melakukan pencarian solusi sampai ke daun baru melanjutkan ke cabang selanjutnya maka pada Algoritma backtracking ini pencarian akan dibatasi kedalamannya. Batas tersebut sesuai dengan fungsi pembatasnya.

Adapun komponen-komponen penting dalam algoritma backtracking ini, yaitu :

1. **Solusi persoalan.**
  - Solusi dinyatakan sebagai vektor dengan  $n$ -tuple:  
 $X = (x_1, x_2, \dots, x_n)$ ,  $x_i \in S_i$ .
  - Mungkin saja  $S_1 = S_2 = \dots = S_n$ .
  - Contoh:  $S_i = \{0, 1\}$ ,  $x_i = 0$  atau 1
2. **Fungsi pembangkit nilai  $x_k$**   
Dinyatakan sebagai predikat:  $T(k)$   
 $T(k)$  membangkitkan nilai untuk  $x_k$ , yang merupakan komponen vektor solusi.
3. **Fungsi pembatas**
  - Dinyatakan sebagai predikat  
 $B(x_1, x_2, \dots, x_k)$
  - $B$  bernilai *true* jika  $(x_1, x_2, \dots, x_k)$  mengarah ke solusi.
  - Jika *true*, maka pembangkitan nilai untuk  $x_{k+1}$  dilanjutkan, tetapi jika *false*, maka  $(x_1, x_2, \dots, x_k)$  dibuang.

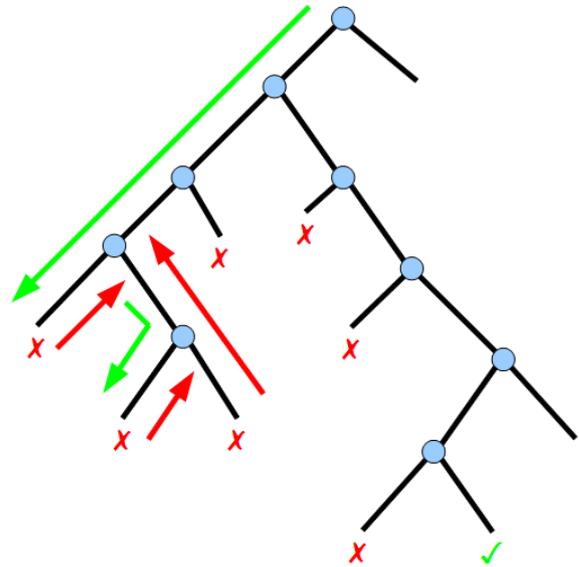
Adapun prinsip-prinsip dalam mencari solusi dalam algoritma runut balik ini ialah :

- Solusi dicari dengan membentuk lintasan dari akar ke daun. Aturan pembentukan yang dipakai adalah mengikuti aturan *depth-first order* (DFS).
- Simpul-simpul yang sudah dilahirkan dinamakan **simpul hidup** (*live node*).
- Simpul hidup yang sedang diperluas dinamakan **simpul-E** (*Expand-node*).
- Tiap kali simpul-E diperluas, lintasan yang dibangun olehnya bertambah panjang.
- Jika lintasan yang sedang dibentuk tidak mengarah ke solusi, maka simpul-E tersebut “dibunuh” sehingga menjadi **simpul mati** (*dead node*).
- Fungsi yang digunakan untuk membunuh simpul-E adalah dengan menerapkan **fungsi pembatas** (*bounding function*).
- Simpul yang sudah mati tidak akan pernah diperluas lagi.
- Jika pembentukan lintasan berakhir dengan

simpul mati, maka proses pencarian *backtrack* ke simpul aras di atasnya

- Lalu, teruskan dengan membangkitkan simpul anak yang lainnya.
- Selanjutnya simpul ini menjadi simpul-E yang baru.
- Pencarian dihentikan bila kita telah sampai pada *goal node*.

Ilustrasi dari langkah langkah ini dapat dilihat dalam gambar berikut.



### B. Algoritma Brute Force

Algoritma brute force merupakan algoritma yang paling mendasar.

