Penggunaan Algoritma Brute Force dalam Pengecekan Password

Ezra Maringan Christian Mastra Hutagaol – 13521073
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13521073@std.stei.itb.ac.id

***Abstrak*—** Password adalah kumpulan karakter atau string yang digunakan oleh user untuk memverifikasi identitas diri kepada sistem keamanan yang dimiliki sistem tersebut. Sistem keamanan akan membandingkan kode – kode yang dimasukkan oleh user dengan daftar yang disimpan oleh sistem keamanan tersebut. Jika kode yang dibandingkan cocok, maka sistem keamanan akan mengizinkan akses kepada pengguna tersebut terhadap layanan yang terdapat di dalam sistem tersebut. Pada makalah kali ini akan dibahas mengenai penggunaan Algoritma brute force dalam pengecekan password.

Keywords— Brute force; Password; Teknologi; Keamanan

I. PENDAHULUAN

Gambar 1. Ilustrasi password
(https://tekno.sindonews.com/read/604707/207/inilah-kode-password-yang-sering-dipakai-penduduk-dunia-1637341916)

Password adalah hal penting yang harus dimiliki oleh setiap orang jika ingin membuat media sosial, blog, email, dan lain sebagainya. Adanya password dibutuhkan untuk menjaga keamanan agar akun terhindar dari tindakan-tindakan oknum yang tidak bertanggungjawab. Meskipun terlihat mudah, tidak jarang sebagian orang lupa password yang dimiliki dan berakhir tidak bisa mengakses akun miliknya.

Dalam penerapannya, ada beberapa jenis-jenis password yang dapat digunakan, yaitu 2FA dan MFA. Sebagai informasi, kedua jenis ini memiliki fungsi yang sama yakni untuk mengamankan suatu website atau layanan online. Biasanya jenis-jenis password tersebut sering ditemui di dalam aplikasi finansial, dompet elektronik, dan lainnya.

Password yang baik merupakan password yang susah untuk ditebak oleh orang lain. Password yang buruk akan menjadi sasaran empuk bagi orang yang memiliki niat jahat kepada kita. Selain itu, seiring berkembangnya teknologi serangan siber juga semakin canggih. Oleh karena itu kita harus bisa membuat dan mengingat password yang kita buat dengan baik.

Pada makalah ini akan dibahas tentang pengecekan Password menggunakan algoritma Brute Force

II. TEORI DASAR

A. Algoritma Brute Force

1. Definisi Algoritma Brute Force

   Algoritma Brute Force adalah pendekatan yang lempang (straightforward) untuk memecahkan suatu persoalan. Biasanya algoritma ini didasarkan pada pernyataan pada persoalan (problem statement) dan definisi/konsep yang dilibatkan. Algoritma brute force memecahkan persoalan dengan sangat sederhana, langsung, jelas caranya (obvious way), *just do it!* Atau *just solve it!*.

   Berikut beberapa contoh dari algoritma Brute Force :

   a. Mencari elemen terbesar

      Pada persoalan ini, algoritma brute force akan membandingkan setiap elemen senarai mulai dari a_1 sampai a_n untuk menemukan elemen terbesar

   b. Pencarian beruntun (Sequential Search)

      Pada persoalan ini, algoritma brute force akan membandingkan setiap elemen senarai dengan x. Pencarian akan selesai jika x ditemukan atau seluruh elemen senarai sudah habis diperiksa.

2. Karakteristik Algoritma Brute Force

   a. Algoritma Brute Force umumnya tidak ”cerdas” dan tidak mangkus, karena ia membutuhkan volume komputasi yang besar dan waktu yang lama dalam penyelesaiannya. Kata ”force” mengindikasikan ”tenaga” ketimbang otak. Kadang-kadang algoritma Brute Force disebut juga algoritma naik (naive algorithm).

b. Algoritma brute force lebih cocok untuk persoalan yang ukuran masukannya (n) kecil. Pertimbangannya karena persoalan yang ukuran masukannya (n) kecil sederhana dan implementasinya mudah. Algoritma Brute Force sering digunakan sebagai basis pembanding dengan algoritma lain yang lebih mangkus.

c. Meskipun bukan metode problem solving yang mangkus, hampir semua persoalan dapat diselesaikan dengan algoritma brute force. Ini adalah kelebihan brute force. Sukar menunjukkan persoalan yang tidak dapat diselesaikan dengan metode brute force.

