

ALGORITMA BACKTRACK DALAM PEMECAHAN TEKA-TEKI LOGIKA “MERLIN MENANTANG ARTHUR”

Andreas Parry Lietara

Program Studi Teknik Informatika,
Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
e-mail: andreas-parry@students.itb.ac.id

ABSTRAK

Permainan teka-teki sudah sangat terkenal di masyarakat. Teka-teki sering digunakan sebagai pengisi waktu luang, *ice breaker*, ataupun pengasah otak. Teka-teki logika adalah jenis teka-teki yang berkaitan dengan logika atau penalaran. Umumnya penyuka teka-teki logika adalah mereka yang senang akan tantangan untuk berpikir.

Salah satu jenis teka-teki logika adalah teka-teki “Merlin Menantang Arthur”. Teka-teki ini dapat dilihat dalam salah satu seri buku “Sip Asyik Buku Teka Teki Kuis”. Pada teka-teki ini, penebak akan diberikan beberapa objek dengan sifat-sifat atau predikatnya, tidak secara lengkap, dan beberapa fakta yang dapat digunakan untuk melengkapi pasangan objek dengan predikat tersebut.

Makalah ini akan membahas pemecahan masalah tersebut dengan menggunakan algoritma yang terstruktur, sehingga bagi mereka yang masih kesulitan dan menebak-nebak dalam menyelesaikan teka-teki seperti ini dapat terbantu. Selain itu, teka-teki ini juga menggambarkan kehidupan kita sehari – hari, dimana kita juga sering memperoleh informasi yang setengah – setengah dan tidak utuh. Algoritma yang akan digunakan dalam penyelesaian masalah ini adalah algoritma *backtrack* beserta teknik-teknik *heuristic* tambahan. Penggunaan teknik-teknik *heuristic* tersebut bertujuan untuk lebih mengefisienkan proses penyelesaian masalah teka-teki ini.

Kata kunci: Teka-teki logika “Merlin menantang Arthur”, *Backtrack*, *heuristic*.

1. PENDAHULUAN

Teka-teki logika “Merlin menantang Arthur” adalah teka-teki untuk melengkapi pasangan objek dengan sifat / predikatnya. Berikut adalah gambaran teka-teki “Merlin

menantang Arthur” : Tiga orang laki-laki, A, B dan C datang ke suatu pesta mengendarai 1 dari 3 jenis mobil BMW, Porsche, atau Mercedes dan menggunakan warna pakaian tertentu, putih, hitam atau biru. Diberikan keterangan-keterangan bahwa, orang A mengenakan pakaian berwarna putih, orang C datang dengan Mercedes, dan orang yang mengenakan pakaian hitam datang naik Porsche. Dengan data-data ini pemain diminta untuk menentukan siapa mengenakan pakaian apa dan mengendarai apa.

2. METODE

Untuk menyelesaikan permasalahan teka-teki ini akan digunakan algoritma *backtrack*. Ide utama dalam penyelesaian masalah ini dengan algoritma *backtrack* adalah memasukkan kemungkinan anggota-anggota solusi himpunan solusi-solusi penyelesaian dengan slot-slot pasangan objek – predikat yang belum terisi. Jika terjadi kontradiksi dengan fakta-fakta yang diberikan akan dilakukan *backtrack* dan mengulang pemasukan anggota himpunan solusi yang lain hingga ditemukan jawaban yang tepat.

Langkah – langkah penyelesaian masalah yang akan dilakukan adalah sebagai berikut :

1. Membuat tabel hubungan objek dengan predikat – predikatnya.
2. Menentukan properti – properti dari algoritma *backtrack* yang terdiri dari :
 - a. Solusi persoalan, yang dinyatakan sebagai vektor dengan *n-tuple*, $X = (x_1, x_2, \dots, x_n)$.
 - b. Fungsi pembangkit nilai x_k , yang dinyatakan sebagai $T(k)$.
 - c. Fungsi pembatas, yang dinyatakan sebagai $B(x_1, x_2, \dots, x_k)$. [1]
3. Mengeliminasi calon – calon anggota solusi persoalan yang berkontradiksi dengan fakta – fakta yang diberikan.
4. Membuat dan mengembangkan ruang status berdasarkan calon – calon anggota solusi persoalan.
5. Jika terjadi status mati pada ruang status, lakukan *backtrack*.

