

# Penyelesaian Teka-Teki Logika dengan Menggunakan Strategi Backtracking

Muhammad Isro Fiordi (10216038)

Program Studi Fisika  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Bandung, Indonesia  
muhammadisrofiordi@s.itb.ac.id

**Abstract**—Teka-teki logika adalah sebuah masalah yang solusinya dapat ditentukan melalui deduksi secara logika. Ada banyak macam teka-teki logika yang sudah beredar di internet lengkap dengan solusinya. Tingkat dan bentuk teka-teki logika pun memiliki banyak variasi, dari yang mudah hingga sukar untuk dipecahkan seperti teka-teka logika Einstein. Proses penentuan solusi untuk teka-teki ini pada umumnya memerlukan penalaran secara deduktif. Pada makalah ini akan dirancang metode pemecahan teka-teka logika dengan menggunakan strategi *backtracking*. Algoritma *backtracking* adalah algoritma berbasis DFS untuk mencari solusi yang sebenarnya merupakan perbaikan dari algoritma *exhaustive search*. Dengan metode ini kita dapat memecahkan berbagai macam teka-teki logika tanpa harus melewati langkah penalaran logika yang memerlukan banyak waktu dan tenaga.

**Keywords**—*Backtracking, teka-teki, DFS, petunjuk.*

## I. PENDAHULUAN

Teka-teki adalah pernyataan atau pernyataan atau frasa yang memiliki makna ganda atau terselubung, yang diajukan sebagai masalah yang harus ditemukan jawabannya. Teka-teki terdiri dari dua jenis: teka-teki yang masalah umumnya dinyatakan dalam bahasa metafora atau alegoris dimana cara untuk menemukan solusinya dibutuhkan kecerdikan dan pemikiran yang cermat. Kemudian ada teka-teki yang disebut *conundra*, yang merupakan sebuah pertanyaan yang jawabannya ditunjukkan untuk permainan kata-kata atau pelanggaran standar pengetahuan umum dan bersifat menghibur.

Archer Taylor mengatakan bahwa "*we can probably say that riddling is a universal art*" dan mengutip teka-teki dari berbagai budaya yang berbeda seperti Finlandia, Hungaria, Amerika, India, Cina, Rusia, Belanda, dan Filipina dan sumber lain. Banyak teka-teki dan ide teka-teki tersebar luas di seluruh dunia. Setidaknya di kebudayaan Barat menurut Taylor teka-teki-teki dalam beberapa dekade terakhir sudah berhenti menjadi tradisi lisan. Tradisi tersebut telah digantikan dengan tradisi lisan lain seperti kuis.

Memecahkan teka-teki adalah kegiatan yang menyenangkan dan kegiatan ini dapat dilakukan oleh berbagai kalangan baik orang tua maupun anak-anak. Selain sebagai sarana rekreasi, teka-teka juga memiliki banyak manfaat seperti

meningkatkan pengetahuan kosa kata (pada teka-teki silang), peningkatan komprehensi pengetahuan untuk anak-anak, meningkatkan memori bagi kalangan lanjut usia.

Makalah ini lebih berfokus pada teka-teki dengan bentuk sebagai berikut:

- Diberikan sebuah situasi permasalahan
- Diberikan petunjuk dan batasan agar persoalan tidak menyebabkan multitafsir
- Diminta untuk menemukan solusi berdasarkan situasi dan petunjuk yang telah diberikan.

## Einstein Riddle

Di antara banyak teka-teki yang beredar di media internet, mungkin yang paling populer adalah teka-teki Einstein (*Einstein Riddle*). Konon 98% dari populasi manusia tidak dapat menyelesaikan teka-teki ini. Mengenai kebenaran apakah teka-teki ini dibuat oleh Einstein sendiri atau bukan adalah di luar bahasan makalah ini. Berikut teka-tekinya dalam bahasa Inggris:

### Situasi

1. *There are 5 houses in five different colors.*
2. *In each house lives a person with a different nationality.*
3. *These five owners drink a certain type of beverage, smoke a certain brand of cigar and keep a certain pet.*
4. *No owners have the same pet, smoke the same brand of cigar or drink the same beverage.*

