

Pencarian Solusi Permainan *Pipe Puzzle* Menggunakan Algoritma Backtrack

Fahmi Dumadi 13512047¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

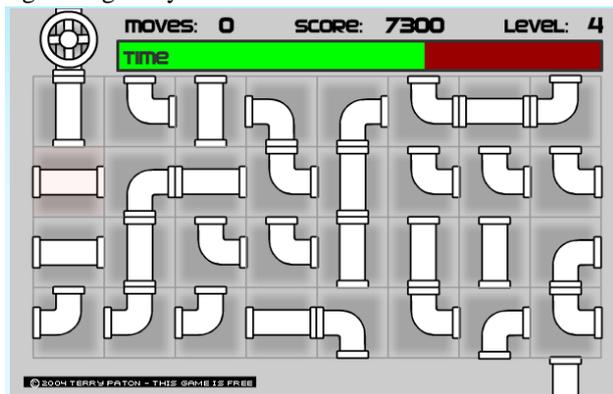
¹13512047@std.stei.itb.ac.id

Abstrak—Makalah ini membahas tentang pencarian solusi dalam permainan *pipe puzzle* atau sejenisnya dengan menggunakan algoritma backtrack. Pencarian solusi akan dijelaskan dengan suatu simbol setiap bentuk pipa dan urutan pengambilan keputusan dalam bentuk pohon serta solusi yang dihasilkan berupa jalur mencapai solusi atau solusi tidak ditemukan.

Kata Kunci—backtrack, *pipe puzzle*, simbol pipa, jalur solusi

I. PENDAHULUAN

Pipe puzzle adalah suatu permainan dengan tujuan menghubungkan suatu pipa sumber ke suatu pipa tujuan. Tugas seorang pemain adalah menghubungkan pipa-pipa tersebut dengan mengubah arah pipa-pipa yang menghubungkannya.

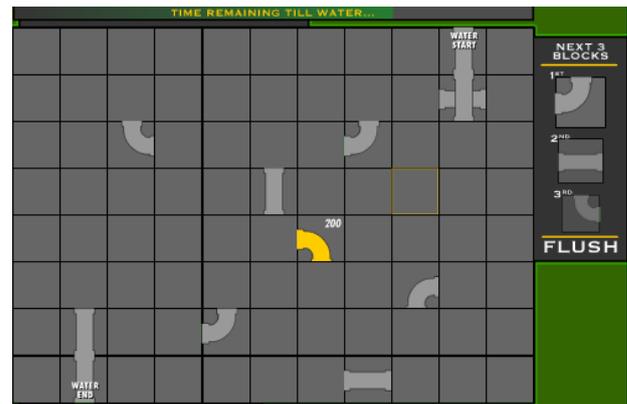


[2] Gambar 1.1 Contoh *pipe puzzle*

Banyak variasi dari permainan *pipe puzzle* ini, ada yang menggunakan air atau bola sebagai objek yang bergerak mengikuti alur pipa yang pemain bentuk, jadi pemain harus segera menyelesaikan penghubung pipa tersebut sebelum objek tersebut menemukan jalur buntu.

Ada juga yang menempatkan pipa penghubungnya sendiri sesuai dengan pipa yang muncul pada slot pipa yang tersedia namun permainan pipa jenis ini lebih sulit untuk pemodelan pencarian solusi menggunakan backtrack karena pipa yang muncul menggunakan fungsi random dan pemain bebas menaruh pipa-pipa tersebut dimana saja selama masih dalam kotak permainan seperti

gambar di bawah ini.



[3] Gambar 1.2 Variasi dari *pipe puzzle*

Pada makalah ini berusaha untuk menganalisis dan mencari solusi permainan *pipe puzzle* ini dengan cara menggunakan backtrack dengan membuat suatu simbol pada setiap bentuk pipa, memodelkan satu cara pencarian solusi pada suatu *pipe puzzle* dengan menggunakan pohon ruang status dan menyimpulkan solusi bisa dihasilkan atau tidak.