3. Kelebihan dan kekurangan Algoritma Brute Force

Kelebihan :

a. Algoritma brute force dapat diterapkan untuk memecahkan hampir sebagian besar masalah (wide applicability).

b. Algoritma Brute Force sederhana dan mudah dimengerti

c. Algoritma Brute Force menghasilkan algoritma yang layak untuk beberapa masalah penting seperti pencarian, pengurutan, pencocokan string, perkalian matriks

d. Algoritma Brute Force menghasilkan algoritma baku (standard) untuk tugas-tugas komputasi seperti penjumlahan/perkalian n buah bilangan, menentukan elemen minimum atau maksimum di dalam senarai (larik)

Kekurangan :

a. Algoritma Brute Force jarang menghasilkan algoritma yang mangkus

b. Algoritma Brute Force umumnya lambat untuk masukan berukuran besar sehingga tidak dapat diterima

c. Tidak sekontruktif/sekreatif strategi pemecahan masalah lainnya

4. Contoh-contoh persoalan Brute Force lainnya

a. Cryptarithmetic

Diberikan sebuah penjumlahan huruf, carilah angka yang merepresentasikan huruf-huruf tersebut.

b. Permainan 24 (24 Game)

Diberikan 4 buah bilangan bulat. Bagaimana mengkombinasikan keempat bilangan bulat tersebut dengan operator aritmetika sehingga hasilnya = 24

c. Crossword Puzzle

Crossword Puzzle atau permainan teka-teki silang adalah permainan yang memasangkan semua kata yang tersedia ke dalam kotak-kotak yang bersesuaian, baik secara mendatar maupun menurun

d. Kakurasu

Kakurasu adalah permainan teka teki logika yang berasal dari Jepang. Teka-teki dimainkan pada grid persegi, misalnya 4 x 4, 5 x 5, 6 x 6, dst. Pada setiap pinggir kotak paling kanan dan kotak paling bawah terdapat angka-angka. Tujuan permainannya adalah memilih kotak-kotak di dalam grid sehingga jumlah nilainya sama dengan angka yang ditunjukkan di kotak paling kanan dan kotak paling bawah (secara horizontal dan vertikal)

e. Futoshiki

Futoshiki adalah permainan teka-teki logika yang berasal dari Jepang. Teka-teki dimainkan pada grid persegi, misalnya 5 x 5. Tujuannya adalah untuk menempatkan angka 1 sampai 5 (atau dimensi lainnya) sehingga setiap baris dan kolom berisi masing – masing angka 1 sampai 5. Beberapa angka diberikan di awal sebagai panduan. Selain itu, beberapa tanda ketidaksamaan (< atau >) diberikan di antara beberapa kotak, sedemikian sehingga nilai yang satuh arus lebih tinggi atau lebih rendah dari tetangganya

f. Kakuro

Tujuan permainan adalah untuk mengisi semua kotak kosong di dalam grid dengan hanya 1-9 angka sehingga angka-angka yang anda masukkan jumlahnya sama dengan clue (penunjuk) yang terletak pada grid di sebelah kiri dan grid sebelah. Jumlah dari setiap blok horisontal sama dengan petunjuk di sebelah kiri, dan jumlah dari setiap blok vertikal sama dengan petunjuk di atasnya. Selain itu, tidak boleh yang sama digunakan di dalam blok yang sama lebih dari sekali

B. Exhaustive Search

Exhaustive Search adalah Teknik pencarian soluci secara Brute Force untuk persoalan-persoalan kombinatorik. Persoalan kombinatorik adalah persoalan di antara objek-objek kombinatorik seperti permutasi, kombinasi, atau himpunan bagian dari sebuah himpunan. Walaupun solusi yang ditemukan dari exhaustive search pasti solusi terbaik, kompleksitas algoritma exhaustive search masih eksponensial, sehingga algoritma ini cenderung dihindari.

