

Implementasi Algoritma Brute Force pada pencarian kata dan pola

Sultan Ahmad Alfath 120140046 (*Author*)

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Sumatera, Jl. Terusan Ryacudu Lampung

E-mail (gmail): sultan.120140046@student.itera.ac.id

Abstract—Bruter force merupakan algoritma yang mencoba segala kemungkinan dan memilah seluruh kemungkinan tersebut untuk mencari solusi terbaik pada sebuah permasalahan. Pencarian kata digunakan untuk mencari kata pada sebuah teks dokumen, dan brute force diimplementasikan pada program pencarian kata yang saya buat. Hasil yang didapatkan adalah algoritma ini pas untuk pengimplementasian dalam pencarian kata pada sebuah teks.

Kata kunci—Brute force; Pencarian kata; algoritma

I. PENDAHULUAN

Sebuah buku, artikel, jurnal, maupun dokumen teks memuat begitu banyak kata dan huruf. Pembaca kadang begitu kesulitan untuk menemukan sebuah kata jika tidak meletakkan tanda pada lembar dokumen. Mencari kata pada dokumen tercetak/biasa berbeda dengan mencari kata pada dokumen didalam komputer, karena terdapat fitur “Search” didalam perangkat lunak pengolah kata. Fitur ini diciptakan untuk mempermudah pekerjaan pengguna komputer dalam mengolah dokumen-dokumen teks didalamnya.

Terdapat berbagai macam algoritma yang diterapkan kepada komputer ini. Fitur search juga memiliki berbagai macam algoritma yang bermacam-macam. Salah satunya adalah brute force. Algoritma ini terkenal sebagai algoritma yang tidak efektif pada pemrosesannya, namun efektif pada hasil prosesnya. Pada makalah ini, saya akan menerapkan algoritma brute force pada pencarian kata dan pola pada sebuah teks kalimat yang sudah terdefinisi pada program python.

II. DASAR TEORI

A. Algoritma brute force

Algoritma ini lebih terkenal karena penggunaannya dalam dunia keamanan siber. Algoritma brute force menggunakan berbagai kemungkinan untuk menghasilkan solusi yang paling optimal pada permasalahannya. Algoritma ini membutuhkan waktu komputasi yang cukup lama jika mengolah permasalahan dengan data yang besar. Penyebabnya adalah, karena algoritma ini menggunakan berbagai kemungkinan yang kemudian akan dicari solusi paling optimal dari seluruh kemungkinan-kemungkinan yang ada.

Proses komputasi kemungkinan ini tidak efisien dibanding algoritma lainnya, karena selain kurangnya efisiensi waktu, algoritma ini juga menggunakan sumber daya komputer yang cukup besar namun tergantung pada permasalahannya. Contoh sederhananya, pada suatu device terdapat 4 digit password berbentuk angka yang digunakan si pengguna. Kita ingin membobol menggunakan algoritma brute force, dan kemungkinan terburuk dari kasus ini adalah password tersebut 9999. Pembobolan ini akan membutuhkan waktu yang relatif lama karena brute force akan mencoba dari digit terkecil ke digit paling maksimal dari sebuah angka, contohnya 0000, 0001, 0002, , 9999. Jika percobaan secara otomatis dilakukan oleh komputer dan membutuhkan waktu 1 detik tiap percobaan, maka untuk ke angka 9999 dari angka 0000 kurang lebih membutuhkan waktu sekitar 2 jam.

Kelebihan brute force:

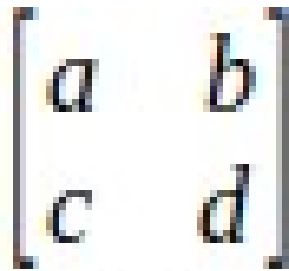
- Sebagian besar masalah dapat dicari solusinya menggunakan algoritma brute force
- Algoritma ini mudah dimengerti alurnya
- Algoritma ini menghasilkan solusi optimal pada suatu permasalahan
- Brute force cocok digunakan untuk proses pencarian kata, pengurutan, dan beberapa hal lainnya.

