

Penerapan Algoritma *Brute Force* pada permainan Countdown Number

Farhan Amin (13515043)

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13515043@std.stei.itb.ac.id

Abstract— permainan Countdown Number adalah permainan dari *British game show* yang menggunakan angka sebagai *puzzle*. Dalam permainan ini diberikan pilihan angka (*goal number*) yang harus disusun dari 6 buah nomor yang dipilih. Pada makalah ini, akan dibahas strategi Algoritma yakni implementasi algoritma *Brute force* pada permainan Countdown Number

Keywords—permainan Countdown Number; Brute Force; puzzle

I. PENDAHULUAN

A. Latar Belakang

Permainan Countdown Number adalah permainan yang berasal dari *Gameshow* di *British*. Permainan ini menggunakan angka sebagai *puzzle*. Permainan Countdown Number ini adalah permainan aritmatika dengan tujuan mencari cara untuk mengubah 6 buah angka yang dipilih secara acak menjadi *goal number* yang juga dipilih secara acak.

Pada awal *gameshow*, peserta diberikan kesempatan untuk memilih 6 angka secara acak yang nantinya digunakan untuk membentuk *goal number*, yang ditentukan secara acak melalui *computer*, dengan menggunakan operator-operator matematika dasar seperti pertambahan, pengurangan, perkalian, dan pembagian.



Gambar 1. Logo *Gameshow Countdown*

B. Tujuan

Tujuan penulis membuat makalah tentang topik ini adalah:

- Menambah pemahaman tentang algoritma *Brute Force*
- Berusaha mengimplementasikan algoritma *Brute Force* untuk permainan *Countdown Number*

II. LANDASAN TEORI

A. Algoritma Brute Force

Algoritma *Brute Force* adalah algoritma yang memecahkan masalah dengan pendekatan yang lempang (*straight-forward*) atau memecahkan persoalan dengan cara yang sederhana, langsung, dan jelas atau disebut juga *Just Do It!* Atau *Just Solve It!*. Algoritma *Brute Force* biasanya didasarkan pada pernyataan pada persoalan (*problem statement*) atau berdasarkan definisi konsep yang dilibatkan.

Ada beberapa contoh penerapan algoritma *Brute Force* berdasarkan pernyataan persoalan :

1. Pencarian elemen dengan nilai Ekstrim.

pada kasus ini kita diharuskan untuk mencari nilai Ekstrim (minimum atau maksimum) dari sebuah senarai. Dengan algoritma *Brute Force*, permasalahan ini diselesaikan dengan cara yang sederhana yakni membandingkan setiap bilangan yang ada di senarai tersebut sampai didapatkan nilai Ekstrim (minimum atau maksimum) yang diinginkan

2. Pencarian beruntun (*Sequential Search*)

Pada kasus ini kita diharuskan untuk mencari sebuah elemen, X, pada sebuah senarai. Dengan algoritma *Brute Force*, permasalahan ini diselesaikan dengan cara yang sederhana yakni setiap elemen pada senarai dibandingkan dengan X. Pencarian selesai jika X ditemukan atau elemen senarai sudah habis diperiksa.

3. Tes bilangan prima

Pada kasus ini kita diharuskan untuk menguji sebuah bilangan bulat positif yang diberikan (n) adalah bilangan prima atau bukan. Dengan algoritma *Brute Force*, permasalahan ini diselesaikan dengan cara membagi n dengan 2 sampai \sqrt{n} . lalu melakukan perulangan dari 2 sampai \sqrt{n} , Jika semua angka di *range* itu tidak habis membagi n , maka n adalah bilangan prima.

4. Pattern Matching

Pada kasus ini kita diharuskan melakukan pengecekan ada atau tidaknya suatu pola yang dicari pada sebuah kumpulan pola yang ada. Umumnya, *pattern matching* yang diselesaikan dengan *brute force* adalah string *matching*. Pada algoritma ini akan membandingkan pola (*pattern*) dan mencarinya di teks dari awal hingga akhir teks hingga akhirnya ditemukan bagian pada teks yang sesuai dengan pola atau tidak ditemukan.

5. Exhaustive Search

Pada kasus ini kita diharuskan untuk menjabarkan semua solusi yang mungkin bisa dijadikan solusi untuk sebuah permasalahan dan kemudian setelah sudah dihasilkan solusi-solusi yang bisa dijadikan solusi pada permasalahan tersebut akan dipilih solusi yang terbaik.