Algoritma exhaustive search dapat diperbaiki kinerjanya sehingga tidak perlu melakukan pencarian terhadap semua kemungkinan solusi. Salah satu ccara untuk mengoptimalkan pencarian solusi pada exhaustive search dapat digunakan Teknik heuristic. Teknik ini mengeliminasi kemungkinan

solusi yang tidak mungkin menjadi solusi terbaik sehingga pencarian yang dilakukan menjadi lebih sedikit.

Langkah – Langkah metode exhaustive search sebagai berikut :

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

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

Contoh – contoh exhaustive search:

a. Travelling Salesperson Problem (TSP)

   Diberikan n buah kota serta diketahui jarak antara setiap kota satu sama lain. Temukan perjalanan (tour) dengan jarak terpendek yang dilakukan oleh seorang pedagang sehingga ia melalui setiap kota tepat hanya sekali dan kembali lagi ke kota asal keberangkatan

b. 1/0 Knapsack Problem

   Diberikan n buah objek dan sebuah knapsack dengan kapasitas bobot K. Setiap objek memiliki properti bobot (weigth) w_i dan keuntungan (profit) p_i

```
Function PencocokanString(input P : string, T : string, m, n : integer, output idx : integer → integer)
  {Luaran : lokasi awal kecocokan (idx) }
Deklarasi
  i : integer
  ketemu : boolean
Algoritma:
  i ← 0
  ketemu ← false
  while (i < n – m) and (not ketemu) do
    j ←i
    while (j < m) and (Pj = Ti+j) do
      j←j+1
    endwhile
    {j > m or Pj != Ti+j}do

    if j = m then {kecocokan string ditemukan}
        ketemu ← true
    else
        i ← i + 1{geser pattern satu karakter ke kanan teks}
    endif
  endwhile
    { i > n – m or ketemu }

if ketemu then return i + 1 else return  -1 endif
```

Teks: NOBODY NOTICED HIM
Pattern: NOT

```
   NOBODY NOTICED HIM
1  NOT
2   NOT
3    NOT
4     NOT
5      NOT
6       NOT
7        NOT
8         NOT
```

Gambar 2. Contoh brute force pada Pattern Matching

(https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-(2022)-Bag1.pdf)

Dibawah ini merupakan contoh notasi algoritmik dengan algoritma brute force untuk pencocokan string.

C. Teknik Heuristik

Algoritma exhaustive search dapat diperbaiki kinerjanya sehingga tidak perlu melakukan pencarian solusi dengan mengeksplorasi semua kemungkinan solusi. Salah satu Teknik yang digunakan untuk mempercepat pencarian solusi di dalam exhaustive search adalah Teknik heuristic heuristic). Terkadang disebut juga fungsi heuristic. Dalam exhaustive search, Teknik heuristic digunakan untuk mengeliminasi beberapa kemungkinan solusi tanpa harus mengeksplorasi seluruh kemungkinan solusi secara penuh

Heuristik merupakan teknik yang dirancang untuk memecahkan persoalan dengan mengabaikan apakah teknik tersebut terbukti benar secara matematis. Sebab teknik ini menggunakan pendekatan yang tidak formal, misalnya berdasarkan penilaian intuiti, terkaan, atau akal sehat.

Contoh : program antivirus menggunakan pola-pola heuristik untuk mengidentifikasi dokumen yang terkena virus atau malware.

Heuristik adalah seni dan ilmu menemukan (art and science of discovery) Kata heuristik diturunkan dari Bahasa Yunani yaitu ”eureka” yang berarti ”menemukan” (to find atau to discover). Matematikawan Yunani yang bernama Archimedes yang

melontarkan kata ”heureka”, dari sinilah kita menemukan kata ”eureka” yang berarti ”I have found it”

Heuristik mengacu pada teknik memecahkan persoalan berbasis pengalaman, dari proses pembelajaran, dan penemuan solusi meskipun tidak dijamin optimal. Heuristik berbeda dengan algoritma, heuristik berlaku sebagai panduan (guideline), sedangkan algoritma adalah urutan langkah-langkah penyelesaian persoalan. Teknik heuristik menggunakan terkaan, instuisi, dan common sense. Secara matematis tidak dapat dibuktikan, namun sangat berguna. Teknik heuristik mungkin tidak selalu memberikan hasil optimal, tetapi secara ekstrim ia berguna pada penyelesaian persoalan.

