

Penerapan Algoritma Brute Force (Exhaustive Search) dalam Permainan Scrabble

Aidil Syaputra

Program Studi Teknik Informatika

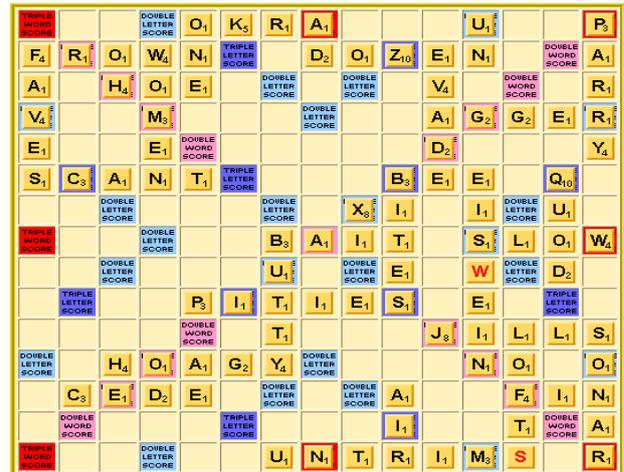
Sekolah Teknik Elektro dan Informatika

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

aidilisyaputra@itb.ac.id dan 13510105@std.stei.itb.ac.id

Abstrak—Scrabble sudah menjadi permainan otak-atik kata yang paling dikenal di seluruh dunia. Permainan ini sigemari oleh banyak kalangan, mulai dari anak kecil hingga orang dewasa. Bahkan, sekarang ini permainan scrabble ii sudah menjadi suaru prtandingan. Di samping menguji kemampuan penguasaan kosakata dalam bahasa Inggris, permainan ini juga menguji strategi untuk mendapatkan nilai berlipat. Permainan Scrabble dimainkan oleh 2 atau 4 orang misinya adalah untuk mengumpulkan poin berdasarkan nilai kata yang tertulis dari setiap huruf yang dapat dilihat di atas papan game scrabble tersebut. Pada makalah ini, dibahas penggunaan algoritma brute force dalam permainan sehingga dapat menghasilkan nilai maksimum.

Kata kunci—scrabble, brute force, exhaustive search, heuristic.



Gambar 1 Scrabble

I. PENDAHULUAN

Scrabble adalah permainan papan dan permainan menyusun kata yang dimainkan 2 atau 4 orang yang mengumpulkan poin berdasarkan nilai kata yang dibentuk dari keping huruf di atas papan permainan berkotak-kotak (15 kolom dan 15 baris).

Biji permainan berupa keping berbentuk bujur sangkar yang bertuliskan huruf pada salah satu sisi. Pemain mengambil hingga sebanyak tujuh buah keping huruf dari kantong, dan berusaha menyusun kata secara mendatar atau menurun seperti teka-teki silang. Kata-kata yang dibuat harus merupakan kata yang diizinkan untuk dimainkan berdasarkan kamus standar sesuai dengan bahasa yang dimainkan. Pemain yang mengumpulkan total poin tertinggi dinyatakan sebagai pemenang.



Gambar 2 Huruf-Huruf Scrabble

1.1 Sejarah

Permainan Scrabble mulanya diciptakan pada tahun 1938 dengan nama "Criss-Crosswords" oleh seorang arsitek bernama Alfred Mosher Butts. Permainan ini merupakan penyempurnaan dari permainan Lexiko yang lebih dulu diciptakannya, tapi dilengkapi papan permainan dan cara bermain seperti teka-teki silang. Permainan tetap menggunakan keping huruf seperti

Lexiko yang dibuat berdasarkan hasil perhitungan distribusi frekuensi penggunaan huruf berbagai tulisan berbahasa Inggris, termasuk artikel surat kabar *The New York Times*. Alfred Butts memproduksi sendiri permainan ini dan menawarkannya ke berbagai perusahaan mainan besar tapi tidak laku.

Pada tahun 1948, pengacara James Brunot, penduduk asal Newtown, Connecticut membeli hak memproduksi permainan "Criss-Crosswords" dengan janji memberi Butts royalti dari setiap unit yang terjual. Sebagai pemilik yang baru, Brunot mengganti nama permainan menjadi "Scrabble", sebuah kata dalam bahasa Inggris yang berarti "berjuang membanting tulang". Brunot hanya sedikit mengubah kotak-kotak "bonus" pada papan permainan, dan menyederhanakan aturan permainan. Permainan Scrabble ternyata laku dijual. Salah satu pembelinya adalah toko serba ada Macy's yang berhasil mengangkat Scrabble menjadi mainan yang ingin dibeli konsumen.