II. DASAR TEORI

Menurut buku *Diktat Kuliah IF2221 Strategi Algoritma* [1] runut-balik atau istilah asingnya backtrack, adalah algoritma yang memiliki basis pada DFS untuk mencari solusi persoalan secara lebih mangkus. Backtrack merupakan perbaikan dari algoritma Brute Force dan secara sistematis mencari solusi persoalan di antara semua kemungkinan yang ada. Dengan backtrack, kita tidak perlu memeriksa semua kemungkinan yang ada. Pencarian yang dilakukan hanya yang mengarah ke solusi saja yang selalu dipertimbangkan sehingga pencarian dapat dihemat dan mengurangi waktu pencarian. Backtrack lebih baik dinyatakan dalam algoritma rekursif.

Orang yang pertama kali memperkenalkan backtrack adalah D.H. Lehmer pada tahun 1950. Selanjutnya diuraikan secara umum oleh R.J. Walker, Golomb, dan Baumert. Algoritma backtrack banyak diterapkan pada program games (seperti tic-tac-toe, maze, dll) dan

berbagai masalah pada bidang kecerdasan buatan (*artificial intelligence*).

Untuk menerapkan metode backtracking, didefinisikan :

1. Solusi Persoalan

Solusi dinyatakan sebagai vektor dengan n -tuple:

$X = (x_1, x_2, \dots, x_n)$, $x_i \in$ himpunan berhingga S_i .

Mungkin saja $S_1 = S_2 = \dots = S_4$.

Contoh: $S_i = \{0, 1\}$

$x_i = 0$ atau 1

2. Fungsi pembangkit nilai x_k

Dinyatakan sebagai :

$T(k)$

$T(k)$ membangkitkan nilai untuk x_k , yang merupakan komponen vektor solusi.

3. Fungsi pembatas

Dinyatakan sebagai

$B(x_1, x_2, \dots, x_k)$

Fungsi pembatas menentukan apakah (x_1, x_2, \dots, x_k) mengarah ke solusi. Jika ya maka pembangkitan nilai untuk x_{k+1} dilanjutkan, jika tidak maka (x_1, x_2, \dots, x_k) dibuang dan tidak dipertimbangkan lagi dalam pencarian solusi.

Fungsi pembatas tidak selalu dinyatakan sebagai fungsi matematis, Ia dapat dinyatakan sebagai predikat yang bernilai true atau false, atau dalam bentuk lain yang ekuivalen.

Langkah-langkah pencarian solusi dengan menggunakan pohon ruang status adalah sebagai berikut.

1. Solusi dicari dengan membentuk lintasan dari akar ke daun. Aturan pembentukan yang dipakai adalah mengikuti metode DFS (Depth First Search). Simpul-simpul yang sudah dilahirkan dinamakan dengan **simpul hidup** (*live node*). Simpul hidup yang sedang diperluas dinamakan **simpul-E** (*Expand mode*). Simpul dinomori dari atas ke bawah sesuai dengan urutan kelahirannya.
2. Setiap 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 mati tidak akan pernah diperluas lagi.
3. Jika pembentukan lintasan berakhir dengan simpul mati, proses pencarian diteruskan dengan membangkitkan simpul anak yang lainnya. Bila tidak ada simpul anak yang dibangkitkan, maka pencatatan solusi dilanjutkan dengan melakukan backtrack ke simpul hidup terdekat (simpul orangtua). Selanjutnya simpul ini menjadi simpul-E yang baru. Lintasan baru dibangun kembali sampai lintasan tersebut membentuk solusi.
4. Pencarian dihentikan bila telah menemukan solusi

atau tidak ada lagi simpul hidup untuk backtrack.

III. PENCARIAN SOLUSI PERMAINAN PIPE PUZZLE MENGGUNAKAN ALGORITMA BACKTRACK

A. Penggunaan Simbol Pada Pipa

Bentuk pipa pada permainan *Pipe puzzle* bermacam-macam. Ada pipa sumber, pipa tujuan, dan pipa penghubung.



