

SOLUSI PERMAINAN CHEMICALS DENGAN ALGORITMA RUNUT BALIK

Irma Juniati

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jalan Ganesha 10, Bandung
e-mail: if16088@students.if.itb.ac.id

ABSTRAK

Permainan *Chemicals* adalah permainan untuk menentukan urutan empat warna yang benar dari sembilan pilihan warna. Ketika permainan dimulai, komputer akan melahirkan sebuah kombinasi warna secara acak untuk ditebak oleh pemain. Permainan berakhir jika pemain dapat menebak dengan benar susunan empat warna berbeda yang dimaksud oleh komputer. Komputer akan memberikan tanda yang berbeda untuk setiap warna, mulai dari warna yang ada pada kombinasi yang dimaksud dan urutannya benar, warna yang ada tetapi urutannya salah, serta warna yang sama sekali tidak termasuk dalam kombinasi warna yang dimaksud. Solusi dari permasalahan ini dapat ditemukan dengan berbagai macam algoritma, salah satunya adalah algoritma runut balik yang akan diuraikan dalam makalah ini. Algoritma runut balik ini berbasis pada algoritma DFS (*Depth First Search*). Simpul ruang solusi yang tidak mengarah ke ruang solusi akan dimatikan, sehingga tidak terlalu banyak langkah yang dihabiskan untuk mencari solusi yang tepat.

Kata kunci: DFS, runut balik, *chemicals*.

1. PENDAHULUAN

Permainan *Chemicals* ini sebenarnya terdapat dalam berbagai versi, dengan nama permainan yang juga berbeda-beda. Selain warna, terdapat juga permainan sejenis yang menggunakan objek yang berbeda, seperti urutan angka, kombinasi busana, dan sebagainya. Permainan ini sudah sejak dulu ada, dan perkembangan saat ini dilakukan dengan membuat berbagai variasi objek yang dikombinasikan, seperti yang telah disebutkan di sebelumnya.

Banyak algoritma yang dapat diimplementasikan untuk menyelesaikan persoalan ini. Misalnya dengan *brute force*, yakni dengan mencoba seluruh kombinasi empat warna yang mungkin, hingga ditemukan solusi yang

benar. Namun tentu saja solusi semacam ini tidak efisien. Oleh sebab itu, penulis mencoba untuk menyelesaikan persoalan ini dengan algoritma runut balik, yang penulis yakini dapat mencari solusi dengan lebih efisien.

2. ISI

Di dalam bab 2 ini, akan dibahas lebih lanjut mengenai peraturan dan cara bermain *Chemicals*, algoritma runut balik yang dipakai beserta penerapannya dalam penyelesaian persoalan dalam permainan ini.



Gambar 1 Tampilan awal permainan *Chemicals*

2.1 Peraturan dan Cara Bermain *Chemicals*

Chemicals dimainkan dengan mencoba berbagai kombinasi warna, dengan mengacu pada petunjuk yang diberikan komputer, yakni sebagai berikut:

- ✓ jika warna yang ditebak oleh pemain terdapat dalam kombinasi yang dimaksud dan urutannya benar
- jika warna yang ditebak oleh pemain terdapat dalam kombinasi warna yang dimaksud, tetapi tidak berada pada urutan yang seharusnya
- ✗ jika warna yang ditebak oleh pemain tidak terdapat dalam kombinasi warna yang dimaksud

Berdasarkan ketiga petunjuk di atas, pemain dapat memperoleh urutan warna yang benar setelah mencoba beberapa kombinasi warna. Dengan menerapkan algoritma runut balik, solusi dapat ditemukan dengan lebih optimal, dengan tidak mencoba kombinasi warna yang tidak mungkin.

2.2 Algoritma Runut Balik[1]

Runut balik adalah algoritma yang berbasis pada *DFS* untuk mencari solusi persoalan secara lebih mangkus. Istilah runut balik pertama kali diperkenalkan oleh D.H. Lehmer pada tahun 1950. Selanjutnya, R.J. Walker, Golomb, dan Baumert menyajikan uraian umum tentang runut balik dan penerapannya pada berbagai persoalan. Saat ini algoritma runut balik banyak diterapkan untuk program *games* dan masalah-masalah pada bidang kecerdasan buatan.

