

Penggunaan Algoritma Greedy Dalam Perancangan Papan Teka Teki Silang

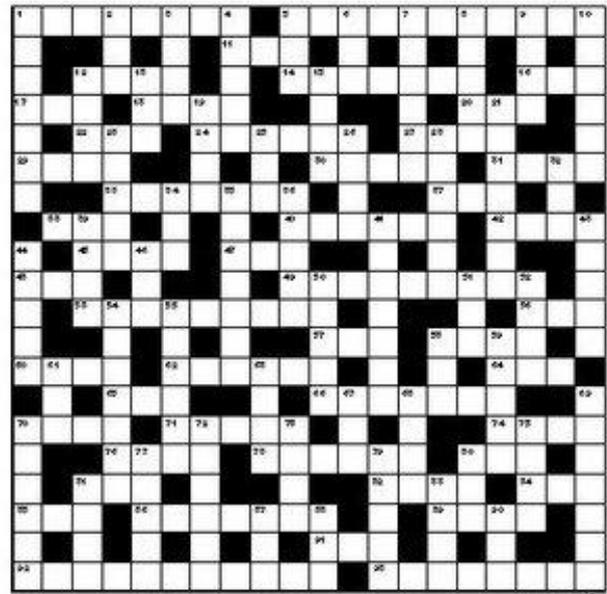
Stefanus Thobi Sinaga / 13510029
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
s.thobi.sinaga@students.itb.ac.id

Abstract— Teka Teki Silang adalah sebuah permainan yang bertujuan untuk mengisi seluruh kotak kosong dengan jawaban berdasarkan petunjuk yang diberikan. Pemain dinyatakan menang apabila dapat mengisi seluruh kotak kosong dengan jawaban yang benar. Ciri-ciri papan permainan Teka Teki Silang yang baik adalah terdapat banyak kotak-kotak putih yang membentuk sebuah loop atau bentuk kotak, sehingga dapat mempermudah pemain dalam menyelesaikan Teka Teki Silang tersebut. Dalam membentuk papan permainan Teka Teki Silang, dibutuhkan peletakan jawaban-jawaban yang baik sehingga tercipta pola yang mendukung pembentukan loop atau bentuk kotak pada papan. Pada makalah ini, penulis mencoba menggunakan Algoritma Greedy untuk menghasilkan sebuah papan permainan Teka Teki Silang dengan menyeleksi kumpulan jawaban sehingga tercipta sebuah papan permainan Teka Teki Silang yang utuh.

Index Terms—Algoritma Greedy, Teka Teki Silang, TTS.

I. PENDAHULUAN

Teka Teki Silang (TTS) adalah sebuah permainan yang muncul pertama kali pada tahun 1913 dalam majalah New York World. Teka Teki Silang tersebut sering disebut sebagai Teka Teki Silang Pertama dengan Arthur Wynne sebagai penemunya. Teka Teki Silang berhasil menarik minat para pembaca hingga akhirnya permainan Teka Teki Silang menjadi fitur mingguan pada majalah New York World. Buku kumpulan Teka Teki Silang pertama terbit pada tahun 1924, diterbitkan oleh Simon dan Schuster. Buku tersebut terbukti laris dan menjadi salah satu benda terpopuler pada tahun 1924. Kesuksesan tersebut diikuti oleh terbitnya buku-buku serupa hingga akhirnya terbit buku “Asah Otak” di Indonesia pada era 70an. Hingga saat ini, teka teki silang masih dapat dengan mudah ditemukan di berbagai majalah atau website di internet. Papan permainan Teka Teki Silang terdiri dari kotak-kotak putih yang saling sambung menyambung dan bercabang untuk nantinya diisi oleh pemain berdasarkan petunjuk-petunjuk yang diberikan.



Gambar 1 Contoh papan permainan Teka Teki Silang

Teka Teki Silang dimainkan dengan mengisi ruang-ruang kosong pada papan(kotak putih) dengan jawaban berdasarkan petunjuk yang diberikan. Terdapat dua jenis ruang kosong pada permainan Teka Teki Silang, yaitu jawaban mendatar (horizontal) dan jawaban menurun (vertikal). Pemain dinyatakan menang apabila dapat mengisi seluruh kotak kosong dengan jawaban yang benar.