(a)



(b)

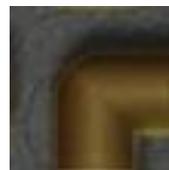
[4] Gambar 3.1 (a) pipa sumber, (b) pipa tujuan

Pipa sumber bisa kita simbolkan menjadi $\langle 0, \text{arah} \rangle$ dengan arah adalah arah awal saluran pipa yaitu up, down, right, dan left. Begitu juga dengan pipa tujuan bisa kita simbolkan menjadi $\langle 1, \text{arah} \rangle$ dengan arah adalah arah akhir pipa terakhir kali.

Pipa penghubung bermacam, macam juga bentuknya, ada yang belok, lurus, pertigaan dan perempatan. Berikut penjelasannya.

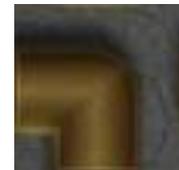
1. Pipa Belok

Ada 4 posisi pipa belok yang bisa kita pilih. Pipa belok berfungsi menggabungkan saluran pipa dengan membelokkan arahnya, maka bisa kita simbolkan menjadi $\langle AX, \text{arah1}, \text{arah2} \rangle$ dengan X adalah posisi pipa belok, arah1 adalah arah awal, dan arah2 adalah arah saluran pipa dibelokkan. Berikut pendefinisian simbol pada pipa belok.



$\langle A1, \text{up}, \text{right} \rangle$

$\langle A1, \text{left}, \text{down} \rangle$



$\langle A2, \text{right}, \text{down} \rangle$

$\langle A2, \text{up}, \text{left} \rangle$



$\langle A3, \text{right}, \text{up} \rangle$

$\langle A3, \text{down}, \text{left} \rangle$



$\langle A4, \text{down}, \text{right} \rangle$

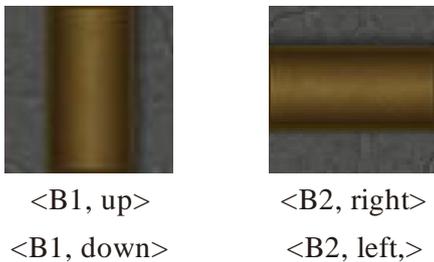
$\langle A4, \text{left}, \text{up} \rangle$

[4] Gambar 3.2 Pendefinisian simbol pada pipa belok

Setiap posisi memiliki 2 simbol karena arah saluran bisa berbeda tergantung lubang pipa yang pertama kali dimasuki.

2. Pipa Lurus

Ada 2 posisi pipa lurus yang bisa kita pilih. Pipa lurus berfungsi menggabungkan saluran pipa tanpa dibelokkan (arah diteruskan), maka bisa kita simbolkan menjadi $\langle BX, arah \rangle$ dengan X adalah posisi pipa lurus dan arah adalah arah saluran pipa diteruskan. Berikut pendefinisian simbol pada pipa lurus.

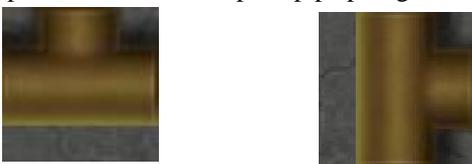


[4] Gambar 3.3 Pendefinisian simbol pada pipa lurus

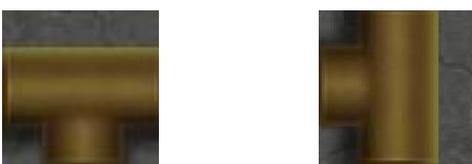
Sama seperti pada pipa belok, setiap posisi memiliki 2 simbol.

3. Pipa Pertigaan

Ada 4 posisi pipa belok yang bisa kita pilih. Pipa pertigaan berfungsi menggabungkan saluran pipa dan meneruskannya ke 2 arah yang berbeda, bisa dibelokkan atau diteruskan. Pipa pertigaan kita simbolkan menjadi $\langle CX, arah1, arah2 \parallel arah3 \rangle$ dengan X adalah posisi pipa pertigaan, arah1 adalah arah awal, dan arah2 atau arah3 adalah arah saluran pipa dibelokkan atau diteruskan. Berikut pendefinisian simbol pada pipa pertigaan.