1.2 Aturan Main

Permainan dimainkan 2 hingga 4 orang dengan papan permainan berkotak-kotak yang terdiri dari 15 kolom dan 15 baris. Sewaktu menyusun kata, setiap kotak diisi dengan satu keping huruf. Di klub resmi atau turnamen, pertandingan hanya dimainkan dua orang pemain (atau, kadang-kadang antara 2 tim yang masing-masing anggotanya bekerja sama menyusun huruf dari rak yang sama).

Setiap huruf memiliki nilai tertentu (antara 1 sampai 10) yang bergantung pada frekuensi kemunculan huruf tersebut pada tulisan yang standar. Pada edisi Scrabble bahasa Inggris, huruf-huruf yang biasa muncul dalam kosakata bahasa Inggris, seperti huruf "E" atau "O" hanya bernilai 1 poin. Huruf-huruf yang jarang muncul, seperti "Q" dan "Z" masing-masing bernilai 10 poin. Selain itu, terdapat dua keping kosong (tidak ditulisi huruf) yang bernilai poin nol, tapi bisa dipakai untuk melambangkan semua huruf mulai dari "A" sampai "Z".

Papan permainan ditandai dengan kotak-kotak "bonus" yang melipatgandakan jumlah poin. Kotak berwarna merah bertuliskan "triple-word" mengalikan 3 total poin yang didapat dari sebuah kata; kotak berwarna merah jambu "double-word" mengalikan 2 total poin yang didapat dari sebuah kata; kotak biru tua "triple-letter" mengalikan 3 nilai huruf yang diletakkan di atasnya, dan kotak biru muda "double-letter" mengalikan 2 nilai huruf yang diletakkan di atasnya. Kotak di bagian tengah papan permainan (H8) bernilai "double-word" dan diberi tanda bintang atau logo.

1.3 Strategi dan Taktik

Tujuan permainan adalah mengumpulkan poin sebanyak-banyaknya untuk menang. Pemain perlu terampil mengenali kata-kata yang boleh atau dilarang

dipakai, dan menyusunnya dari serangkaian huruf-huruf yang tidak beraturan. Sebagian besar peserta turnamen mempelajari kosakata secara serius, dan berlatih menyusun kata-kata dari alfagram atau susunan huruf-huruf yang tidak beraturan. Kata-kata seperti *ZYZZYVA* tidak ada gunanya dihafalkan, tapi pemain Scrabble bahasa Inggris sangat perlu menghafal kata-kata yang disusun dari huruf-huruf yang umum (misalnya *ATRESIA*).

Pemain Scrabble perlu menghafal kata-kata yang terdiri dari 2 huruf, karena bisa diletakkan paralel dengan kata yang sudah ada. Kata yang diletakkan paralel sering menghasilkan lebih banyak poin daripada membuat kata yang bersilangan, atau memanjangkan kata yang sudah ada. Setelah menguasai kata-kata yang terdiri dari 2 huruf, pemain pemula bisa mulai belajar kata-kata pendek yang tersusun dari huruf-huruf bernilai tinggi, misalnya *QAT*, *ZEK*, dan *JEUX*. Kata-kata langka yang hanya dipahami orang tertentu tidak juga menghasilkan poin lebih besar dari kata-kata yang umum. Kata *FAERIE* tergantung pada posisinya di papan permainan, justru menghasilkan poin lebih sedikit dibandingkan kata *FAIRY*.

Huruf-huruf yang bernilai 4 poin atau lebih sedapat mungkin dimainkan pada kotak bonus. Selain itu, huruf-huruf seperti X, H, dan Y bisa mendongkrak perolehan poin hingga 4 atau 6 kali lipat dari nilai yang tertera bila dipakai sebagai titik temu dua buah kata. Huruf vokal yang berada di samping kotak *double letter* atau *triple letter* merupakan "sasaran empuk", karena bisa menghasilkan banyak poin bila di atas kotak bonus diletakkan konsonan bernilai poin tinggi. Pemain tingkat menengah bisa menghafal semua kata-kata yang menggunakan huruf "sakti" (K, J, Q, Z, dan X) yang panjangnya 5 huruf atau kurang.