II. DASAR TEORI

A. Aturan Permainan Teka Teki Silang

Aturan permainan Teka Teki Silang cukup mudah, pemain hanya perlu mengisi ruang-ruang kosong dengan jawaban sesuai petunjuk. Setiap petunjuk memiliki nomor tersendiri. Misal terdapat petunjuk untuk jawaban mendatar dengan nomor 1, tuliskan jawaban untuk petunjuk tersebut pada kotak nomor 1 dengan ruang mendatar pada papan permainan. Begitu pula untuk jawaban menurun, jika terdapat petunjuk dengan nomor 1, tuliskan jawaban untuk petunjuk tersebut pada kotak nomor dengan ruang menurun pada papan permainan.

Misalnya terdapat petunjuk demikian :

Mendatar :

1. Lawannya kecil
 4. Kondisi air laut di malam hari
- dst.

Menurun

2. Nilai terbesar dalam skala Indeks Prestasi
 3. Rumah Sakit Jiwa
- dst.

Maka untuk petunjuk jawaban mendatar nomor 1, isikan pada kotak dengan nomor 1 dan ruang mendatar. Dalam hal ini jawabannya adalah besar. Saat kita ingin mengisi jawaban untuk petunjuk nomor 2 (jawaban menurun), pada ruang jawaban sudah terdapat 1 huruf dari jawaban petunjuk 1 (dalam hal ini huruf pertama untuk jawaban petunjuk nomor 2), sehingga terdapat dua kemungkinan yaitu,

1. Jawaban untuk petunjuk nomor 2 diawali dengan huruf E
2. Jawaban nomor 1 salah

Jika tidak terdapat jawaban yang benar untuk petunjuk nomor dengan dengan huruf awalan 'E', maka kemungkinan besar jawaban untuk petunjuk nomor 1 salah.

4.



Gambar 2 Contoh papan permainan Teka Teki Silang yang sudah terisi

Akhirnya dengan berdasarkan pada kedua aturan tersebut pemain akan dituntut untuk mencari kombinasi jawaban yang cocok dan dapat mengisi semua ruang jawaban yang telah disediakan di papan.

B. Algoritma Greedy

Algoritma greedy merupakan salah satu metode untuk memecahkan persoalan optimasi. Optimasi adalah sebuah persoalan untuk mencari solusi optimal dari suatu masalah. Algoritma greedy adalah algoritma yang

membentuk solusi langkah per langkah. Pada setiap langkah, terdapat banyak pilihan yang perlu diseleksi. Oleh karena itu, harus dibuat fungsi seleksi yang dapat menghasilkan keputusan terbaik untuk langkah saat itu. Keputusan yang telah diambil pada suatu langkah tidak akan berubah lagi dan tidak mempengaruhi pengambilan keputusan pada langkah-langkah selanjutnya. Dengan algoritma greedy, pada setiap langkah kita memilih optimum-optimum lokal dengan harapan kumpulan optimum lokal tersebut membentuk sebuah solusi optimum global.

Elemen-elemen algoritma greedy:

1. Himpunan kandidat, C
Himpunan kandidat adalah himpunan yang berisi elemen-elemen pembentuk solusi.
2. Himpunan solusi, S
Himpunan solusi adalah himpunan yang berisi kandidat-kandidat yang terpilih sebagai solusi persoalan.
3. Fungsi seleksi (selection function)
Fungsi seleksi adalah fungsi yang memilih kandidat-kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang telah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.
4. Fungsi kelayakan (feasible)
Fungsi ini memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.
5. Fungsi obyektif
Fungsi ini memaksimumkan atau meminimumkan nilai solusi.