### Petunjuk

- *the Brit lives in the red house*
- *the Swede keeps dogs as pets*
- *the Dane drinks tea*
- *the green house is on the left of the white house*
- *the green house's owner drinks coffee*
- *the person who smokes Pall Mall rears birds*
- *the owner of the yellow house smokes Dunhill*
- *the man living in the center house drinks milk*
- *the Norwegian lives in the first house*

- *the man who smokes blends lives next to the one who keeps cats*
- *the man who keeps horses lives next to the man who smokes Dunhill*
- *the owner who smokes BlueMaster drinks beer*
- *the German smokes Prince*
- *the Norwegian lives next to the blue house*
- *the man who smokes blend has a neighbor who drinks water*

*Pertanyaannya siapa yang memelihara ikan?*

Untuk memecahkan persoalan tersebut dapat dilakukan berbagai macam metode. Di antaranya dengan menggunakan bantuan tabel dan logika deduksi. Atau digunakan metode menebak solusi. Cara kedua tentunya tidak banyak menguras tenaga dibanding dengan cara pertama. Pada makalah ini kita akan menyelesaikan teka-teki yang berbentuk serupa dengan metode menebak solusi yang lebih terstruktur dan sistematis. Strategi ini dikenal dengan algoritma runut balik atau *backtracking*.

## II. DASAR TEORI

### A. Algoritma Traversal di Dalam Graf

Algoritma traversal dalam graf adalah strategi mengunjungi simpul-simpul dengan cara yang sistematis. Ada 2 macam algoritma traversal dalam graf yaitu pencarian melebar (*Breadth First Search* atau BFS) dan pencarian mendalam (*Depth First Search* atau DFS).

### B. Algoritma Pencarian Melebar (BFS)

BFS adalah algoritma traversal untuk mencari solusi pada pohon atau graf dengan prinsip memulai pencarian dari sebuah simpul (atau akar jika pada pohon) lalu diteruskan ke setiap tetangganya sebelum beralih ke aras berikutnya. Berikut langkah-langkahnya (traversal dimulai dari simpul v):

- 1) Kunjungi simpul v.
- 2) Kunjungi semua simpul yang bertetangga dengan simpul v terlebih dahulu.
- 3) Kunjungi simpul yang belum dikunjungi dan masih bertetangga dengan simpul yang sebelumnya dikunjungi, demikian seterusnya.

Untuk graf yang berbentuk pohon berakar, semua simpul pada aras a dikunjungi terlebih dahulu kemudian kunjungi simpul-simpul pada aras a+1.

Algoritma BFS tidak digunakan dalam penyelesaian masalah teka-teki karena algoritma ini akan melakukan penelusuran ke seluruh kemungkinan solusi dari teka-teki sehingga penelusuran tidak mangkus.

### C. Algoritma Pencarian Mendalam (DFS)

DFS adalah algoritma traversal untuk mencari solusi pada pohon atau graf dengan prinsip pencarian yang dimulai dari akar (untuk pohon) lalu diteruskan dengan memilih rute sejauh mungkin sebelum runut balik ketika menemui ujung simpul atau keadaan tertentu. Berikut langkah-langkahnya:

- 1) Kunjungi simpul v,
- 2) Kunjungi simpul w yang bertetangga dengan simpul v.
- 3) Ulangi langkah DFS dimulai dari simpul w.
- 4) Ketika mencapai simpul x sedemikian sehingga semua simpul yang bertetangga dengannya telah dikunjungi, runut balik pencarian ke simpul terakhir yang dikunjungi sebelumnya dan mempunyai simpul y yang belum dikunjungi.
- 5) Pencarian berakhir bila tidak ada lagi simpul yang belum dikunjungi yang dapat ditempuh dari simpul yang sudah dikunjungi.

Algoritma DFS ini akan dikembangkan lagi menjadi algoritma runut balik atau *backtracking* agar dalam pencarian solusi teka-teki tidak terlalu boros dalam hal pemeriksaan tebakan solusi.