Pengaturan komposisi huruf sering terlewatkan oleh pemain tingkat pemula. Di atas rak sebaiknya tidak perlu ada huruf duplikat yang terlalu banyak, atau proporsi huruf vokal dan konsonan yang tidak seimbang. Rangkaian huruf-huruf AADIIKR bisa membentuk kata *DARK*, tapi semua konsonan habis dipakai dan hanya menyisakan 2 keping huruf "I". Pada permainan Scrabble, keping huruf vokal lebih banyak daripada konsonan, sehingga rangkaian huruf yang sulit dimainkan (seperti AIIIEUAO) mungkin saja didapati sewaktu mengambil huruf dari kantong.

Strategi yang tidak kalah pentingnya adalah menahan diri untuk tidak membuat kata yang gampang diubah menjadi kata baru oleh lawan dengan cuma menambahkan satu atau dua huruf. Kata *QUIT* bisa memberi bonus 14 poin bagi lawan yang meletakkan huruf "E" dan mengubahnya menjadi *QUITE*. Pemain yang berpengalaman juga biasanya meletakkan konsonan di kotak yang bertetangga dengan kotak bonus. Bila

huruf "Q" belum dimainkan, huruf "U" juga berbahaya bila diletakkan bersebelahan dengan kotak bonus.

1. 4 Scrabble dalam Komputer

Scrabble sering menjadi objek penelitian peneliti dan peminat kecerdasan buatan. Semata-mata bermain dengan menyusun kata-kata bernilai tinggi tidak selalu merupakan strategi terbaik, sehingga komputer perlu diajar dengan sejumlah strategi yang lebih cerdas agar lebih pintar.

Maven adalah nama program komputer untuk bermain Scrabble karya Brian Sheppard yang menerapkan kecerdasan buatan. Salah satu versi program Maven digunakan pada versi resmi permainan komputer Scrabble keluaran Atari di Amerika Utara. Maven juga digunakan pada perangkat lunak resmi Scrabble yang dijual perusahaan Funkitron, dan Ubisoft untuk peredaran di luar Amerika Utara. Selain itu, versi sumber terbuka permainan komputer Scrabble juga tersedia dengan nama Quackle.

II. DASAR TEORI

2.1 Algoritma Brute Force

Brute force adalah sebuah pendekatan yang lempang (straightforward) untuk memecahkan suatu masalah (problem statement) dan definisi konsep yang melibatkan Algoritma brute force memecahkan masalah dengan sangat sederhana, langsung, dan dengan cara yang jelas (obvious way) meskipun bukan merupakan solusi yang paling mangkus.

Karakteristik Algoritma Brute Force:

1. Algoritma Brute Force umumnya tidak "cerdas" dan tidak mangkus karena ia membutuhkan jumlah langkah yang besar dalam penyelesaiannya. Kadang pula algoritma Brute Force disebut juga algoritma naif (naïve algorithm).
2. Algoritma Brute Force lebih cocok untuk masalah yang berukuran kecil.
3. Meskipun bukan metode yang mangkus, hampir semua masalah dapat diselesaikan dengan algoritma Brute Force.

Kekuatan dan Kelemahan Metode Brute Force

Kekuatan:

1. Metode brute force dapat digunakan untuk memecahkan hampir sebagian besar masalah (wide applicability).
2. Metode brute force sederhana dan mudah dimengerti
3. Metode brute force menghasilkan algoritma yang layak untuk beberapa masalah penting

seperti pencarian, pengurutan, pencocokan string, perkalian matriks.

4. Metode brute force menghasilkan algoritma baku (standard) untuk tugas-tugas komputasi seperti penjumlahan/perkalian n buah bilangan, menentukan elemen minimum atau maksimum di dalam tabel (list).

Kelemahan:

1. Metode brute force jarang menghasilkan algoritma yang mangkus.
2. Beberapa algoritma brute force lambat sehingga tidak dapat diterima.
3. Tidak sekonstruktif/sekreatif teknik pemecahan masalah lainnya.

2.2 Exhaustive Search

Exhaustive search adalah teknik pencarian solusi secara solusi *brute force* untuk masalah-masalah kombinatorik. Biasanya di antara objek-objek kombinatorik seperti permutasi, kombinasi, atau himpunan bagian dari sebuah himpunan.

Langkah-langkah metode exhaustive search:

1. Enumerasi (*list*) setiap solusi yang mungkin dengan cara yang sistematis.
2. Evaluasi setiap kemungkinan solusi satu per satu, simpan solusi terbaik yang ditemukan sampai sejauh ini (*the best solusi found so far*).
3. Bila pencarian berakhir, umumkan solusi terbaik (*the winner*).