Dengan kata lain algoritma greedy melibatkan pencarian sebuah himpunan bagian dari himpunan kandidat, dimana yang dalam hal ini, himpunan solusi harus memenuhi beberapa kriteria yang ditentukan, yaitu menyatakan suatu solusi dan himpunan solusi dioptimasi oleh fungsi obyektif. Adapun psuedo-code dari algoritma greedy adalah sebagai berikut:

```
function greedy(input C:
himpunan_kandidat) → himpunan_kandidat
{ Mengembalikan solusi dari persoalan
optimasi dengan algoritma greedy
Masukan : himpunan_kandidat C
Keluaran: himpunan_solusi yang
bertipe himpunan_kandidat
}
Deklarasi
x : kandidat
```

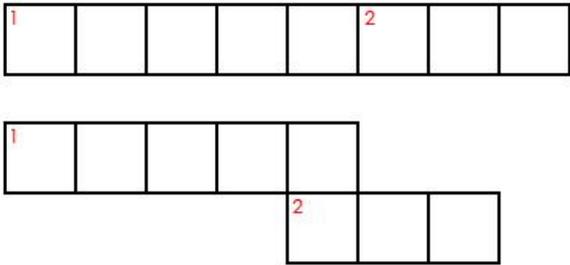
```

S : himpunan_kandidat
Algoritma:
S ← {} {inisialisasi S dengan kosong}
while (not SOLUSI(S)) and (C !={} ) do
  x ← SELEKSI(C) { pilih sebuah kandidat dari C}
  C → C - {x} { elemen himpunan kandidat berkurang satu }
  if LAYAK(S U{x}) then
    S ← S U {x}
  endif
endwhile
{SOLUSI(S) or C = {} }
if SOLUSI(S) then
  return S
else
  write('Tidak ada solusi')
endif

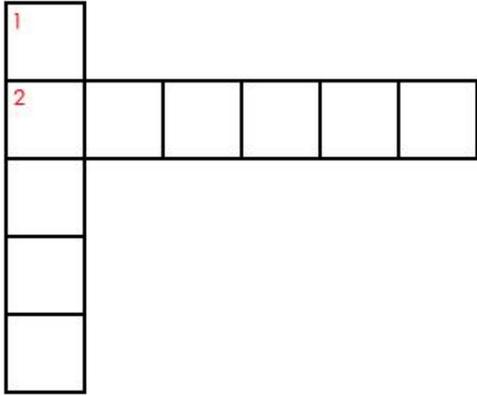
```

III. APLIKASI

Dalam merancang papan permainan Teka Teki Silang, kita harus memanfaatkan kumpulan jawaban dari berbagai petunjuk sehingga dapat menghasilkan kombinasi jawaban terbaik yang membentuk pola yang baik dan menarik. Hal pertama yang harus diperhatikan dalam merancang papan permainan Teka Teki Silang yaitu dalam menempelkan satu jawaban dengan jawaban lainnya, kategori dua jawaban tersebut harus berbeda. Maksudnya adalah jika kita ingin menempelkan jawaban mendatar, kita harus mencari jawaban menurun yang cocok dengan jawaban mendatar tersebut. Kita tidak dapat menempelkan jawaban mendatar dengan jawaban mendatar lainnya, karena hal tersebut akan membingungkan pemain dengan jumlah kotak pada suatu ruang mendatar. Begitu pula dengan jawaban menurun, kita harus mencari jawaban mendatar yang cocok dengan jawaban menurun tersebut.

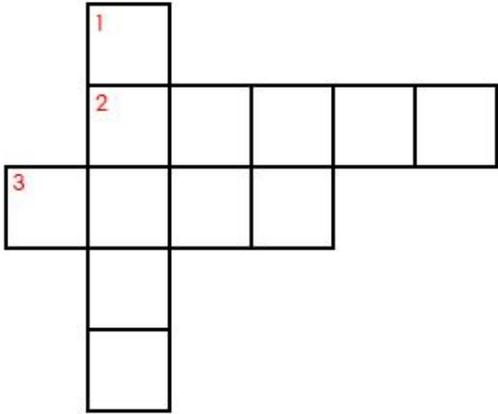


Gambar 3 Contoh penempelan yang salah

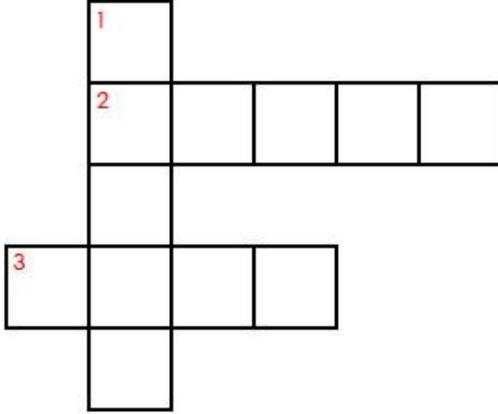


Gambar 4 Contoh penempelan yang benar

Hal lain yang harus diperhatikan adalah tidak boleh ada 4 kotak yang saling berhubungan satu sama lain, karena hal ini dapat menimbulkan kebingungan pada pemain, sehingga dalam menempelkan ke satu jawaban ke jawaban lain, perlu diperhatikan apakah dengan penempelan kedua jawaban tersebut akan menghasilkan pola 4 kotak yang telah disebutkan tadi. Untuk lebih jelasnya, perhatikan gambar dibawah.