### D. Penerapan Algoritma Traversal dalam Pencarian Solusi

Dalam pencarian solusi dari teka-teki yang telah dibahas di pendahuluan diperlukan sebuah struktur pencarian solusi dalam bentuk pohon dinamis. Pohon dinamis adalah pohon yang dibangun selama pencarian solusi berlangsung. Pohon dinamis ini dapat dibangun melalui metode BFS dan DFS. Pohon dinamis menyatakan status-status persoalan pada saat pencarian solusi berjalan. Berikut beberapa komponen pada pohon dinamis:

- 1) Status persoalan (*problem state*): simpul-simpul di dalam pohon dinamis yang memenuhi kendala (*constraints*).
- 2) Status solusi (*solution state*): satu atau lebih status yang menyatakan solusi persoalan.
- 3) Status tujuan (*goal state*): status solusi yang merupakan simpul daun.
- 4) Ruang solusi (*solution space*): himpunan semua status solusi.
- 5) Ruang status (*state space*): seluruh simpul di dalam pohon dinamis dan pohonnya dinamakan pohon ruang status (*state space tree*).

### E. Algoritma Runut-balik (Backtracking)

Algoritma runut-balik atau *backtracking* adalah algoritma traversal yang berbasis DFS untuk mencari solusi secara lebih mangkus. Algoritma ini sebenarnya adalah perbaikan dari algoritma *exhaustive search*, karena langkah-langkah penelusurannya dilakukan secara sistematis dan ada *heuristic* yang memberi batasan agar pencarian tidak terlalu melebar atau mendalam. Dengan metode ini tidak perlu memeriksa semua kemungkinan solusi yang ada. Hanya pencarian yang mengarah ke solusi saja yang dipertimbangkan. Dengan begitu waktu penelusuran solusi dapat dihemat. Runut-balik merupakan bentuk tipikal dari algoritma rekursif. Berdasarkan properti dari algoritma ini, teka-teki einstein dapat dicari solusinya tanpa harus melewati tahap penalaran yang melelahkan.

**F. Properti Umum Metode Runut-balik**

Ada beberapa properti pada algoritma runut-balik yang akan dijelaskan sebagai berikut:

- 1) Solusi Persoalan  
Solusi dinyatakan sebagai vektor dengan n-tuple:  
 $X = (x_1, x_2, \dots, x_n)$ ,  $x_i \in$  himpunan berhingga  $S_i$ .  
 $S_i$  boleh saja  $S_1 = S_2 = \dots = S_n$ .
- 2) Fungsi pembangkit nilai  $X_k$   
Fungsi ini dinyatakan sebagai  $T(k)$ .  $T(k)$  berfungsi untuk membangkitkan nilai untuk  $X_k$ , yang merupakan bagian dari komponen solusi.
- 3) Fungsi Pembatas / Fungsi Kriteria  
Fungsi pembatas dinyatakan sebagai  $B(x_1, x_2, \dots, x_k)$ . Fungsi ini untuk menentukan apakah  $(x_1, x_2, \dots, x_k)$  mengarah ke solusi. Jika benar, maka pembangkitan nilai untuk  $X_{k+1}$  diteruskan, tetapi jika tidak maka  $(x_1, x_2, \dots, x_k)$  dibuang dan tidak dipertimbangkan lagi dalam pencarian solusi.

Semua kemungkinan solusi dari persoalan disebut ruang solusi (*solution space*). Jika  $X_i \in$  maka  $S_1 \times S_2 \times \dots \times S_n$  disebut ruang solusi. Jumlah anggota di dalam ruang solusi adalah  $|S_1| \cdot |S_2| \cdot \dots \cdot |S_n|$ .

**G. Prinsip Pencarian Solusi dengan Metode Runut-balik**

Ada beberapa langkah yang membedakan metode pencarian runut-balik dengan metode DFS. Langkahnya dijelaskan sebagai berikut:

- 1) Solusi dicari dengan membentuk lintasan dari akar ke daun. Aturan pembentukan yang dipakai adalah mengikuti metode pencarian mendalam (DFS). Simpul-simpul yang sudah 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) 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 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 yang lain. Bila tidak ada lagi simpul anak yang dapat dibangkitkan maka pencarian solusi dilanjutkan dengan melakukan runut-balik 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 kita telah menemukan solusi atau tidak ada lagi simpul hidup untuk runut-balik.