Meskipun algoritma exhaustive secara teoritis menghasilkan solusi, namun waktu atau sumberdaya yang dibutuhkan dalam pencarian solusinya sangat besar.

2.3 Teknik Heuristik untuk Mempercepat Algoritma Exhaustive Search

Algoritma *exhaustive search* dapat diperbaiki kinerjanya sehingga tidak perlu melakukan pencarian terhadap semua kemungkinan solusi. Salah satu teknik yang digunakan untuk mempercepat pencarian solusi adalah teknik heuristik (heuristic). Teknik heuristik digunakan untuk mengeliminasi beberapa kemungkinan solusi tanpa harus mengeksplorasinya secara penuh. Selain itu, teknik heuristik juga membantu memutuskan kemungkinan solusi mana yang pertama kali perlu dievaluasi.

Contoh penggunaan heuristik untuk mempercepat algoritma exhaustive search

Permasalahan *anagram*. *Anagram* adalah penukaran huruf dalam sebuah kata atau kalimat sehingga kata atau kalimat yang baru mempunyai arti lain.

Contoh-contoh *anagram*:

lived → *devil*

tea → *eat*

charm → *march*

Bila diselesaikan secara exhaustive search, kita harus mencari semua permutasi huruf-huruf pembentuk kata atau kalimat, lalu memeriksa apakah kata atau kalimat yang terbentuk mengandung arti.

Teknik heuristik dapat digunakan untuk mengurangi jumlah pencarian solusi. Salah satu teknik heuristik yang digunakan misalnya membuat aturan bahwa dalam Bahasa Inggris huruf *c* dan *h* selalu digunakan berdampingan sebagai *ch* (lihat contoh *charm* dan *march*), sehingga kita hanya membuat permutasi huruf-huruf dengan *c* dan *h* berdampingan. Semua permutasi dengan huruf *c* dan *h* tidak berdampingan ditolak dari pencarian.

Keuntungan penerapan teknik heuristik dalam pencarian solusi:

1. Heuristic search memiliki fleksibilitas tinggi yang memungkinkan untuk digunakan pada masalah yang kompleks dan tidak terstruktur.
2. Blind search menjamin ditemukannya solusi optimal, tetapi kurang layak untuk digunakan dalam komputasi karena kebutuhan memory (dan waktu) yang sangat besar.
3. Metode heuristic lebih sederhana untuk dipahami oleh pengambil keputusan, secara khusus apabila didukung oleh analisis kualitatif.
4. Suatu metode heuristic dapat digunakan sebagai bagian dari prosedur iteratif yang tetap menjamin ditemukannya sebuah solusi optimal.

III. PENERAPAN ALGORITMA BRUTE FORCE PADA PERMAINAN SCRABBLE

Pada permainan scrabble ini bertujuan memperoleh poin terbanyak dibandingkan lawan. Untuk memperoleh poin tersebut, pemain harus membuat kata yang ada dalam bahasa inggris dengan menggunakan huruf-huruf yang ada di papan scrabble dan huruf-huruf yang ada dalam kantong pemain itu sendiri.

Algoritma exhaustive search ini adalah membantu pemain dalam menemukan kata-kata apa saja yang dapat dibuat oleh pemain tersebut dengan menggunakan huruf-huruf yang ada di papan scrabble dan huruf-huruf yang ada dalam kantong pemain itu sendiri.

Misalkan dalam papan scrabble tersebut terdapat kata "SMART" dan pada huruf kita terdapat huruf-huruf berikut : O, D, L, O, C, A, H
Maka untuk itu dengan memanfaatkan huruf-huruf tersebut kita dapat membuat kata-kata.

Berikut pseudocode untuk menemukan kata yang mempunyai score tertinggi :

```
Function Solusi (T : array of char) → String
{
  Fungsi ini memanfaatkan algoritma exhaustive search
  untuk mencari semua kemungkinan kata yang valid dari
  huruf-huruf yang tersedia.
}
```