Untuk menerapkan metode runut balik, properti berikut didefinisikan:

1. Solusi persoalan
Solusi dinyatakan sebagai vektor dengan *n-tuple*:

$X = (x_1, x_2, \dots, x_n), x_i \in \text{himpunan berhingga } S_i$
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:

$T(k)$

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

3. Fungsi pembatas (pada beberapa persoalan fungsi ini dinamakan fungsi kriteria)
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, tetapi 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.

Di sini kita hanya akan meninjau solusi pada pohon ruang status yang dibangun secara dinamis. Langkah-langkah pencarian solusi adalah sebagai berikut:

1. Solusi dicari dengan membentuk lintasan dari akar ke daun. Aturan yang dipakai adalah *DFS*. Simpul-simpul yang dilahirkan dinamakan **simpul hidup** (*live node*). Simpul hidup yang sedang diperluas dinamakan **simpul-E** (*expand node*). Simpul dinomori dari atas ke bawah sesuai dengan urutan kelahirannya.
2. Jika lintasan yang sedang dibentuk tidak mengarah ke ruang solusi, maka simpul-E tersebut “dibunuh” sehingga menjadi **simpul mati** (*dead node*) dengan menggunakan **fungsi pembatas** (*bounding function*). Simpul yang sudah mati tidak akan pernah diperluas lagi.
3. Jika pembentukan lintasan berakhir dengan simpul mati, maka proses pencarian diteruskan dengan membangkitkan simpul anak lainnya. Bila tidak ada lagi simpul anak yang dapat dibangkitkan, maka pencarian solusi dilanjutkan dengan melakukan runut balik ke simpul hidup terdekat (simpul orang tua). Selanjutnya simpul ini menjadi simpul-E yang baru. Lintasan baru dibangun kembali sampai lintasan tersebut membentuk solusi.
4. Pencarian dihentikan bila kita telah menemukan solusi atau tidak ada lagi simpul hidup untuk runut balik.

2.3 Penerapan Algoritma Runut Balik

Untuk mempermudah penulisan, penulis menerapkan ketentuan sebagai berikut:

- Sembilan warna yang ada disimbolkan dengan huruf A-I, dan disimpan dalam sebuah senarai warna secara urut dari abjad terkecil.



Gambar 2 Karakter A-I digunakan untuk merepresentasikan warna-warna yang ada

- Setiap simpul yang bertanda silang (tidak terdapat dalam kombinasi warna) akan langsung dihapus dari senarai warna.

Adapun algoritma yang penulis terapkan adalah sebagai berikut:

1. Bangkitkan simpul anak sampai mendapatkan simpul dengan warna dan urutan yang benar. Sebelum kondisi ini tercapai, matikan semua simpul yang dibangkitkan karena tidak mengarah ke ruang solusi. Untuk warna yang tidak terdapat dalam kombinasi warna yang dimaksud, akan dihapus dari senarai.
2. Ekspansi simpul tersebut.
3. Ulangi langkah 1 hingga elemen dalam senarai habis (senarai kosong), penanda bahwa solusi telah ditemukan.

Sebagai contoh, misalkan urutan yang dicari adalah H-E-B-A. Maka, berdasarkan algoritma runut balik yang penulis terapkan, urutan pemecahan masalah beserta pembentukan pohon ruang statusnya adalah sebagai berikut:

Simpul 1 sudah dibangkitkan pada awal permainan.

1. Bangkitkan simpul 2

Ambil elemen pertama dari senarai, yaitu warna A. Untuk warna-warna selanjutnya akan diambil sesuai urutan pada senarai.



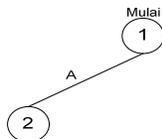
Gambar 3 Langkah 1

Warna A terdapat dalam kombinasi yang dimaksud, tetapi bukan di urutan 1.

Kondisi senarai sekarang:

A - B - C - D - E - F - G - H - I

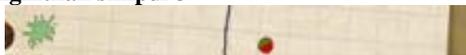
Pohon ruang status:



Gambar 4 Pohon ruang status pada langkah 1

Simpul 2 tidak mengarah ke ruang solusi, runut balik ke simpul 1, matikan simpul 2 dan bangkitkan simpul 3.

2. Bangkitkan simpul 3



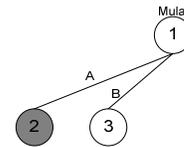
Gambar 5 Langkah 2

Warna B juga terdapat dalam kombinasi yang dimaksud, tetapi bukan di urutan 2.