Gambar 5 Contoh penempelan yang salah



Gambar 6 Contoh penempelan yang benar

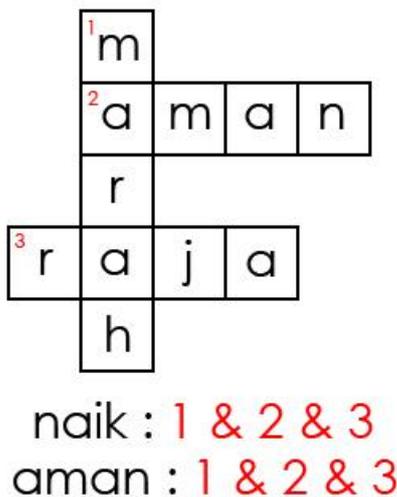
Greedy berdasarkan kecocokan terbesar

Algoritma ini mengutamakan pembuatan papan permainan Teka Teki Silang yang memiliki banyak loop atau bentuk kotak dengan memilih jawaban-jawaban dengan kecocokan tertinggi berdasarkan kombinasi jawaban yang sudah ada saat ini. Dalam menggunakan algoritma ini, kita akan memprioritaskan hal-hal berikut :

1. Jawaban yang dapat menghasilkan pola loop atau bentuk kotak pada papan permainan.
2. Jawaban yang dapat ditempel ke jawaban yang sudah ada dimasukkan ke dalam papan permainan.

Sehingga elemen-elemen algoritma greedy yang digunakan adalah sebagai berikut :

- Himpunan kandidat : Himpunan jawaban-jawaban dari berbagai petunjuk.
- Himpunan solusi : Himpunan jawaban-jawaban yang telah diseleksi, terbukti memiliki kecocokan, dan dapat membentuk sebuah papan permainan Teka Teki Silang.
- Fungsi Seleksi : pilihlah jawaban dengan nilai kecocokan tertinggi. Nilai kecocokan disini berarti terdapat alphabet pada jawaban dari himpunan kandidat yang sama dengan alphabet pada jawaban dari himpunan solusi. Jika terdapat alphabet yang sama dari satu jawaban dengan satu jawaban lainnya, maka nilai kecocokan bertambah satu. Hal tersebut adalah syarat perlu untuk menempelkan satu jawaban dengan jawaban lainnya. Namun nilai kecocokan dari suatu jawaban bisa lebih besar dari satu jika jawaban tersebut memiliki alphabet lain yang cocok dengan alphabet pada jawaban lainnya dari himpunan solusi dengan syarat ketiga jawaban tersebut harus bisa saling menempel. Arti dari saling menempel disini adalah saat direpresentasikan ke dalam bentuk kotak dalam papan (misal menggunakan array 2 dimensi, dengan kotak sebagai elemen-elemen pada array), kotak-kotak dari kandidat jawaban melewati kotak dari dua jawaban pada kandidat solusi. Untuk lebih jelasnya perhatikan gambar dibawah.



Gambar 7 Ilustrasi Fungsi Kelayakan

Pada gambar diatas, kata naik memiliki kecocokan secara alphabet dengan jawaban 1,2,dan 3 tetapi tidak bisa ditempel ke dalam papan permainan, sedangkan kata aman dapat ditempel ke dalam papan permainan dengan menempelkannya ke jawaban 2 dan 3 (nilai kecocokan 2), sehingga kata aman lah yang lolos fungsi Seleksi, sedangkan kata naik tidak. Jika terdapat lebih dari satu jawaban dengan nilai kecocokan yang sama, pilih salah satu secara acak.