Deklarasi
L : array of string
K : String;
N : int

```
Algoritma
L = ListSolusi(T);
N = 0;
K = null;
For (int i = 0; i < L.length(); i++)
  If (IsValid(L[i]) and Score(L[i]) > N)
    K = L[i];
    N = Score(L[i]);
  Endif
Endfor
→ K
```

```
Function ListSolusi(T : array of char) → array of string
{
  Fungsi ini membuat kata-kata yang mungkin dengan
  mempermutasikan setiap huruf yang ada pada array T.
  Kombinasi dimulai dengan 2 huruf hingga kata dengan
  panjang huruf yang sama dengan banyak huruf pada
  array T
}
```

```
Function Score(S :String) → int
{
  Fungsi ini menghitung total score yang diperoleh dari
  kata yang dimasukkan. Score diperoleh dengan
  menjumlahkan nilai yang ada pada setiap huruf dalam
  kata tersebut.
}
```

```
Function IsValid(S :String) → boolean
{
  Fungsi ini mengecek apakah kata yang dimasukkan
  tersebut terdapat dalam kamus bahasa inggris. Jika ada
  maka fungsi mengembalikan nilai true dan jika tidak ada
  fungsi mengembalikan nilai false.
}
```

Dari fungsi tersebut akan didapatkan solusi yang memiliki score tertinggi. Akan tetapi, kompleksitas waktunya sangat besar.

Kita dapat mempercepatnya dengan memanfaatkan teknik heuristik. Misalnya kita mempunyai huruf

C,O,L,H,O., dan S. Disana kita mempunyai 6 item yang digunakan untuk kombinasi. Dengan menggunakan teknik heuristik kita dapat menjadikannya 5 item, dimana huruf C,H dijadikan 1 item yaitu CH. Karena dalam bahasa inggris huruf H sering diawali dengan huruf C. Dengan memanfaatkan teknik ini kompleksitasnya waktunya akan berkurang. Tetapi hal ini dapat menyebabkan solusi yang didapat belum tentu optimal.

Menggunakan algoritma exhaustive search dalam penyelesaian permainan scrabble masih memiliki banyak kekurangan. Dengan menggunakan algoritma exhaustive search kita tidak mempertimbangkan giliran selanjutnya. Kita hanya fokus pada giliran sekarang saja. Terdapat kemungkinan karena telah banyak menggunakan huruf vokal pada giliran sekarang, sehingga pada giliran berikutnya kita tidak dapat membentuk satu kata pun untuk membentuk kata yang bermakna.

Oleh karena itu, untuk penelitian lebih lanjut bisa digunakan algoritma lain yang dapat menyelesaikan permainan scrabble dengan lebih efektif lagi dan dapat mempertimbangkan giliran-giliran berikutnya.

IV. KESIMPULAN DAN SARAN

Kesimpulan yang dapat diambil dari pembahasan di atas antara lain:

1. Algoritma Brute Force cocok untuk menyelesaikan permainan scrabble.
2. Algoritma Brute Force ini merupakan algoritma yang mampu menyelesaikan masalah dengan cara sederhana, langsung, dan jelas meskipun bukanlah algoritma yang mangkus.
3. Dengan memanfaatkan teknik heuristik pencarian solusi akan lebih cepat lagi.
4. Untuk jumlah karakter (huruf) yang semakin besar, pemrosesan pencarian jawaban semakin lambat sehingga dibutuhkan algoritma yang lebih mangkus lagi.

Saran untuk pengembangan selanjutnya:

1. Diperlukan penelitian lebih lanjut dengan menggunakan algoritma lainnya sehingga pemrosesan pencarian jawaban dalam permainan scrabble ini dapat lebih efektif lagi dan bisa mempertimbangkan giliran-giliran selanjutnya.

V. ACKNOWLEDGMENT

Ucapan terima kasih saya ucapkan kepada Pak Rinaldi Munir atas dukungan, bimbingan dan berbagai referensi-referensi yang dapat digunakan untuk penyelesaian makalah ini. Terima kasih juga saya ucapkan kepada teman-teman prodi IF maupun STI yang telah bersedia untuk berdiskusi terhadap berbagai permasalahan dalam makalah ini.

REFERENCES

- [1] Munir, Rinaldi, *Diktat Kuliah IF3051 Strategi Algoritma*, Program Studi Teknik Informatika ITB, 2009, hal 8-19
- [2] <http://www.msoworld.com/mindzine/news/proprietary/scrabble/features/history.html>
Tanggal akses : 20 Desember 2012
- [3] <http://paman-guru.blogspot.com/2012/02/download-game-scrabble-gratis-di.html>
Tanggal akses : 20 Desember 2012
- [4] <http://menatap-ilmu.blogspot.com/2011/07/download-game-scrabble-gratis.html>
Tanggal akses : 20 Desember 2012

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



Aidil Syaputra (13510105)