*Brute force*: pendekatan yang lempang (*straightforward*) untuk memecahkan suatu persoalan.

Biasanya didasarkan pada:

- pernyataan pada persoalan (*problem statement*)
- definisi konsep yang dilibatkan.

Algoritma *brute force* memecahkan persoalan dengan

- sangat sederhana,
- langsung,
- jelas (*obvious way*).

Mengapa algoritma ini dinamakan brute force, karena kepolosan dalam algoritma ini dan algoritma ini terkesan *nguli* maksudnya ialah terlalu banyak mencoba keputusan, dan biasanya orang *nguli* itu berotot jadi itu sebabnya dinamakan brute force. Istilah lain dalam algoritma ini adalah *Just do it!* atau *Just Solve it!*

### C. Aturan Permainan

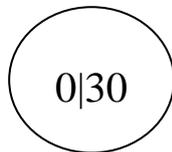
Permainan logika ini mungkin sederhana, namun ada beberapa aturan yang harus dipenuhi untuk menang. Aturannya ialah:

1. ada 5 orang yang akan menyeberangi jembatan kayu, yaitu sebut saja si kurus, si kekar, si tante,

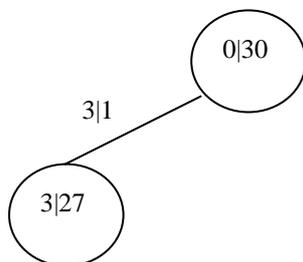
- si gendut, dan si kakek. masing2 dari mereka butuh waktu menyeberang yang berbeda-beda, yaitu 1 detik, 3 detik, 6 detik, 8 detik, 12 detik (dari kiri ke kanan).
- saat menyeberang orang harus membawa lampu, disini lampu tersebut hanya menyala selama 30 detik.
  - maksimal 2 orang yang boleh menyeberang bersama-sama, satu dari mereka harus kembali dengan membawa lampu.
  - saat dua orang menyeberang bersama-sama, waktu yang dibutuhkan sesuai dengan orang yang butuh waktu paling banyak (diantara 2 orang yang menyeberang tsb).[1]

### III. ANALISIS DAN PEMBAHASAN

Pertama-tama kita tentukan dulu kriteria node pada pohon solusinya. Node atau state tersebut merupakan kondisi kedua tempat penyebrangan yaitu kondisi jumlah orang yang terdapat pada sisi sisi jembatan kayu tersebut. Jadi akar dalam pohon tersebut adalah status dimana kiri kosong dan kanan terisi oleh seluruh anggota keluarga. Angka tersebut merupakan jumlah waktu yang dibutuhkan orang-orang tersebut untuk menuju keseberang.



Lalu tahap selanjutnya kita menentukan rangkaian aksinya. Karena aksinya pasti bolak balik maka setiap satu aksi menunjukkan jumlah waktu yang diperlukan untuk menyebrang dan kembali menjemput anggota keluarga yang belum menyebrang. Daun pada pohon ini jika angka pada saat kembali sama dengan 0.



Kodisi pada contoh node yang bawah di permainannya seperti gambar berikut ini



Contoh diatas menjelaskan bahwa manusia yang kecepatannya satu dan tiga menyebrang, lalu yang kecepatannya satu kembali lagi untuk menjemput anggota yang lainnya. Sementara yang 3 tetap tinggal di seberang.

Karena sesuai dengan peraturan bahwa setiap penyebrangan harus ada salah satu yang kembali jika yang lainnya ingin menyebrang, sampai tidak ada lagi yang akan menyebrang.

Tahap selanjutnya kita menentukan prioritas subpohon yang akan digali. Digali disini maksunya sebuah subpohon akan ditelusuri sampai daun. Ini sesuai dengan prinsip DFS, yaitu memprioritaskan kedalaman daripada pengaktifan seluruh anak pohon. Urutan orang berdasarkan kecepatannya ialah 1 3 6 8 dan 12, urutan pemilihan orang di bagian kanan ialah pasangan yang paling cepat dan belum pernah atau sudah lama menyebrang. Sedangkan yang sebelah kiri ialah satu orang yang paling cepat menyebrang.