$\langle C1, down, right \parallel left \rangle$
 $\langle C1, right, right \parallel up \rangle$
 $\langle C1, left, left \parallel up \rangle$
 $\langle C2, up, up \parallel right \rangle$
 $\langle C2, down, down \parallel right \rangle$
 $\langle C2, left, up \parallel down \rangle$



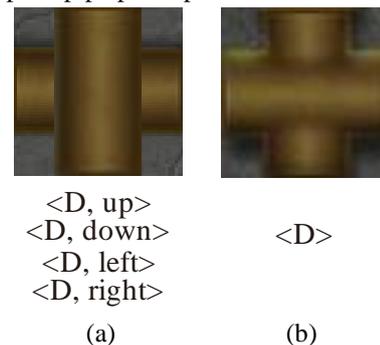
$\langle C3, up, right \parallel left \rangle$
 $\langle C3, right, right \parallel down \rangle$
 $\langle C3, left, left \parallel down \rangle$
 $\langle C4, up, up \parallel left \rangle$
 $\langle C4, down, down \parallel left \rangle$
 $\langle C4, right, up \parallel down \rangle$

[4] Gambar 3.4 Pendefinisian simbol pada pipa pertigaan

Pada pipa pertigaan setiap posisi memiliki 3 simbol karena lubang saluran pipa ada 3 sehingga arah akhirnya ada 2.

4. Pipa Perempatan

Pipa perempatan adalah pipa yang unik karena hanya ada 1 posisi dan bisa dilewati dua kali karena pipa ini sebenarnya adalah 2 pipa lurus yang saling menumpuk, jadi pipa perempatan ini hanya meneruskan saluran pipa. Namun ada juga yang mendefinisikan pipa perempatan itu adalah pipa yang bisa meneruskan ke segala arah selain arah awalnya. Pipa perempatan bisa kita simbolkan menjadi $\langle D, arah \rangle$ dengan arah adalah arah saluran pipa diteruskan dan satu lagi simbol untuk pipa perempatan yang meneruskan ke segala arah cukup dinyatakan dengan $\langle D \rangle$. Berikut pendefinisian simbol pada pipa perempatan.



[4] Gambar 3.5 Pendefinisian simbol pada pipa perempatan

Simbol pipa perempatan (a) adalah gabungan dari simbol-simbol pada pipa lurus.

B. Pencarian Solusi Menggunakan Backtrack

Pencarian solusi pada *pipe puzzle* akan dipaparkan sebagai berikut. Pertama-tama kita tentukan dulu fungsi pembatasnya, yaitu arah awal simpul anak yang akan dibangkitkan harus sama arah simpul orangtuanya, jadi simpul anak yang berbeda arahnya tidak perlu dibangkitkan karena tidak akan mengarah pada solusi. Lalu untuk urutan prioritas arah yang akan dipakai adalah tergantung posisi pipa sumber dan pipa tujuan berada.

Jika posisi pipa sumber berada di samping kiri pipa tujuan maka *right* lebih diprioritaskan dibanding *left*, dan sebaliknya. Jika posisi pipa sumber berada di atas pipa tujuan maka *down* lebih diprioritaskan dibanding *up*. Jadi jika posisi pipa sumber berada di kiri bawah area permainan dan pipa tujuan ada di kanan atas area permainan, urutannya adalah *right, up, left, down*. Posisi pipa berubah sesuai urutan angka, jika sudah mencapai posisi akhir, posisi kembali ke posisi pertama.

Setelah menentukan fungsi pembatas, maka pencarian solusi dilakukan dengan membuat pohon ruang status, agar terlihat urutan pembangkitan simpul sesuai pipa yang ditemui dan fungsi pembatas. Agar lebih jelas, akan kita