- Pencarian berhenti jika kita telah menemukan solusi atau tidak ada lagi simpul hidup pada pohon ruang status untuk melakukan runut balik (*backtrack*).

2.1 Contoh Persoalan

Berikut adalah contoh persoalan teka-teki “Merlin menantang Arthur” :

Merlin ingin agar Arthur dapat menjadi seorang raja yang bijaksana, berani dan mulia. Untuk meningkatkan kekuatan nalar sang raja, Merlin menantang Arthur dengan teka – teki logika :

Pada malam tahun baru, empat orang ksatria tiba di kastil Camelot untuk menghadiri pesta akhir tahun dengan menunggang kuda yang dihiasi dengan umbul-umbul indah dengan warna bendera masing-masing ksatria. Nama – nama ksatria ini adalah Sir Pure, Sir Good, Sir Pious, dan Sir Venerable.

Dari petunjuk – petunjuk berikut, Arthur diminta untuk mengurutkan kedatangan empat orang ksatria ini dan warna hiasan masing – masing ksatria, yaitu : biru, emas, ungu, dan putih.

- Keempat ksatria ini adalah Sir Pious, ksatria kedua yang tiba, ksatria yang kudanya berhiaskan warna putih, dan Sir Pure.
- Sir Pious tidak tiba pertama kali, dan Sir Venerable bukanlah ksatria yang tiba sebelumnya.
- Kuda Sir Venerable tidak mengenakan hiasan warna putih.
- Sir Good tiba sebelum ksatria yang kudanya mengenakan hiasan warna biru, dan ksatria tersebut bukanlah Sir Pious.
- Sir Pure tidak mengenakan hiasan warna ungu. [2]

2.2 Penyelesaian Persoalan

Pembuatan dan inisialisasi tabel hubungan awal berdasarkan fakta nomor 1 :

Tabel 1 Tabel hubungan awal

Nama Ksatria	Kedatangan	Warna
Pious		
	Ke – 2	
		Putih
Pure		

Menentukan properti – properti algoritma *backtrack* :

- Calon – calon anggota solusi persoalan yang akan digunakan adalah pasangan nama – nama ksatria dan warna – warna yang masih mungkin (belum digunakan pada pengisian tabel awal) beserta pasangan nama – nama ksatria dan urutan kedatangan yang masih mungkin (belum digunakan pada pengisian tabel awal).
Calon – calon anggota solusi untuk permasalahan ini adalah :
 - (Pious , 1) ; (Pious , 3) ; (Pious , 4)
 - (Pious , biru) ; (Pious , emas) ; (Pious , ungu)
 - (Venerable , 2) ; (Good , 2)
 - (Venerable,1) ; (Venerable, 3) ; (Venerable, 4)
 - (Venerable , biru) ; (Venerable , emas) ; (Venerable , ungu) ; (Venerable , putih)
 - (Good,1) ; (Good, 3) ; (Good, 4)
 - (Good , biru) ; (Good , emas) ; (Good , ungu) ; (Good , putih)
 - (Pure , 1) ; (Pure , 3) ; (Pure , 4)
 - (Pure , biru) ; (Pure , emas) ; (Pure , ungu)

Perhatikan bahwa pembuatan pasangan – pasangan di atas adalah berdasarkan pasangan nama ksatria dengan tipe slot yang ingin diisi, atau nama ksatria dengan tipe slot yang telah diketahui isinya pada baris nama ksatria tersebut.