6. Pengurutan bilangan

Pada kasus ini, kita diharuskan untuk melakukan pengurutan bilangan yang terdapat pada senarai, pengurutan bisa dilakukan menaik atau menurun. Permasalahan ini diselesaikan dengan membandingkan setiap elemen dengan elemen lain sesuai dengan keinginan pengurutan (menaik atau menurun), proses perbandingan ini dilakukan untuk semua elemen pada senarai

3. Meskipun bukan metode yang mangkus, hampir semua persoalan dapat diselesaikan dengan algoritma *brute force*.

Kekuatan dari metode *brute force*:

1. Hampir semua masalah dapat dipecahkan dengan menggunakan metode *brute force* (*wide applicability*).
2. metode yang mudah untuk dimengerti karena berupa metode yang sederhana.
3. metode *brute force* menghasilkan algoritma standard untuk tugas-tugas komputasi seperti penjumlahan/perkalian n buah bilangan, menentukan elemen ekstrim (maksimum atau minimum) di dalam senarai / tabel / *list*.

Kelemahan dari metode *brute force*:

1. metode *brute force* jarang menghasilkan algoritma yang mangkus.
2. untuk beberapa kasus metode *brute force* tidak *feasible* atau terlalu lambat sehingga tidak dapat diterima.
3. tidak memerlukan kreativitas yang tinggi untuk menemukan teknik untuk memecahkan masalah.

B. Permainan Countdown Number

- Deskripsi singkat permainan *Countdown Number*

Permainan *Countdown Number* merupakan sebuah *gameshow* dari *british* yang mengasah kemampuan para pesertanya dengan *puzzle* aritmatika.

Pada awal *gameshow*, peserta diberikan kewajiban untuk memilih 6 buah angka yang terbagi menjadi dua tingkatan, angka besar / *Large Number* (100,75,50,25) lalu angka kecil / *Small Number* (1,2,3,4,5,6,7,8,9). Peserta bebas memilih kombinasi *Large Number* dan *Small Number* hingga terpilih 6 angka, proses pemilihan angka ini dilakukan dengan kondisi peserta tidak bisa melihat angka yang dipilihnya (posisi papan nomor terbalik), setelah peserta memilih 6 angka, *host* akan membuka angka-angka yang telah dipilih tadi (papan nomor terbuka), setelah 6 angka tadi sudah diketahui oleh peserta, maka *computer* akan membangkitkan 3 digit angka secara random (101-999) yang menjadi *goal number*, lalu *goal number* inilah yang akan dibuat sedekat mungkin jika tidak berhasil menemukan angka yang sama persis dari 6 angka yang sudah dipilih tadi, tapi pada eksperimen ini hanya dihasilkan kombinasi yang menghasilkan hasil yang sama yang ditampilkan.

Algoritma *Brute Force* memiliki beberapa karakteristik, diantaranya:

1. Algoritma *brute force* umumnya tidak “cerdas” dan tidak mangkus, karena algoritma ini membutuhkan jumlah komputasi yang besar dan waktu yang lama dalam penyelesaiannya. Sehingga kadang algoritma *brute force* disebut sebagai algoritma naif.
2. Algoritma *brute force* lebih cocok untuk persoalan yang berukuran kecil, karena pertimbangannya yakni sederhana dan implementasinya mudah, alih-alih memikirkan cara yang efisien dan efektif untuk kasus yang kecil (yang bisa menghabiskan waktu untuk memikirkannya) lebih baik jika menggunakan *brute force*, karena kekuatan komputasi komputer saat ini sudah jauh lebih baik daripada puluhan tahun yang lalu. Biasanya algoritma *brute force* juga digunakan sebagai basis perbandingan dengan algoritma lain yang lebih mangkus.


```

C:\JAVA\makalahStima>java Makalah
+++++
-----
COUNTDOWN NUMBER
-----
+++++

berapa goal numbernya ?
985
masukkan angka ke 1
100
masukkan angka ke 2
50
masukkan angka ke 3
9
masukkan angka ke 4
5
masukkan angka ke 5
1
masukkan angka ke 6
7
Kombinasi yang ditemukan adalah
100+50-9-1*7+5

```

Gambar 4. Test Case pertama

```

C:\JAVA\makalahStima>java Makalah
+++++
-----
COUNTDOWN NUMBER
-----
+++++

berapa goal numbernya ?
5
masukkan angka ke 1
100
masukkan angka ke 2
75
masukkan angka ke 3
50
masukkan angka ke 4
25
masukkan angka ke 5
9
masukkan angka ke 6
8
Kombinasi yang ditemukan adalah
100-75+50-25-9/8

```

Gambar 6. Test Case ketiga dimana Goal number tidak berada pada range yang telah disepakati yakni (101 sd 999), tetapi program tetap bisa menemukan kombinasinya

```

+++++
-----
COUNTDOWN NUMBER
-----
+++++

berapa goal numbernya ?
125
masukkan angka ke 1
100
masukkan angka ke 2
50
masukkan angka ke 3
75
masukkan angka ke 4
25
masukkan angka ke 5
1
masukkan angka ke 6
1
Kombinasi yang ditemukan adalah
100-50+75