Kekurangan brute force:

- Proses algoritma lama apabila menerima inputan data yang besar
- Algoritma ini kaku, dan tidak kreatif seperti algoritma lainnya
- Brute force jarang menjadi algoritma mangkus

B. Matriks

Matriks merupakan bilangan atau ekspresi atau simbol yang disusun didalam kolom dan baris yang jika diperhatikan akan berbentuk seperti persegi atau persegi panjang. Contoh gambar matriks:



Gambar 2.1 Matriks 2x2

Pada matriks tersebut terdapat huruf a, b, c, d. Baris adalah a, c (atas kebawah) sedangkan kolom adalah a, b (kiri kekanan). Pada matriks didalam pemrograman matriks memiliki index yang dimulai dengan angka 0 pada masing-masing baris dan kolom. Jika dilihat huruf “a” terdapat pada baris 0 dan kolom 0. huruf “b” terdapat pada baris 0 dan kolom 1. Pada pemrograman, matriks baris 0 kolom 1 ditulis dengan format <nama matriks>[baris][kolom], contohnya: “matriks[0][1]”.

Berikut Jenis-jenis matriks yang ada:

- Matriks baris
Matriks ini terdiri dari 1 baris saja.
- Matriks kolom
Matriks terdiri dari 1 kolom saja.
- Matriks persegi
Matriks persegi adalah matriks yang AxB dimana A sama dengan B.
- Matriks diagonal
Matriks ini memiliki nilai pada diagonalnya, sedangkan selain diagonal seluruhnya bernilai 0.
- Matriks identitas
Matriks ini diagonalnya bernilai 1 seluruhnya, sedangkan lainnya bernilai 0
- Matriks nol
Matriks ini seluruh elemennya bernilai 0.

C. Python

Python merupakan bahasa pemrograman tingkat tinggi. Dikatakan tingkat tinggi karena bahasa pemrograman ini mudah dimengerti oleh manusia. Python adalah bahasa yang tidak sekompleks pemrograman c++ atau javascript, karena syntax nya yang simple dan mudah diterapkan. Python sering

digunakan untuk pembelajaran mesin (machine learning), data science dan pengembangan A.I (Artificial Intelligence). Python tidak memerlukan kompilasi untuk menjalankan kodenya seperti c++. Python adalah *interpreted programming language* dimana kode tersebut dapat dijalankan bertahap pada komputer.

Berikut beberapa penggunaan python untuk implementasi:

- Pengembangan web
- Pengembangan game
- Kecerdasan buatan
- Pengembangan software
- Pengaplikasian pada bidang bisnis
- Pengembangan bahasa pemrograman, dan lain-lain.

Dari contoh-contoh diatas, dapat diasumsikan bahwa python cukup populer dan luas pengimplementasiannya di dunia teknologi. Python dipilih menjadi bahasa pemrograman yang digunakan karena syntax yang digunakan cukup mudah dipahami dibanding bahasa pemrograman lainnya. Terdapat pula banyak module/library python dan fungsi-fungsi bawaan yang dapat banyak membantu programmer untuk meluncurkan produktivitasnya. Beberapa contoh fungsi yang sering digunakan pada python dan terbukti berguna:

- Numpy
Library ini menampung fungsi dan kode untuk memanipulasi matriks, array, dan memiliki banyak fungsi untuk operasi matematika.
- Sorted
Fungsi ini digunakan untuk mengurutkan sebuah string atau integer pada variabel atau array yang ingin diurutkan. Biasanya, fungsi ini akan mengurutkan dari kecil ke besar.
- Split
Fungsi ini digunakan untuk membuat string menjadi sebuah array yang isinya dibatasi sampai whitespace atau spasi yang ditemukan pada sebuah string.
- Opencv
Library ini digunakan untuk machine learning, computer vision, data science, dan juga artificial intelligence. Library ini bersifat eksternal atau perlu di download dari website untuk menggunakannya. Berbeda dengan library bawaan, dimana dapat langsung digunakan pada saat selesai penginstallan python pada device pengguna.

Selain fungsi diatas, terdapat ratusan fungsi lainnya yang berguna untuk dunia pemrograman.

D. Search

Algoritma search dibuat untuk mencari atau mengecek elemen untuk mencari yang ingin dicari. Algoritma ini terbagi menjadi 2:

1. Sequential search

Algoritma ini digunakan untuk mengecek elemen array.

2. Interval search

Algoritma ini digunakan untuk melakukan pencarian pada data yang telah disort.