- Pengisian tabel akan dilakukan berurutan dari yang paling kiri atas ke kanan bawah, oleh karena itu fungsi pembangkitan adalah fungsi yang mencari pasangan yang sesuai untuk mengisi slot. Jika diketahui nama ksatria pada baris tersebut, maka fungsi akan mencari pasangan nama ksatria dengan tipe slot yang ingin diisi, sehingga jika kita ingin mengisi baris 1 kolom 2 pada tabel awal, fungsi pembangkitan akan mencari pasangan (Pious, x) dengan x adalah integer yang melambangkan urutan kedatangan. Sementara jika nama ksatria tidak diketahui, maka kita mencari info yang diketahui pada baris tersebut dan mencari pasangan nama ksatria dengan info tersebut, misalnya jika kita ingin mencari isi baris 2 kolom 1, fungsi akan mencari pasangan (x, 2) dengan x adalah nama ksatria.
- Fungsi pembatas pada persoalan ini adalah fungsi yang menerima seluruh isi dari tabel dan mengeceknya dengan fakta – fakta yang diberikan, jika terjadi kontradiksi maka akan dilakukan *backtrack*.

Langkah berikutnya adalah mengeliminasi calon – calon anggota solusi yang berkontradiksi dengan fakta – fakta yang ada.

Tabel 2 Tabel eliminasi calon – calon anggota solusi

Fakta	Calon anggota solusi yang dihilangkan
2	(Pious, 1)
3	(Venerable, putih)
4	(Pious, biru), (Good, 4), (Good, biru)
5	(Pure, ungu)

Setelah eliminasi calon – calon anggota solusi dilakukan maka calon – calon anggota solusi yang tersisa adalah :

1. (Pious , 3) ; (Pious , 4)
2. (Pious , emas) ; (Pious , ungu)
3. (Venerable , 2) ; (Good , 2)
4. (Venerable,1) ; (Venerable, 3) ; (Venerable, 4)
5. (Venerable , biru) ; (Venerable , emas) ; (Venerable , ungu)
6. (Good,1) ; (Good, 3)
7. (Good , emas) ; (Good , ungu) ; (Good , putih)
8. (Pure , 1) ; (Pure , 3) ; (Pure , 4)
9. (Pure , biru) ; (Pure , emas)

Proses pembuatan ruang status adalah pengisian tabel – tabel yang masih kosong dari kiri atas ke kanan bawah dengan mengambil calon – calon anggota :

Tabel 3 Tabel pembuatan ruang status

Nama Ksatria	Kedatangan	Warna
Pious	Ke – 3 (Pious, 3)	Emas (Pious, emas)
???	Ke – 2	
		Putih
Pure		

Pada pengisian slot baris 2 kolom 1 terjadi kontradiksi. Perhatikan bahwa baik Sir Venerable tidak bisa mengisi slot tersebut karena bertentangan dengan fakta 2, sementara Sir Good tidak bisa mengisi slot tersebut karena bertentangan dengan fakta 4. Oleh karena itu, di titik ini kita lakukan *backtrack*. Perlu diperhatikan bahwa *backtrack* dilakukan tidak ke status sebelumnya, melainkan ke status yang berhubungan dengan kontradiksi yang terjadi, pada kasus ini terjadi kontradiksi dengan masalah kedatangan dan warna. Namun, walau warna dari Sir Pious diganti ke biru, Sir Good tetap tidak dapat mengisi slot tersebut karena fakta 4, oleh karena itu kita mengubah kedatangan dari Sir Pious. Proses *backtrack* yang langsung menuju ke permasalahan ini bertujuan untuk memberikan efisiensi pada proses pencarian solusi. Kita ubah baris 1 kolom 2 menjadi :

Tabel 4 Tabel Bactrack I

Nama Ksatria	Kedatangan	Warna
Pious	Ke – 4 (Pious, 4)	Emas (Pious, emas)
Venerable (Venerable, 2)	Ke – 2	Biru (Venerable, biru)

Good (Good, putih)	Ke – 1 (Good, 1)	Putih
Pure	Ke – 3	???