Kondisi senarai sekarang:

A - B - C - D - E - F - G - H - I

Pohon ruang status:



Gambar 6 Pohon ruang status pada langkah 2

Simpul 3 tidak mengarah ke ruang solusi, runut balik ke simpul 1, matikan simpul 3 dan bangkitkan simpul 4.

3. Bangkitkan simpul 4



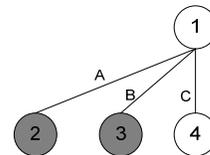
Gambar 7 Langkah 3

Warna C tidak terdapat dalam kombinasi yang dimaksud, hapus C dari senarai warna.

Kondisi senarai sekarang:

A - B - D - E - F - G - H - I

Pohon ruang status:



Gambar 8 Pohon ruang status pada langkah 3

Simpul 4 tidak mengarah ke ruang solusi, runut balik ke simpul 1, matikan simpul 4 dan bangkitkan simpul 5.

4. Bangkitkan simpul 5



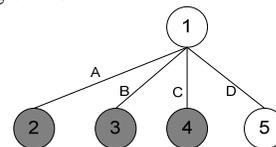
Gambar 9 Langkah 4

Warna D tidak terdapat dalam kombinasi yang dimaksud, hapus D dari senarai warna.

Kondisi senarai sekarang:

A - B - E - F - G - H - I

Pohon ruang status:



Gambar 10 Pohon ruang status pada langkah 4

Simpul 5 tidak mengarah ke ruang solusi, runut balik ke simpul 1, matikan simpul 5 dan bangkitkan simpul 6.

5. Bangkitkan simpul 6



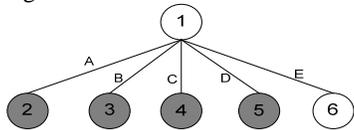
Gambar 11 Langkah 5

Warna E terdapat dalam kombinasi yang dimaksud, tetapi bukan di urutan 1.

Kondisi senarai sekarang:

A – B – E – F – G – H – I

Pohon ruang status:



Gambar 12 Pohon ruang status pada langkah 5

Simpul 6 tidak mengarah ke ruang solusi, runut balik ke simpul 1, matikan simpul 6 dan bangkitkan simpul 7.

6. Bangkitkan simpul 7



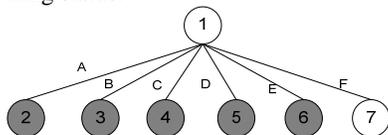
Gambar 13 Langkah 6

Warna F tidak terdapat dalam kombinasi yang dimaksud, hapus F dari senarai warna.

Kondisi senarai sekarang:

A – B – E – G – H – I

Pohon ruang status:



Gambar 14 Pohon ruang status pada langkah 6

Simpul 7 tidak mengarah ke ruang solusi, runut balik ke simpul 1, matikan simpul 7 dan bangkitkan simpul 8.

7. Bangkitkan simpul 8



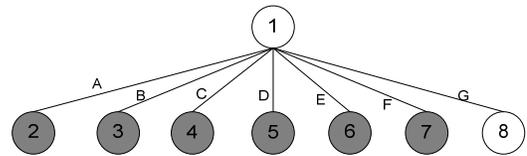
Gambar 15 Langkah 7

Warna G tidak terdapat dalam kombinasi yang dimaksud, hapus G dari senarai warna.

Kondisi senarai sekarang:

A – B – E – H – I

Pohon ruang status:



Gambar 16 Pohon ruang status pada langkah 7

Simpul 8 tidak mengarah ke ruang solusi, runut balik ke simpul 1, matikan simpul 8 dan bangkitkan simpul 9.

8. Bangkitkan simpul 9



Gambar 17 Langkah 8

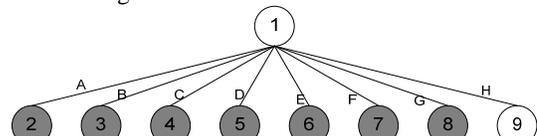
Warna H terdapat dalam kombinasi warna yang dimaksud dan berada pada urutan yang tepat. Dengan kata lain, warna H adalah warna pertama dalam urutan, dan simpul 9 akan diekspansi untuk menentukan urutan warna selanjutnya.