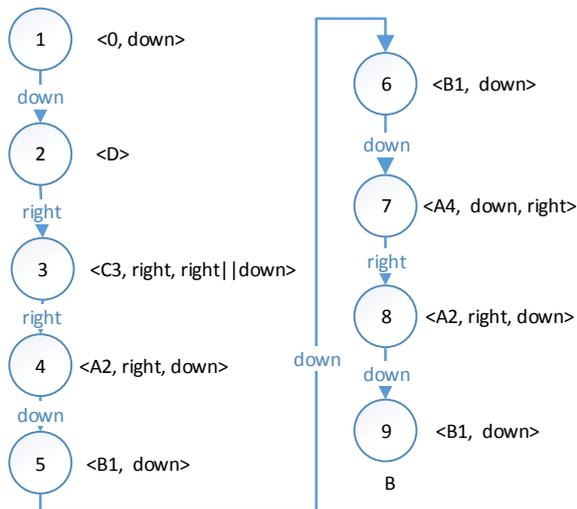
pecahkan salah satu *pipe puzzle* yang ada pada sebuah game. Berikut gambar area permainan *pipe puzzle* yang akan dipecahkan.



[4] Gambar 3.6 Contoh area permainan *pipe puzzle*

Bisa kita lihat bahwa pipa tujuan berada di kanan bawah pipa sumber. Jadi urutan prioritas arahnya adalah *right, down, up, left*. *Up* lebih diprioritaskan dibanding *left* karena arah pipa tujuannya adalah *up*. Kita simbolkan pipa sumber dengan $\langle 0, down \rangle$ dan pipa tujuan dengan $\langle 1, up \rangle$.

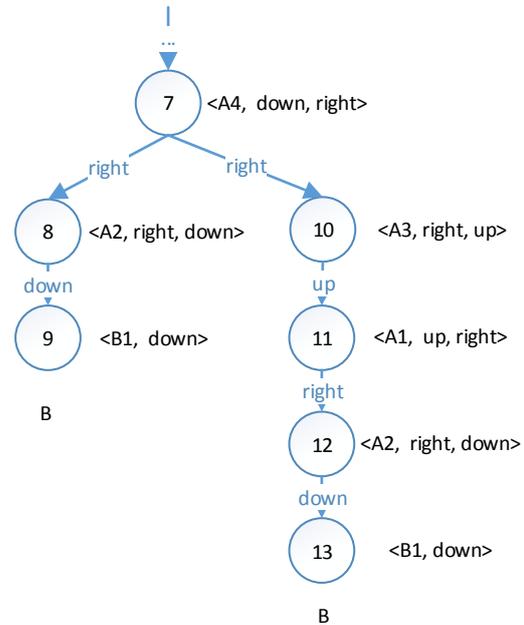
Simpul pertama (1) yang dihidupkan adalah pipa sumber dengan $\langle 0, down \rangle$. Lalu kita cek bentuk pipa yang ada di bawah pipa sumber yaitu pipa perempatan $\langle D \rangle$ dan kita hidupkan simpul anaknya (2). Karena simpul (2) hanya 1 posisi maka langsung saja dibangkitkan anaknya sesuai prioritas arah. Di samping kanan ada pipa pertigaan dengan bentuk sudah terhubung, jadi posisi tidak perlu diubah. Karena dari kanan, maka simpul anak (3) dibangkitkan dengan simbol $\langle C3, right, right \parallel down \rangle$. Lalu ke kanan lagi dan mengecek bentuk pipa yaitu pipa belok. Karena posisinya tidak terhubung maka posisinya diubah menjadi $\langle A2, right, down \rangle$ karena *down* prioritasnya lebih tinggi dibanding *up*. Simpul anak(4) dibangkitkan dari simpul(3). Sekarang arahnya menjadi ke bawah dan dicek pipa di bawahnya yaitu pipa lurus. Simpul anak(5) dibangkitkan dengan simbol $\langle B1, down \rangle$.