**H. Strategi Runut-balik dalam Pemecahan Teka-Teki Logika**

Pada penyelesaian teka-teki logika, kita membatasi permasalahan hanya pada jenis teka-teki yang petunjuknya menjamin mengarahkan algoritma runut-balik kita ke satu solusi yang unik. Dengan begitu kita dapat memandang situasi dan petunjuk dari teka-teki sebagai fungsi kriteria yang akan selalu memeriksa simpul dari algoritma runut-balik yang dibangkitkan mengarah kepada solusi atau tidak. Berikut adalah contoh teka-teki logika yang memiliki bentuk yang sama seperti teka-teki einstein akan tetapi lebih sederhana.

Teka-teki logika 1:

**Situasi dan tujuan**

*Match each auctioned spider specimen (darkback, ipstin spider, redback, wolf spider) to its owner (Isaac, Jan, Melinda, Tim), and determine the final price paid for each (\$45, \$60, \$75, \$90)!*

**Petunjuk**

- 1) *Isaac’s purchase cost 30 dollars more than the wolf spider.*
- 2) *Tim’s purchase was the ipstin spider.*
- 3) *Melinda’s purchase cost 15 dollars less than the darkback.*
- 4) *The four spiders were Melinda’s purchase, the arachnid that sold for \$90, the spider that sold for \$60 and the spider that sold for \$75.*

Solusi dari teka-teki ini dalam bentuk tabel sebagai berikut:

Tabel 1. solusi teka-teki 1

Prices	Winners	Spiders
\$45	Melinda	wolf spider
\$60	Jan	darkback
\$75	Isaac	redback
\$90	Tim	ipstin spider

Dari bentuk tabel di atas, penentuan solusi hanya perlu mengisi kolom ke-2 (*winners*) dan kolom ke-3 (*spiders*). Ingat, yang berperan sebagai fungsi kriteria di persoalan ini adalah petunjuk dari teka-teki. Jadi ketika kita membangkitkan sebuah tebakan solusi untuk kolom ke-2, sebelum melanjutkan pengisian tebakan kolom ke-3 periksa dulu apakah tebakan pada kolom ke-2 sudah memenuhi kendala pada petunjuk. Berikut langkah-langkah penyelesaiannya:

- 1) Solusi berbentuk vektor yang memuat 2 vektor yaitu vektor *winners* dan *spiders*. Urutan penulisan pada vektor ini sesuai dengan vektor *prices* (\$45, \$60, \$75, \$90).

- 2) Akar diinisiasi dengan 0.
- 3) Bangkitkan sebuah kemungkinan solusi untuk vektor *winner*s. (contoh (Tim, Isaac, Melinda, Jan), (Tim, Melinda, Jan, Isaac) , ... dst) ini merupakan permasalahan permutasi.
- 4) Periksa apakah tebakan solusi memenuhi kendala (lihat apakah tebakannya sesuai dengan petunjuk yang diberikan). Jika tidak memenuhi kembali ke langkah 3. Jika memenuhi lanjutkan ke tahap 5.
- 5) Bangkitkan sebuah kemungkinan solusi untuk vektor *spider*s seperti pada langkah ke-3.
- 6) Periksa apakah tebakan solusi vektor *spider*s dan *winner*s memenuhi kendala yang diberikan oleh petunjuk. Jika tidak memenuhi kembali ke tahap 5 (jika tidak ada tebakan *spider*s lagi yang dapat dibangkitkan, kembali ke tahap 3). Misal tebakannya adalah  $\langle \text{(Tim, Melinda, Jan, Isaac), (redback, ipstin spider, darkback, wolf spider)} \rangle \rightarrow (\$45, \$60, \$75, \$90)$  tidak memenuhi kendala karena pada petunjuk nomor 2 laba-laba yang dibeli Tim adalah *ipstin spider*.
- 7) Pencarian dilakukan sampai ditemukan tebakan yang memenuhi kendala dari petunjuk.
- 8) Solusi ditemukan.