Karena pemain hanya diminta menentukan empat urutan warna, dan telah terdapat empat warna yang termasuk dalam kombinasi yang dimaksud oleh komputer, maka warna I dihapus dari senarai, karena tidak mungkin dieksekusi lebih lanjut. Warna H juga dihapus karena simpul 9 akan diekspansi, dan satu warna tidak dapat digunakan lebih dari satu kali.

Kondisi senarai sekarang:

A – B – E

Pohon ruang status:



Gambar 18 Pohon ruang status pada langkah 8

9. Ekspansi simpul 9



Gambar 19 Langkah 9

Mirip dengan langkah-langkah sebelumnya, sekarang simpul-E yang dimaksud adalah simpul 9. Ekspansi dilakukan dengan mengambil elemen pertama senarai sebagai anak pertama. Proses yang terjadi adalah sebagai berikut:

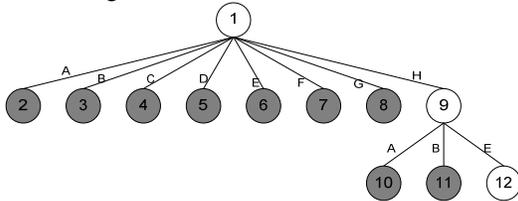
Warna A tidak mengarah ke ruang solusi, runut balik ke simpul 9, matikan simpul 10 dan bangkitkan simpul 11. Demikian juga dengan simpul 11. Warna B tidak mengarah ke ruang solusi sehingga simpul 12 dibangkitkan.

Warna E dihapus dari senarai karena simpul 12 akan diekspansi.

Kondisi senarai sekarang:

A – B

Pohon ruang status:



Gambar 10 Pohon ruang status pada langkah 9

10. Ekspansi simpul 12



Gambar 21 Langkah 10

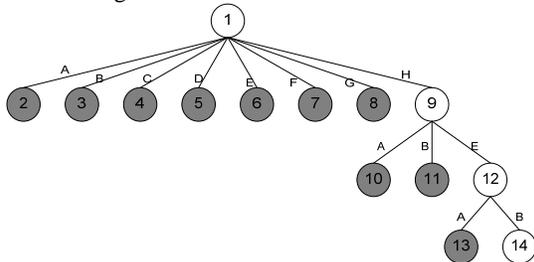
Warna A terdapat dalam kombinasi yang dimaksud, tetapi bukan di urutan 3.

Warna B dihapus dari senarai karena simpul 12 akan diekspansi.

Kondisi senarai sekarang:

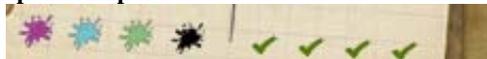
A

Pohon ruang status:



Gambar 22 Pohon ruang status pada langkah 10

11. Ekspansi simpul 14

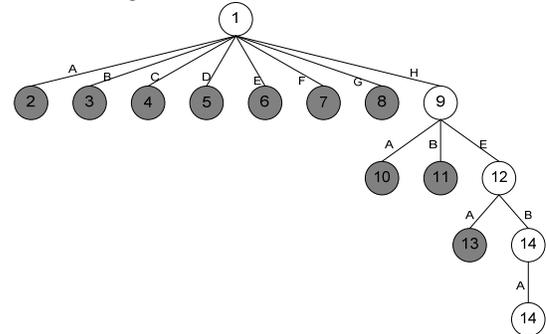


Gambar 23 Langkah 11

Permainan selesai setelah urutan warna yang benar berhasil ditebak.

Warna A dihapus dari senarai sehingga senarai sekarang kosong.

Pohon ruang status:



Gambar 24 Pohon ruang status pada langkah 11

3. KESIMPULAN

Algoritma runut balik cukup mangkus untuk menyelesaikan persoalan dalam permainan *Chemicals*. Penentuan fungsi pembatas yang baik akan menentukan kemangkusannya algoritma ini. Seperti contoh di atas, meskipun pada pembangkitan anak pada aras ke-1 membangkitkan hingga delapan simpul anak, namun untuk aras berikutnya jauh berkurang karena elemen dalam senarai juga dihapus sesuai ketentuan yang ada. Oleh karena itu, simpul anak yang dibangkitkan juga menjadi jauh lebih sedikit daripada jika menggunakan algoritma *DFS* biasa.

REFERENSI

- [1] Munir, Rinaldi, "Strategi Algoritmik", Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2007, 125-149.