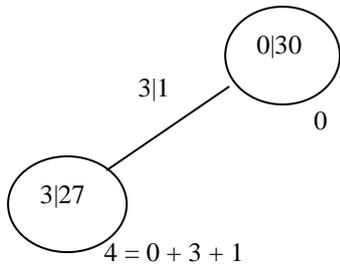
Contoh : (dari sudut pandang bagian jembatan yang kanan)

1 3 6 8 12 ---3|1---->1 6 8 12 ----8|3 ---> 1 3 12 ->.....

Orang yang kecepatannya 6 dan 8 dipilih daripada 1 dan 6 karena 1 sudah pernah menyebrang sebelumnya sedangkan 6, 8 dan 12 belum pernah menyebrang sebelumnya atau sudah lama tidak menyebrang disbanding yang lainnya.

Selanjutnya kita menentukan fungsi pembatas untuk membatasi kedalaman sebuah subpohon. Fungsi pembatasnya ini sudah ditentukan oleh permainannya sendiri, dalam permainan ini ialah 30, atau dapat dibilang sejumlah kecepatan seluruh orang yang akan menyebrang. Fungsi pembatas ini akan membatasi waktu yang telah ditempuh untuk melakukan kegiatan penyebrangan dari awal atau dari akar pohon solusi tersebut.

Waktu sebuah node yaitu jumlah dari aksi kanan dan kiri ditambah dengan jumlah node sebelumnya. Jika kurang jelas dapat dilihat pada gambar berikut ini.



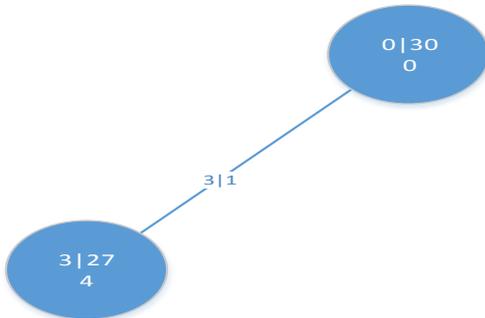
Setelah kita menentukan pembatasnya barulah kita menelusuri pohon solusi tersebut samapi mencapai solusi yang diinginkan. Tahap-tahap pencarian solusi adalah sebagai berikut.

Pertama-tama kita mulai dari akar kondisi pada akar ialah sebagai berikut.

0 | 1 3 6 8 12

Dari kondisi tersebut kita pilih orang dengan kecepatan 1 dan 3 satuan waktu, sesuai dengan prioritas yang sudah kita tentukan sebelumnya, dan orang yang kecepatannya 1 untuk kembali lagi ke sebelah kanan, jadi hasil akhirnya ialah seperti ini

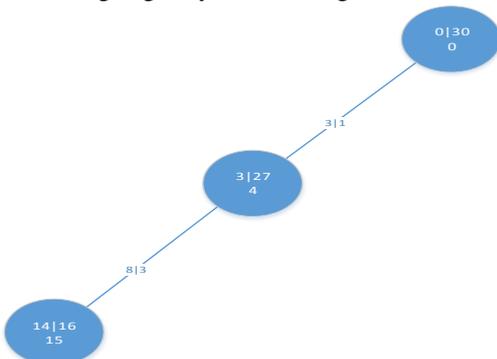
3 | 1 6 8 12  
Dengan graf pohonnya



Dengan waktu yang telah ditempuh oleh mereka untuk menyebrang dari awal sebesar 4 satuan waktu. Lalu kita memilih orang dengan kecepatan 6 dan 8 satuan waktu, sesuai dengan prioritas yang telah ditentukan. Lalu untuk kembali kita memilih orang dengan kecepatan tiga satuan waktu, sesuai juga dengan prioritas yang sudah ditentukan. Sehingga total waktu yang ditempuh dari awal sebanyak 4 diatmah dengan 8 dan 3 yaitu sebesar 15 satuan waktu. Kondisi pada node tersebut ialah:

6 8 | 1 3 12

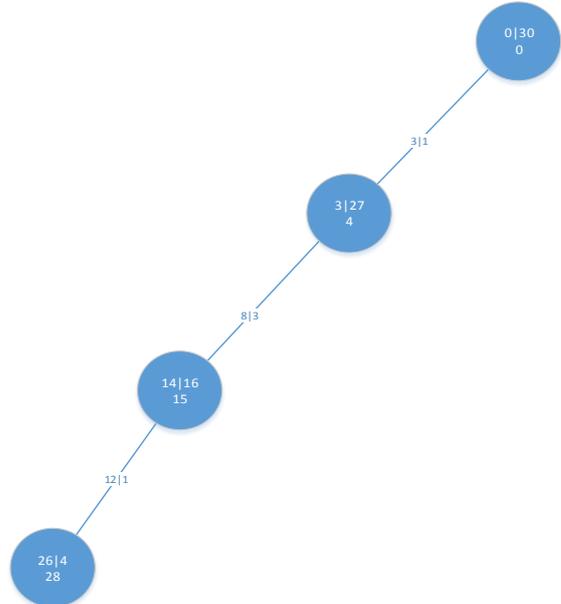
Dengan grafnya ialah sebagai berikut



Selanjutnya kita pilih 1 dan 12 dan 1 untuk kembali lagi

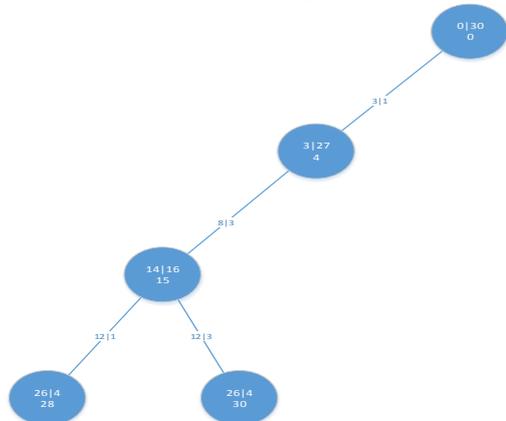
ke kanan sesuai dengan prioritas yang telah ditentukan sebelumnya. Dengan begitu total waktu yang diperlukan ialah 15 ditambah dengan 12 ditambah dengan 1 sebanyak 28 satuan waktu. Keadaan node dapat dilihat sebagai berikut.

6 8 12 | 1 3  
Grafnya ialah sebagai berikut



Karena sisanya tinggal 2 orang dan waktu diantara mereka yang paling besar adalah tiga maka  $28 + 3 > 30$ , maka node anak dari itu dibatasi (bounded) dan pencarian akan keblai ke "saudara" nya. Kita akan mengambil orang yang kecepatan menyebrangnya 3 dan 12 untuk dan yang kecepatannya 3 untuk kembali lagi ke sebelah kanan. Dengan tempuh waktu 15 ditambah dengan 12 ditambah dengan 3 sebanyak 30 satuan waktu. Keadaan node dapat dilihat sebagai berikut. Sehingga keadaan nodenya

6 8 12 | 1 3  
Grafnya ialah sebagai berikut



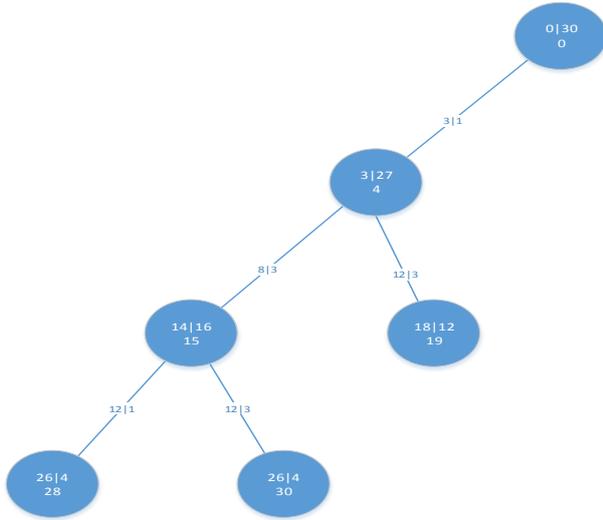
Waktu mereka menyebrang sudah 30 satuan waktu namun masih ada 2 orang yang belum berada sebelah kiri jembatan. Jadi node ini dibatasi dan pencarian naik kembali ke node saudara dari orang tua node tersebut.