Di makalah ini kita mengasumsikan bahwa petunjuk yang diberikan menjamin kita menemukan sebuah solusi. Dengan begitu tugas perlu yang dilakukan hanya menebak solusi dan memeriksa kendala. Berbeda dengan penyelesaian secara umum yaitu melalui penalaran dan deduksi.

### III. PEMBAHASAN

Tinjau teka-teki logika 1 yang telah diberikan pada bab sebelumnya.

#### A. Properti Penyelesaian Teka-teki Logika dengan Algoritma Backtracking

##### 1) Pohon Status

status awal adalah akar yang berupa sebuah “kosong”. Setiap tebakan pada tiap vektor *winner*s dan *Spiders* menyatakan status persoalan. Keseluruhan dari komponen tersebut membentuk ruang status atau pohon status.

##### 2) Solusi Persoalan (S)

Solusi yang diinginkan berupa vektor yang memuat vektor *winner*s dan vektor *spider*s.

$$S = \{ (winners1, winners2, winners3, winners4) , (spiders1, spiders2, spiders3, spiders4) \}$$

Dimana:

$$Spiders = \{ redback, ipstin spider, darkback, wolf spider \}$$

$$Winners = \{ \text{Tim, Melinda, Jan, Isaac} \}$$

$$winners_i \in \text{Winners}, \text{Winners}_i \neq \text{Winners}_{i+1}$$

$$spiders_i \in \text{Spiders}, \text{spiders}_i \neq \text{spiders}_{i+1}$$

Urutan pada vektor *winner*s dan *spider*s sesuai dengan urutan kolom pertama (\$45, \$60, \$75, \$90). Oleh karena itu cara membaca notasinya (contoh):

$$\langle \text{(Tim, Melinda, Jan, Isaac), (redback, ipstin spider, darkback, wolf spider)} \rangle \rightarrow (\$45, \$60, \$75, \$90)$$

Dibaca sesuai urutan, spesimen laba-laba *redback* dibeli oleh Tim dengan harga \$45.

##### 3) Fungsi Pembangkit

Fungsi pembangkit dibagi menjadi 2 fungsi yaitu  $F_{spiders}(K)$  dan  $F_{winners}(K)$  yang didapat dari permutasi dari *Spiders* = { *redback, ipstin spider, darkback, wolf spider* } dan *Winners* = { Tim, Melinda, Jan, Isaac }.

##### 4) Fungsi Pembatas

Fungsi pembatas didapatkan dari petunjuk yang diberikan oleh teka-teki. Sebagai contoh dari petunjuk 3 dapat kita temukan beberapa fungsi pembatas (fungsi pembatasnya >1) diantaranya laba-laba yang dibeli Melinda harus lebih murah \$15 dari spesimen *darkback*. Laba-laba yang dibeli Melinda bukan spesimen *darkback*. Pada strategi ini bagian penerjemahan petunjuk menjadi fungsi pembatas adalah harga yang harus dibayar karena telah melewati langkah penalaran dan deduksi pada metode penyelesaian teka-teki secara umum. Berikut adalah beberapa fungsi kendala yang bisa diperoleh dari 4 petunjuk yang diberikan:

- 1) Harga beli laba-laba Isaac lebih mahal \$30 daripada harga beli spesimen *wolf spider*.
- 2) Laba-laba yang dibeli isaac bukan spesimen *wolf spider*.
- 3) Laba-laba yang dibeli oleh Tim adalah spesimen *ipstin spider*.
- 4) ... dst

Dari fungsi pembatas yang didapatkan tersebut dapat diperiksa apakah tebakan solusi yang dibangkitkan mengarah kepada solusi yang sebenarnya atau tidak.

#### B. Penyelesaian Teka-teki Logika dengan Algoritma Backtracking

Solusi dari teka-teki logika 1 berbentuk tabel dengan 3 kolom dimana permasalahannya adalah bagaimana cara mengisi 2 kolom berikut agar memenuhi semua fungsi kriteria yang diberikan oleh petunjuk.

Tabel 2. Bentuk solusi teka-teki

	Prices	Winners	Spiders
1	\$45	...	...
2	\$60	...	...
3	\$75	...	...
4	\$90	...	...