Kali ini pengisian slot terakhir berkontradiksi dengan fakta 5. Kali ini kontradiksi terjadi pada bagian warna, sehingga kita akan mengubah status pengesetan warna pada baris – baris sebelumnya (tidak satu persatu dari status sebelumnya). pengisian baris 3 kolom 3 berdasarkan fakta yang diberikan, sehingga kita tidak bisa mengubahnya, jadi kita akan mengubah baris 2 kolom 3 :

Tabel 5 Tabel Backtrack II

Nama Ksatria	Kedatangan	Warna
Pious	Ke – 4 (Pious, 4)	Emas (Pious, emas)
Venerable (Venerable, 2)	Ke – 2	Ungu (Venerable, ungu)
???		Putih
Pure		

Namun perubahan tersebut tidak menyelesaikan masalah, sebab akan menyebabkan pengisian baris 3 kolom 1 berkontradiksi dengan fakta 4, oleh karena itu kita *backtrack* lagi ke baris 1 kolom 3, menjadi :

Tabel 6 Tabel Backtrack III

Nama Ksatria	Kedatangan	Warna
Pious	Ke – 4 (Pious, 4)	Ungu (Pious, ungu)
Venerable (Venerable, 2)	Ke – 2	Biru (Venerable, biru)
Good	Ke – 1 (Good, 1)	Putih
Pure	Ke – 3 (Pure, 3)	Emas (Pure, emas)

Dengan ini selesailah, permasalahan ini, sehingga urutan nama ksatria beserta kedatangan dan warna hiasannya adalah :

Tabel 7 Tabel solusi

Nama Ksatria	Kedatangan	Warna
Good	Ke – 1	Putih
Venerable	Ke – 2	Biru
Pure	Ke – 3	Emas
Pious	Ke – 4	Ungu

3. ANALISIS

Dari penyelesaian masalah di atas dapat dilihat bahwa algoritma *backtrack* dapat memberikan langkah – langkah pasti dalam menyelesaikan teka-teki tersebut. Sementara

itu, teknik – teknik *heuristic* seperti penghilangan calon anggota solusi yang tidak sesuai dengan fakta pada awal proses dan proses *backtrack* yang langsung menuju ke status permasalahan (bukan ke status sebelumnya) dapat mempercepat perolehan solusi.

Hal ini membuktikan algoritma *backtrack* dapat menyelesaikan permasalahan teka – teki “Merlin menantang Arthur”, sebab karakteristik dari teka – teki ini memang pantas diselesaikan dengan algoritma *backtrack*. Pada teka – teki ini kita tidak dapat langsung menentukan orang mana memakai warna apa dan datang pada urutan ke berapa secara langsung hanya menggunakan fakta, kita perlu untuk mencoba – coba salah satu kemungkinan yang ada (algoritma DFS) yang kemudian kita *backtrack* jika terjadi status mati (*backtrack* merupakan pengembangan dari algoritma DFS).

Sayangnya algoritma ini akan sedikit sulit untuk direalisasikan sebagai sebuah program. Kita harus mencari representasi yang tepat bagi fakta – fakta yang diberikan. Namun algoritma ini mudah dimengerti karena kurang lebih sama dengan pola pikir manusia saat mengerjakan persoalan tersebut, dan lagi jika algoritma ini berhasil direalisasikan menjadi sebuah, maka program tersebut dapat menyelesaikan masalah ini secara cukup efisien.

4. KESIMPULAN

Algoritma *backtrack* sangat sesuai dalam menyelesaikan permasalahan teka – teki “Merlin menantang Arthur”. Sayangnya algoritma ini masih cukup sulit direalisasikan sebagai program karena kita harus mencari representasi yang tepat untuk fakta – fakta yang ada. Namun demikian, algoritma ini dapat menyelesaikan permasalahan – permasalahan yang berbentuk seperti teka – teki di atas secara efisien.

Tentu saja masih banyak bentuk teka – teki lain yang memiliki pola seperti ini. Dalam pemecahannya yang harus kita lakukan adalah menyesuaikan ruang solusi yang diperlukan persoalan dan fungsi pembangkit nilainya.

REFERENSI

- [1] M. Rinaldi, “Diktat Kuliah IF 2251 Strategi Algoritmik”, Program Studi Teknik Informatika, 2006, Bandung, Indonesia.
- [2] Ibrahim, Raden Ikrar, “Teka – Teki Logika Untuk Meningkatkan Kecerdikan”, Studio RITO, 2005, Jakarta, Indonesia.