Gambar 3.7 Ruang Status 1

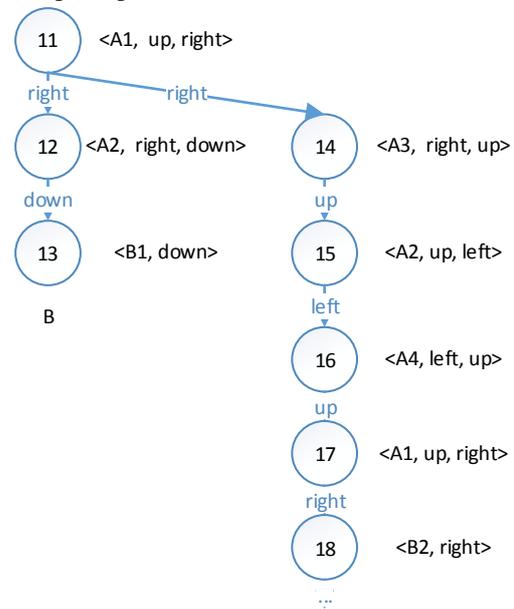
Karena di bawah simpul (9) merupakan batas area

permainan maka simpul (9) tidak mendekati solusi sehingga simpul tersebut dibunuh dan melakukan backtrack ke simpul (8). Di simpul (8) tidak ada lagi anak yang bisa dibangkitkan karena tidak ada posisi pipa lurus yang terhubung pada simpul (8), sehingga kembali melakukan backtrack. Di simpul (7) dibangkitkan simpul (10) yaitu $\langle A3, right, up \rangle$ sampai seperti ini.

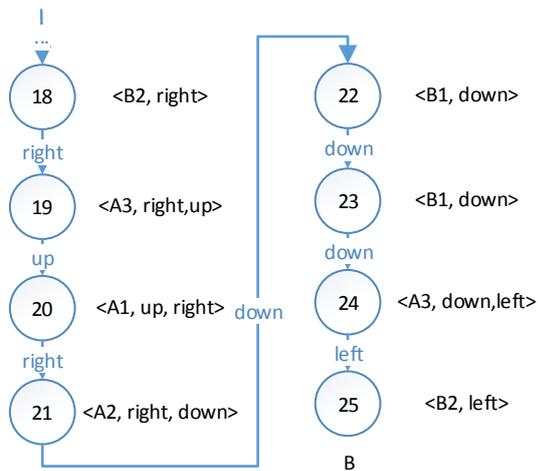


Gambar 3.8 Ruang Status 2

Simpul (13) dibunuh juga karena di bawahnya ada penutup pipa yang menutup jalur menuju solusi, sehingga dilakukan kembali backtrack ke simpul (12). Di simpul tersebut tidak ada anak lagi yang bisa dibangkitkan sehingga backtrack kembali ke simpul (11). Simpul (11) membangkitkan simpul anaknya (14) yaitu $\langle A3, right, up \rangle$ sampai seperti ini.