Heuristik yang bagus dapat secara dramatis mengurangi waktu yang dibutuhkan untuk memecahkan persoalan dengan cara mengeliminir kebutuhan untuk mempertimbangkan kemungkinan solusi yang tidak perlu. Heuristik tidak menjamin selalu dapat memecahkan persoalan, tetapi seringkali memecahkan persoalan dengan cukup baik untuk kebanyakan persoalan, dan seringkali pula lebih cepat daripada pencarian solusi secara exhaustive search. Sudah sejak lama heuristik digunaan secara intensif di dalam bidang intelijensia buatan (artificial intelligence), misalnya di dalam metode hill climbing, best first search, algoritma A*, dan lain-lain

Contoh penggunaan heuristik untuk mempercepat algoritma exhaustive search ada pada persoalan anagram. Anagram adalah penukaran huruf dalam sebuah kata atau kalimat sehingga kata atau kalimat yang baru mempunyai arti lain. Contoh anagram misalnya :

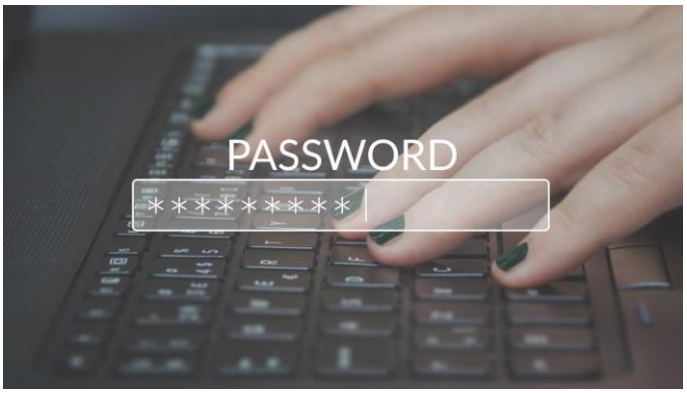
lived →devil

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.

D. Password



Gambar 3. Ilustrasi password (https://tirto.id/daftar-password-terburuk-tahun-2020-salah-satunya-123456-f7bC)

Password adalah serangkaian karakter yang digunakan untuk mengautentikasi pengguna pada sistem komputer. Password sering juga disebut dengan kata sandi. Dalam penerapannya ada beberapa jenis password yang dapat digunakan, yaitu 2FA dan MFA. Kedua jenis ini memiliki fungsi yang sama untuk mengamankan suatu layanan.

Berikut penjelasan jenis-jenis password tersebut :

a. 2FA (Two Factor Authentication)
   Two factor authentication (2FA) adalah metode keamanan cyber yang harus melewati dua tahap autentikasi sebelum bisa mengakses suatu data. Tahap pertamanya adalah password, kemudian tahap keduanya adalah kode khusus yang dikirimkan ke perangkat tertentu. Dengan mengaktifkan 2FA, seeorang yang berhasil mendapatkan username dan password seseorang, belum tentu bisa mengakses data karena si peretas tidak memiliki kode khusus untuk mengkonfirmasi kepemilikannya.
b. MFA (Multi Factor Authentication)
   Selain 2FA, ada juga Multi Factor Authentication (MFA). MFA adalah salah satu cara terbaik untuk mencegah peretas mendapatkan akses ke akun email. Dengan autentikasi multifaktor, akun email tidak dapat diakses kecuali memiliki dua informasi

III. IMPLEMENTASI

Implementasi pengecekan password dengan algoritma brute force pada makalah ini menggunakan Bahasa Python