Pada tahap ini kita pilih kembali orang yang berkecepatan 6 satuan waktu namun di pasangkan dengan

yang berkecepatan 12 satuan waktu serta yang berkecepatan 3 untuk kembali ke sebelah kanan. Waktu tempuhnya ialah 19 satuan waktu. Dengan begitu kondisi nodenya adalah sebagai berikut.

6 12 | 1 3 8

Dengan gambar grafnya ialah

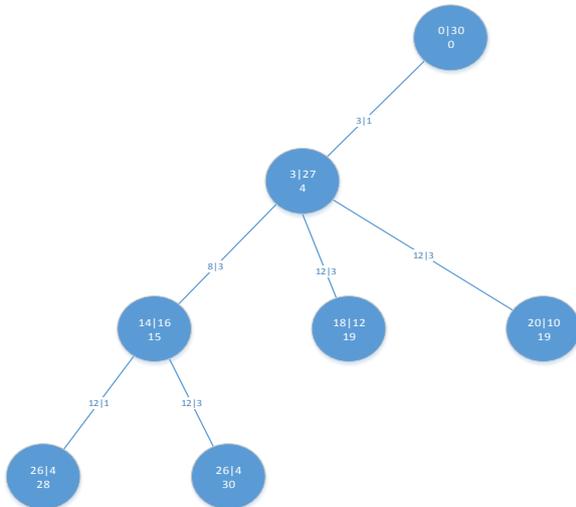


Graf ini memiliki anak anak yang sama persis dengan anak-anak yang dihasilkan pada graf saudara yang sudah dibangkitkan sebelumnya, karena kita tahu bahwa anak-anaknya tidak memenuhi batas, maka node ini langsung kita batasi, dan pencarian lanjut ke saudara yang lain.

Pada tahap ini kita memilih orang yang berkecepatan 12 dan 8 satuan waktu untuk menyebrang, dan orang yang berkecepatan 3 untuk menyebrang kembali ke kanan. Dengan begitu waktu tempuh yang dicapai untuk menyebrang dari awal mula penyebrangan ialah 19 satuan waktu. Kondisi node pada tahap ini adalah

8 12 | 1 3 6

Dengan graf nya ialah sebagai berikut

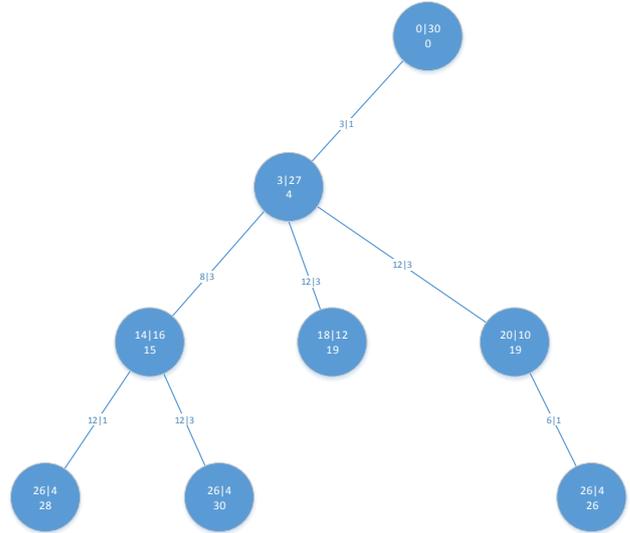


Kondisi ini masih memenuhi untuk di lanjutkan, karena anak-anak yang dihasilkan oleh node tersebut berbeda dari sebelumnya. Lalu kita ambil orang yang kecepatan 1 dan 6 untuk menyebrang ke kiri dan orang yang berkecepatan 1 untuk kembali ke kanan. Dengan begitu waktu yang

mereka tempuh dari awal memulai penyebrangan sebesar 26 satuan waktu. Kondisi akhir nodenya ialah:

6 8 12 | 1 3

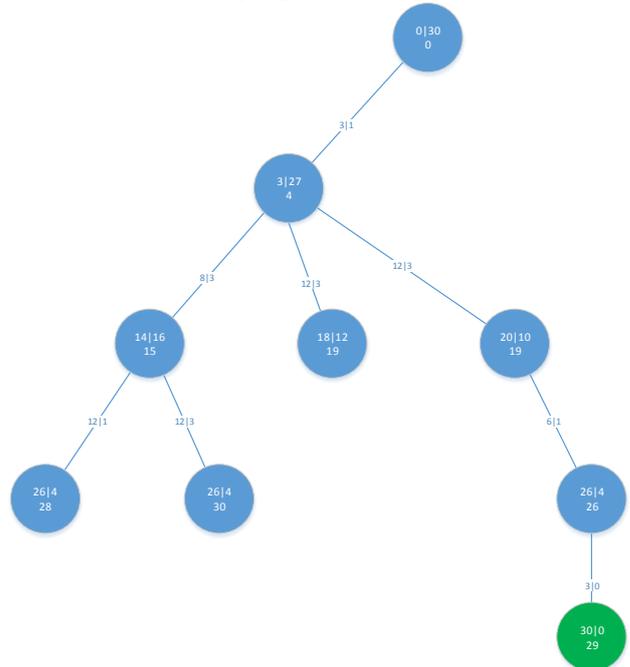
Dengan gambar grafnya ialah



lalu kita lanjut dengan memilih 2 orang yang tersisa di kanan yaitu orang yang berkecepatan 1 dan 3 dan tidak ada yang kembali karena sudah tidak ada orang lagi di sebelah kanan, dengan kata lain kita sudah mencapai daun atau solusi. Kondisi akhirnya ialah sebagai berikut.

1 3 6 8 12 | 0

Dengan grafnya ialah



Dengan begitu kita telah sampai ditahap solusi, dengan kata lain kita telah memenangkan permainan. Jumlah node yang dibangkitkan sebanyak 9 node. Dengan ke

Sekarang kita bandingkan dengan Algoritma brute force atau dengan kata lain menggunakan DFS murni, kita akan mendapatkan lebih dari 9 node karena node yang tadi kita batasi akan diteruskan sampai mencapai daun, atau semua

orang telah menyebrang jembatan kayu tersebut, hanya saja bukan merupakan solusi. Hal ini akan membuang-buang memori, karena membangun node butuh memori.

#### IV. KESIMPULAN

Berdasarkan hasil percobaan dapat disimpulkan bahwa algoritma runut balik atau backtracking sangat cocok untuk persoalan ini dibanding dengan algoritma yang konvensional atau algoritma bruteforce. Dengan begitu dapat dikatakan, sejauh ini algoritma yang cocok untuk permasalahan permainan logika ini ialah algoritma runut balik atau backtracking. Karena selain bentuk nya yang cocok juga tidak menghabiskan banyak memori disbanding algoritma konvensional lainnya

#### V. UCAPAN TERIMA KASIH

Penulis berterima kasih kepada Bapak Dr. Ir. Rinaldi Munir, MT. dan Ibu Dr. Masayu L. K., ST. Mt. selaku dosen mata kuliah strategi algoritma atas pengajaran yang telah beliau berikan kepada penulis untuk membantu pelaksanaan pembuatan paper ini.

Penulis juga ingin berterimakasih kepada rekan-rekan penulis setra pihak-pihak lain yang tak dapat penulis sebutkan satu per satu yang telah membantu mendukung penulis dalam penyusunan makalah ini.

#### REFERENSI

- [1] <http://alrasyid.heck.in/cara-menamatkan-game-flash-pc-berjudul-lxhtml> diakses pada tanggal 17 Mei 2014 pukul 20:00
- [2] Munir, Rinaldi. Diktat Kuliah IF 2211 Strategi Algoritma. Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2008, Hal. 167 -185.

#### 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, 18 Mei 2014

ttd

Handwritten signature in black ink, appearing to read 'Muhammad Husain Jakfari' with a stylized flourish below it.

Muhammad Husain Jakfari 13512067