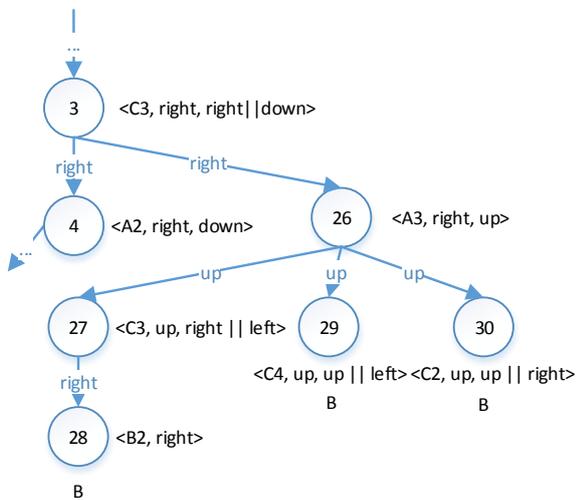


Gambar 3.9 Ruang Status 3.1



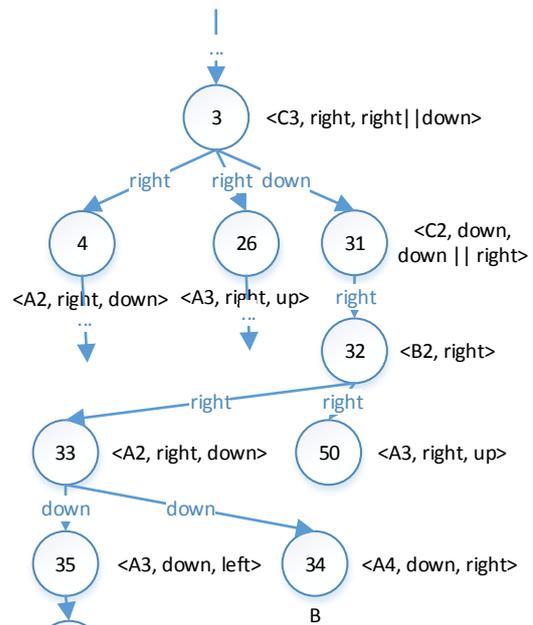
Gambar 3.10 Ruang Status 3.2

Pada simpul (25) tidak ada pipa yang bisa dihubungkan karena pipa di sebelah kirinya telah menjadi simpul sehingga dilakukan backtrack hingga simpul (16). Pada simpul (16) tidak bisa membangkitkan simpul anaknya karena pipa sebelah kiri simpul (17) telah dipakai juga, sehingga dilakukan backtrack kembali hingga simpul (3). Pada simpul (3) dibangkitkan simpul anak (26) yaitu <A3, right, up>, hingga seperti berikut.



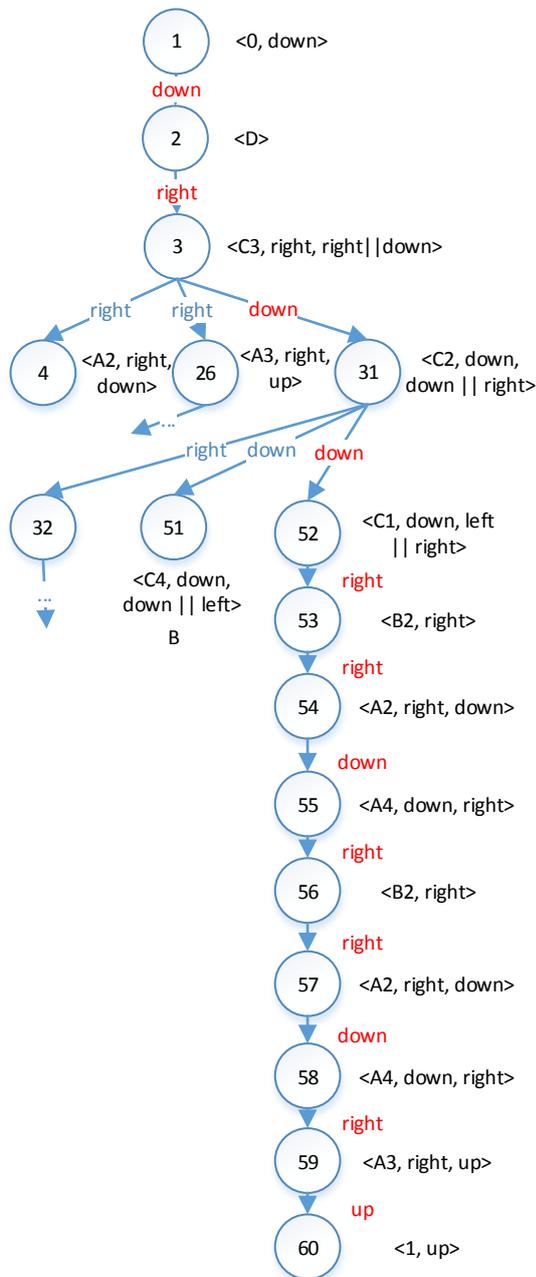
Gambar 3.11 Ruang Status 4

Pada simpul (28) jalur mengalami kebuntuan karena calon simpul anak merupakan objek penutup pipa. Backtrack dilakukan ke simpul (26). Pada simpul (29) tidak ditemukan calon simpul anak yang bisa diperluas, karena hanya ada objek penutup pipa dan batas area permainan. Pada simpul (30) calon simpul anak <B2, right> tidak dibangkitkan karena jalur tersebut sudah pernah dilewati. Oleh karena itu simpul (30) dibunuh menjadi simpul mati. Backtrack dilakukan sampai simpul (3) dan membangkitkan simpul anak (31) yaitu <C2, down, down || right>. Simpul (31) menjadi simpul-E dan melakukan perluasan ke sebelah kanan. Hasilnya seperti berikut.