Berikut merupakan implementasi kode dalam python dan penjelasannya.

```
Def brute_force(password):

# Daftar password yang ada
        Password_list = [‘aposd9%’, ‘sksa2298j&’, ‘password3’, ‘aksiemf72$’, ‘slpformg82^’]

# Memeriksa password pada daftar password yang ada
        for p in password_list:
                if password == p:
                        return True
        return False

# Meminta input password dari pengguna
input_password = input(“Masukkan password: ”)

# Memeriksa kecocokan password menggunakan brute force
If brute_force(input_pasword):
        Print(“Password yang Anda masukkan sesuai dengan password yang ada.”)
else :
        print(”Password yang Anda masukkan tidak sesuai dengan password yang ada.”)
```

```
def brute_force(password):
    # Daftar password yang ada
    password_list = ['aposd9%', 'sksa2298j&', 'password3', 'aksiemf72$', 'slpformg82^']
```

Gambar 4. Implementasi program (dok. Pribadi)

Pada bagian ini, password_list akan menampung password yang dimiliki.

```
# Memeriksa password pada daftar password yang ada
for p in password_list:
    if password == p:
        return True
return False
```

Gambar 5. Implementasi program (dok. Pribadi)

Pada bagian ini, akan dilakukan pengecekan password untuk setiap password yang ada di dalam password list

```
input_password = input("Masukkan password: ")
```

Gambar 6. Implementasi program (dok. Pribadi)

Pada bagian ini, akan diminta input password, untuk dicek apakah password yang diinput terdapat di password_list atau tidak

```
if brute_force(input_password):
    print("Password yang Anda masukkan sesuai dengan password yang ada.")
else:
    print("Password yang Anda masukkan tidak sesuai dengan password yang ada.")
```

Gambar 7. Implementasi program (dok. Pribadi)

Pada bagian ini, program akan memberitahu apakah password yang di input terdapat di password list atau tidak.

IV. Pengujian

1. Test case 1

Pada Test case ini akan dicoba ketika input password terdapat di dalam password_list

Input : aposd9%

```
Masukkan password: aposd9%
Password yang Anda masukkan sesuai dengan password yang ada.
```

Gambar 8. Test case 1

2. Test case 2

Pada Test case ini akan dicoba ketika input password tidak terdapat di dalam password_list

Input: abc123

```
Masukkan password: abc123
Password yang Anda masukkan tidak sesuai dengan password yang ada.
```

Gambar 9. Test case 2

V. Kesimpulan

Dari percobaan yang dilakukan di atas dapat disimpulkan bahwa algoritma Brute Force dapat digunakan untuk pengecekan password. Algoritma Brute Force dapat digunakan pada hampir seluruh persoalan yang ada. Algoritma brute force cocok untuk peroalan kecil dan sederhana yang tidak membutuhkan komputasi yang besar, namun algoritma ini bukanlah algoritma yang mangkus. Walaupun demikian berdasarkan test case, algoritma ini mampu memberikan solusi yang tepat. Kompleksitas dari algoritma ini adalah 2^n, sehingga untuk list password yang lebih besar dibutuhkan algoritma lain yang dapat memberikan waktu komputasi yang lebih cepat.

VI. Ucapan terima kasih

Penulis mengucapkan terima kasih dan puji syukur terhadap Tuhan Yang Maha Esa atas rahmatnya, penulis bisa menyelesaikan makalah ini dengan tepat waktu. Penulis juga menyampaikan terima kasih kepada keluarga, teman-teman, dan semua pihak yang telah mendukung studi dan proses pembelajaran dalam mata kuliah IF2211 Strategi lgoritma. Penulis juga mengucapkan terima kasih kepada dosen pengampu mata kuliah IF2211 Strategi Algoritma yitu, Bapak Dr. Rinaldi Munir, MT atas ilmu yang telah diberikan selama semester 2 tahun ajaran 2022/2023.

References

[1] https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-(2022)-Bag1.pdf , Diakses pada 22 mei 2023

[2] https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-(2022)-Bag2.pdf , Diakses pada 22 mei 2023

[3] https://www.dewaweb.com/blog/apa-itu-password/ , Diakses pada 22 mei 2023

[4] https://course-net.com/blog/apa-itu-password/ , Diakses pada 22 mei 2023

Ezra M C M H / 13521073

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, 22 Mei 2023