- Fungsi Kelayakan : memeriksa apakah jumlah jawaban pada himpunan solusi saat ini masih cukup untuk menampung satu jawaban lagi, dan memeriksa apakah sudah ada jawaban yang sama pada himpunan solusi saat ini.
- Fungsi Objektif : tempelkan kandidat jawaban dengan jawaban lain pada himpunan solusi

Dengan menggunakan elemen-elemen di atas, maka pseudo-code dari algoritma greedy berdasarkan kecocokan terbesar adalah sebagai berikut

```
function greedyKecocokan (input C  
: himpunan_jawaban, input nJawaban :  
integer) -> himpunan_jawaban
```

```
{ mengembalikan kombinasi jawaban  
yang dapat membentuk sebuah papan  
permainan Teka Teki Silang dari  
kumpulan jawaban pada  
himpunan_jawaban menggunakan  
algoritma greedy  
Masukan : himpunan jawaban C,  
jumlah jawaban yang  
diperlukan  
Keluaran : himpunan solusi yang  
bertipe himpunan_jawaban  
}
```

Deklarasi

```
S : himpunan_jawaban  
X : kandidat  
n : integer
```

Algoritma

```
S <- {C[random()]} {jawaban awal}  
n = 1  
while ((n < nJawaban) and (C <> {}))  
do  
  X <- Seleksi(C)  
  C <- C - {X}  
  if Layak(X) then  
    S <- S U {X}  
    n++  
  endif  
endwhile  
if (n > 1) {terdapat solusi}  
  return S  
else  
  printf("Tidak ada solusi")
```

IV. HASIL DAN PEMBAHASAN

Setelah melakukan ujicoba terhadap algoritma yang dibuat, ternyata solusi yang dihasilkan oleh program sangat bergantung dengan himpunan jawaban yang ada. Himpunan jawaban dengan jumlah elemen yang sangat besar membuat algoritma ini menjadi efektif. Hal ini disebabkan oleh pilihan jawaban yang besar menyebabkan kemungkinan untuk mendapatkan kandidat jawaban dengan kecocokan lebih besar dari satu menjadi lebih tinggi. Oleh karena itu kombinasi jawaban sangat terpengaruh oleh besarnya himpunan jawaban.

Pengujian yang berulang-ulang juga membuktikan bahwa jawaban pertama yang dipilih mempengaruhi pola papan permainan Teka Teki Silang yang dihasilkan. Sehingga algoritma ini sangat baik untuk menghasilkan banyak papan permainan Teka Teki Silang dari satu himpunan jawaban yang sama, dengan catatan elemen himpunan jawaban cukup banyak untuk menghasilkan banyak papan permainan Teka Teki Silang.

V. KESIMPULAN DAN SARAN

Dari pembahasan di atas, dapat disimpulkan :

1. Algoritma greedy dapat digunakan untuk merancang papan permainan Teka Teki Silang dari kumpulan berbagai jawaban.
2. Penggunaan algoritma greedy untuk merancang papan permainan Teka Teki Silang dapat menghasilkan solusi yang optimal.

Adapun saran untuk pengembangan selanjutnya adalah sebagai berikut :

1. Algoritma greedy yang digunakan pada makalah ini hanya berdasarkan kecocokan kandidat jawaban dengan jawaban pada papan permainan, sedangkan panjang string jawaban yang dipilih dapat membantu perancangan papan permainan yang baik, karena semakin panjang string maka kemungkinan kecocokan alphabet antar jawaban akan semakin membesar. Oleh karena itu panjang string juga sebaiknya digunakan sebagai parameter penyeleksian kandidat solusi.

DAFTAR PUSTAKA

- [1] Rinaldi Munir, "Diktat Kuliah IF3051 Strategi Algoritma", Institut Teknologi Bandung, 2009, hal 26–68.
- [2] <http://ttsmania.com/index.php?go=tekatekisilang>
Waktu akses : 21 Desember 11.40
- [3] <http://rengkodriders.files.wordpress.com/2012/03/tts.jpg>
Waktu akses : 21 Desember 11.40
- [4] <http://ngerumpy.blogspot.com/2011/02/sejarah-dan-asal-usul-teka-teki-silang.html>
Waktu akses : 21 Desember 11.40

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, 21 Desember 2012



Stefanus Thobi Sinaga
13510029