Gambar 3.12 Ruang Status 5

Pada simpul (43) tidak dilakukan perluasan karena calon simpul anak merupakan jalur yang sudah diambil, sehingga dilakukan backtrack. Pada simpul (45) tidak dilakukan perluasan karena jalur terhalangi oleh objek penutup pipa sehingga dilakukan backtrack lagi. Pada simpul (46) tidak dilakukan perluasan karena sudah pernah dilewati. Pada simpul (47) tidak dilakukan perluasan karena jalur menjauhi pipa tujuan (dilihat pada posisi calon simpul anak lebih jauh dibanding posisi pipa sumber secara horizontal. Simpul (48), (49), dan (50) sama seperti penjelasan sebelumnya yaitu calon simpul anak merupakan jalur yang sudah pernah dilewati.



Gambar 3.13 Solusi Pohon Ruang Status

Solusi jalur yang tepat untuk dapat menghubungkan pipa sumber dengan pipa tujuan bisa dilihat pada gambar di atas. Fungsi pembatas sangatlah berpengaruh pada pencarian solusi backtrack. Tanpa fungsi pembatas, jalur yang harus diperiksa akan lebih banyak dari solusi ini, hal ini biasa dikenal dengan algoritma brute-force. Pendefinisian fungsi pembatas juga sangat berpengaruh pada pencarian solusi. Jika kita memprioritaskan *down* terlebih dahulu dibanding *right*, solusi *pipe puzzle* ini bisa kita dapatkan hanya dengan membangkitkan 13 simpul saja dan angka tersebut sama dengan jumlah langkah dari pipa sumber sampai dengan pipa tujuan. Pencarian solusi dapat lebih mangkus dengan algoritma lain seperti Branch & Bound atau Program Dinamis karena algoritma tersebut menggunakan fungsi biaya (cost). Jika dalam permainan ini fungsi cost dapat didefinisikan sebagai jarak antara pipa

sumber dengan pipa tujuan atau fungsi cost dapat didefinisikan juga dengan banyaknya pemain mengubah posisi pipa dengan cara mengklik mouse atau semacamnya.

IV. KESIMPULAN

Permainan *pipe puzzle* dapat dicari solusinya dengan cara algoritma backtrack. Fungsi pembatas pada algoritma backtrack sangat berpengaruh kepada kemangkusan pencarian solusi. Pembahasan makalah ini dapat dikembangkan lagi dengan mencari algoritma yang lebih mangkus untuk mendapatkan solusi

VII. PENGAKUAN

Pertama, penulis mengucapkan terima kasih kepada Allah SWT yang telah mengizinkan penulis untuk menyelesaikan makalah ini. Kedua, penulis mengucapkan terima kasih kepada orang tua penulis yang telah membesarkan dan mendidik penulis hingga saat ini. Ketiga, penulis mengucapkan terima kasih kepada para dosen dan asisten mata kuliah IF2211 Strategi Algoritma yang telah memberikan banyak hal yang sangat bermanfaat. Semoga segala kegiatan yang telah terjadi pada proses belajar mengajar ini menjadi pahala yang terus mengalir sampai akhir zaman. Akhir kata penulis mengucapkan mohon maaf bila ada kesalahan dalam penulisan. Sekian dan terima kasih.

DAFTAR PUSTAKA

- [1] Munir, Rinaldi. *Diktat Kuliah IF2211 Strategi Algoritma*. Bandung : Penerbit Teknik Informatika Institut Teknologi Bandung. 2009. Bab 7.
- [2] http://www.funnygames.in/game/the_plumber.html diakses tanggal 18 Mei 2014 pukul 08.10
- [3] <http://www.engineering.com/GamesPuzzles/Pipes/tabid/4683/Default.aspx> diakses tanggal 18 Mei 2014 pukul 08.20
- [4] <http://www.freeonlinegames.name/en/games/plumber-boy-2.html> diakses tanggal 18 Mei 2014 pukul 17.33

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, 19 Mei 2014

Fahmi Dumadi 13512047