Berdasarkan langkah-langkah algoritma backtracking yang telah dirancang pada bab dasar teori prosedurnya sebagai berikut:

- Bangkitkan tebakan solusi untuk kolom *wINNERS*
- Periksa apakah tebakan solusi sudah memenuhi fungsi kriteria yang diberikan oleh petunjuk.

1. w1, w2, w3, w4	w1 = Melinda
2. w1, w2, w4, w3	w2 = Tim
3. w1, w3, w2, w4	w3 = Isaac
4. w1, w3, w4, w2	w4 = Jan
5. w1, w4, w2, w3	
6. w1, w4, w3, w2	
7. w2, w1, w3, w4	
8. w2, w1, w4, w3	
9. w2, w3, w1, w4	
10. w2, w3, w4, w1	
11. w2, w4, w1, w3	
12. w2, w4, w3, w1	
13. w3, w1, w2, w4	
14. w3, w1, w4, w2	
15. w3, w2, w1, w4	
16. w3, w2, w4, w1	
17. w3, w4, w1, w2	
18. w3, w4, w2, w1	
19. w4, w1, w2, w3	
20. w4, w1, w3, w2	
21. w4, w2, w1, w3	
22. w4, w2, w3, w1	
23. w4, w3, w1, w2	
24. w4, w3, w2, w1	

- Tebakan no.1 memenuhi seluruh kendala yang diberikan petunjuk. Oleh karena itu tebakan ini disimpan dahulu untuk selanjutnya digabung dengan tebakan solusi untuk kolom *spIDERS*.
- Bangkitkan tebakan solusi untuk kolom *spIDER*. Jika tidak ada tebakan yang dapat dibangkitkan lagi, kembali ke tahap sebelumnya (tebak solusi *wINNERS*).
- Periksa apakah tebakan solusi sudah memenuhi fungsi kriteria yang diberikan oleh petunjuk. Jika tebakan tidak memenuhi kriteria petunjuk, kembali ke tahap sebelumnya. Jika memenuhi kriteria, maka tebakan tersebut adalah solusi finalnya. Untuk tebakan nomor 1 maka solusinya adalah sebagai berikut:

Tabel 3. Perkiraan solusi

	Prices	Winners	Spiders
1	\$45	Melinda	<i>wolf spider</i>
2	\$60	Tim	<i>darkback</i>
3	\$75	Isaac	<i>ipstin spider</i>
4	\$90	Jan	<i>redback</i>

1. s1, s2, s3, s4	s1 = <i>wolf spider</i>
2. s1, s2, s4, s3	s2 = <i>darkback</i>
3. s1, s3, s2, s4	s3 = <i>ipstin spider</i>
4. s1, s3, s4, s2	s4 = <i>redback</i>
5. s1, s4, s2, s3	
6. s1, s4, s3, s2	
7. s2, s1, s3, s4	
8. s2, s1, s4, s3	
9. s2, s3, s1, s4	
10. s2, s3, s4, s1	
11. s2, s4, s1, s3	
12. s2, s4, s3, s1	
13. s3, s1, s2, s4	
14. s3, s1, s4, s2	
15. s3, s2, s1, s4	
16. s3, s2, s4, s1	
17. s3, s4, s1, s2	
18. s3, s4, s2, s1	
19. s4, s1, s2, s3	
20. s4, s1, s3, s2	
21. s4, s2, s1, s3	
22. s4, s2, s3, s1	
23. s4, s3, s1, s2	
24. s4, s3, s2, s1	

Setelah dilakukan pemeriksaan fungsi kriteria, ternyata tabel 3 tidak memenuhi beberapa kriteria. Menurut petunjuk 3, laba-laba yang dibeli Tim adalah spesimen *ipstin spider* bukan *darkback*. Runut-balik ke langkah sebelumnya.

- Setelah melakukan proses di atas secara terus menerus ditemukan solusi dari teka-teki sebagai berikut:

Tabel 3. Perkiraan solusi

	Prices	Winners	Spiders
1	\$45	Melinda	<i>wolf spider</i>
2	\$60	Tim	<i>darkback</i>
3	\$75	Isaac	<i>ipstin spider</i>
4	\$90	Jan	<i>redback</i>

- Berdasarkan strategi ini kompleksitas algoritma untuk kasus paling buruk adalah  $O((m!)^{n-1})$ . Dimana n adalah macam variabel teka-teki (harga laba-laba, *wINNERS*, *spIDERS*) dan m adalah banyaknya anggota tiap variabel.