```

Gambar 5. Test Case kedua dimana tidak semua angka digunakan untuk menemukan *goal number*x

```

C:\JAVA\makalahStima>java Makalah
+++++
-----
COUNTDOWN NUMBER
-----
+++++

berapa goal numbernya ?
5
masukkan angka ke 1
75
masukkan angka ke 2
25
masukkan angka ke 3
9
masukkan angka ke 4
8
masukkan angka ke 5
50
masukkan angka ke 6
100
Kombinasi yang ditemukan adalah
75-25-9/8

```

Gambar 7. Test Case keempat dimana *goal number* dan angka-angka yang digunakan sama hanya berbeda di letaknya saja, akan menghasilkan kombinasi yang berbeda

```
C:\JAVA\makalahStima>java Makalah
+++++
-----
COUNTDOWN NUMBER
-----
+++++

berapa goal numbernya ?
999
masukkan angka ke 1
1
masukkan angka ke 2
2
masukkan angka ke 3
10
masukkan angka ke 4
3
masukkan angka ke 5
5
masukkan angka ke 6
8
tidak ditemukan kombinasi
```

Gambar 8. Test Case kelima saat tidak ditemukan kombinasi yang bisa digunakan untuk mencapai *goal number*

B. Analisis

Pada Test Case pertama, semua angka digunakan untuk mendapatkan kombinasi *goal number* karena pengulangan yang dilakukan baru berhenti setelah pembangkitan kombinasi dengan lima operator.

Pada Test Case kedua, tidak semua angka digunakan untuk mendapatkan kombinasi *goal number* karena pengulangan yang dilakukan sudah berhenti setelah pembangkitan kombinasi dengan dua operator yakni “+”

Pada Test Case ketiga dan keempat semua angka yang diperbolehkan untuk digunakan sama, *goal numbernya* juga sama, tetapi hasil kombinasinya berbeda. Ini terjadi karena program melakukan perulangan menggunakan elemen array yang paling awal terlebih dahulu, sehingga pada test case ketiga perulangan baru berhenti pada lima operator, sedangkan pada test case keempat perulangan berhenti pada tiga operator, sehingga bisa disimpulkan bahwa proses memasukkan angka-angka bisa menentukan banyaknya perulangan yang dilakukan.

Pada Test case kelima tidak ditemukan kombinasi yang dapat digunakan untuk membentuk *goal number*, yang berarti program sudah melakukan perulangan dari satu operator

hingga lima operator dan tidak menemukan kombinasi yang bisa dijadikan *goal number*.

V. KESIMPULAN

Pada kasus permainan *countdown number* penggunaan algoritma *brute force* bisa dibilang cukup bisa diandalkan, karena kasus ini tidak terlalu besar sehingga masih *feasible* untuk melakukan pencarian menggunakan teknik *exhaustive search*. Proses pencarian kombinasi sangat tergantung dari urutan angka-angka yang dimasukkan ke tabel karena proses perulangan dicari dari elemen yang paling awal sehingga jika berbeda tabel walaupun komposisi isi tabel sama akan menghasilkan jumlah perulangan yang berbeda juga.

VI. UCAPAN TERIMAKASIH

Pertama – tama saya mengucapkan terima kasih kepada Tuhan Yang Maha Esa yang telah melimpahkan rahmat dan hidayah-Nya sehingga makalah Strategi Algoritma ini dapat diselesaikan tepat waktu. Saya juga mengucapkan terima kasih kepada kedua orang tua saya yang selalu memberi dukungan dan doa restu kepada saya sehingga dapat menempuh pendidikan sampai saat ini. Tak lupa saya juga mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, MT., ibu Dr. Nur Ulfa Maulidevi, S.T., dan ibu Masayu Leylia Khodra, ST., MT. yang berperan sebagai dosen mata kuliah IF 2211 Strategi Algoritma sehingga dengan ilmu pengetahuan seputar Strategi Algoritma, saya dapat membuat dan menyelesaikan makalah ini.

REFERENCES

- [1] Munir, Rinaldi. 2009. Diktat Kuliah Strategi Algoritmik IF2251 Strategi Algoritmik. Departemen Teknik Informatika ITB.
- [2] <http://stackoverflow.com/questions/15293232/how-to-design-an-algorithm-to-calculate-countdown-style-maths-number-puzzle> (diakses pada 18 Mei 2017 pukul 18.00)
- [3] Slide kuliah IF2211 Strategi Algoritma Algoritma Brute Force (2016).ppt
- [4] <http://www.datagenetics.com/blog/august32014/index.html> (diakses pada 18 Mei 2017 pukul 19.00)
- [5] <http://happysoft.org.uk/countdown/numgame.php> (diakses pada 18 Mei 2017 pukul 21.00)

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, 18 Mei 2017

A handwritten signature in black ink, appearing to read 'Farhan Amin', written over a horizontal line.

Farhan Amin - 13515043