### C. Analisis Strategi Penyelesaian Teka-teki Logika

Strategi penyelesaian Teka-teki logika menggunakan algoritma backtracking mempunyai keuntungan dibanding dengan metode penyelesaian secara deduksi karena proses pemeriksaan fungsi kriteria dan pembangkitan tebakan solusi adalah permasalahan yang relatif lebih mudah dibandingkan

dengan permasalahan deduksi kasus dan penalaran. Untuk kasus terburuknya algoritma ini sebenarnya sangat mirip dengan algoritma *exhaustive search*. Untuk menghindari hal tersebut kita dapat menambahkan beberapa heuristic yang dapat diambil dari petunjuk yang diberikan teka-teki. Dengan menambahkan *Heuristic* pada algoritma *backtracking*, proses menebak solusi menjadi lebih terarah sehingga tebakan yang harus diperiksa tidak sebanyak algoritma yang telah dijelaskan (<m!).

Sebuah solusi unik pasti dapat ditemukan selama petunjuk yang diberikan oleh teka-teki memadai untuk membuat fungsi-fungsi kriteria.

#### IV. SIMPULAN

Memecahkan teka-teki adalah kegiatan yang menyenangkan sekaligus menantang. Pemecah masalah ditantang untuk berpikir secara kreatif, induktif, dan deduktif untuk menemukan solusi dari teka-teki. Akan tetapi proses berpikir yang sangat melelahkan tersebut dapat dihindari dengan metode penyelesaian secara *backtracking* khususnya untuk teka-teki dengan format seperti yang telah dibahas pada makalah ini. Dengan menggunakan strategi ini artinya manusia juga dapat mengajarkan komputer bagaimana cara menyelesaikan teka-teki. Tentunya ada beberapa usaha yang harus dilakukan sebelum keadaan tersebut dapat dicapai. Akan tetapi ketika manusia sudah bisa mengajarkan komputer menyelesaikan teka-teki tentu komputer dapat memecahkan banyak teka-teki dalam waktu yang relatif lebih singkat dibanding dikerjakan secara manual oleh manusia. Teka-teki Einstein yang sudah disebut di bab 1 dapat diselesaikan dengan algoritma ini tetapi tidak akan diperlihatkan di makalah ini karena keterbatasan halaman dalam penulisan makalah.

#### UCAPAN TERIMA KASIH

Penulis tentunya mengucapkan puji syukur kepada Allah Subhanahu wa ta'ala atas karunia dan rahmatNya penulis dapat menyelesaikan dan mendapatkan ide untuk menyelesaikan makalah tugas IF2211 ini. Kemudian penulis juga mengucapkan terima kasih kepada Ibu Dr. Nur Ulfa Maulidevi ST,M.Sc. sebagai pengajar kelas K2 matakuliah IF2211 dan memperkenalkan penulis betapa menyenangkan mata kuliah Prodi Informatika. Tidak lupa juga penulis mengucapkan terima kasih untuk Bapak Dr. Ir. Rinaldi Munir, MT. atas buku dan diktat yang telah beliau tulis sangat membantu penulis dalam memahami topik di mata kuliah ini. Ucapan terima kasih juga penulis ucapkan untuk rekan-rekan mahasiswa Informatika ITB 2016 dan 2015 atas responnya yang positif terhadap penulis ketika mengambil mata kuliah ini.

#### REFERENSI

- [1] Munir, Rinaldi. 2004. Diktat IF2211 Strategi Algoritma. Bandung.
- [2] Einstein Riddle from : <https://udel.edu/~os/riddle.html> diakses pada tanggal 12 Mei 2018.
- [3] Contoh teka-teki 1 : <http://www.logic-puzzles.org/init.php> diakses pada tanggal 12 Mei 2018.
- [4] Archer Taylor, English Riddles from Oral Tradition (Berkeley: University of California Press, 1951)

#### 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, 13 Mei 2018



Muhammad Isro Fiordi (10216038)