Kedua jenis tersebut, nantinya akan dibagi-bagi lagi menjadi beberapa bagian. Berikut jenis-jenisnya:

1. Linear
2. Jump
3. Binary
4. Fibonacci
5. Interpolation
6. Exponential
7. Sublist, dan lainnya.

Pada program yang saya buat, saya menggunakan metode sequential search yang repetitif (berulang-ulang). Lebih spesifiknya, saya menerapkan linear search pada program kali ini.

III. ANALISIS DAN PEMBAHASAN

Pada makalah ini saya mengambil topik berupa penerapan algoritma brute force. Berikut spesifikasi sistem yang saya gunakan:

1. Sistem Operasi: Ubuntu 20.04 LTS
2. Code editor: Visual Code Studio
3. Processor: Intel® Celeron(R) CPU 1017U @ 1.60GHz × 2
4. Memory: 2GB
5. Tipe arsitektur: 64-bit

Pada penerapan saya menggunakan beberapa hal seperti matriks, manipulasi teks, loop, dan percabangan. Pada kode ini saya menggunakan bahasa pemrograman python.

A. Manipulasi teks

Pada kode program terdapat 3 manipulasi teks. Pertama adalah menjadikan teks kalimat tersebut menjadi array yang isinya adalah kata dari kalimat tersebut. Contoh:

Kalimat: "saya adalah manusia"

Array: ["saya", "adalah", "manusia"]

Manipulasi ini menggunakan syntax <variabel str>.split() pada program. Syntax tersebut cara kerjanya adalah memasukkan seluruh karakter sampai bertemu sebuah spasi

atau *whitespace* pada kalimat. Kedua, manipulasi terdapat pada output pattern atau kata yang ditemukan. Contoh:

Kalimat: "saya adalah manusia"

Search: "a"

Hasil: "sAyA AdAlAh mAnusiA"

Pada program ini, saya akan mencetak kapital huruf yang ditemukan pada kalimat. Sebenarnya, lebih baik jika menggunakan output berupa colored text, tetapi dikarenakan hambatan yang sulit ditemukan jalan keluarnya, saya memilih untuk memanipulasinya menjadi huruf uppercase.

Ketiga, saya menggunakan manipulasi untuk mengakses karakter dari sebuah string pada array kalimat yang terkonversi. Misal:

String: "Halo!"

kode: print(string[0])

output: "H"

Pada contoh diatas, karakter string sebenarnya dapat diakses seperti array, dan cara aksesnya cukup menambahkan indeks array setelah nama karakter tanpa spasi. Contoh penulisan:

<NamaStringVariabel>[Posisi karakter]

Posisi karakter diawali dengan index 0 dan kemudian seterusnya.

B. Algoritma pencarian

Algoritma pencarian yang saya gunakan agak kompleks. Pertama, kita akan memanipulasi teks tersebut menjadi sebuah array dengan isinya adalah kalimat dari teks tersebut. Lalu, setelahnya kita akan membuat inputan untuk kata yang akan dicari. Setelahnya, kita akan memulai pemilahan perkata dari array tersebut. Jika tidak sama, maka karakter yang tidak sama akan dicetak normal/huruf kecil. Jika ditemukan kesamaan, maka karakter yang ditemukan kesamaannya akan dicetak uppercase. Contoh sederhana:

kalimat: "saya andi"

array: ["saya ", "andi"]

search: "ya"

Berikutnya, kita akan mengecek array[0] untuk mencari apakah ada karakter dari variabel search yang sesuai pada array. Kita akan membuat variabel string sementara "temp" untuk menampung karakter yang akan dicek. Proses berikutnya:

```
temp = "" → tanpa spasi
```

```
for i in range len(array[0]):
```

```
temp += array[0][i]
```

```
if panjang(temp)>panjang(search):
```

```
print(temp[0])
```

```
temp.pop(0)
```

```

if panjang(temp) == panjang(search):
    # cek apakah sama
    if sama:
        #output temp dengan kapital
        temp = "" → tanpa spasi
if loop mau habis:
    print(temp)

```

```

text = "dengan ini saya nyatakan anda luar biasa"
print(f"Teks awal = {text}")
text1 = text.split()
word = str(input("search = "))
print("Teks yang setelah diproses: ", end="")
for i in text1:
    check(i, word)

```

Gambar 4.1 Kode program

Kurang lebih begitulah gambaran sederhana prosesnya. Perlu juga mengecek apakah panjang array kalimat yang akan dicek lebih kecil dari panjang string search. Jika lebih kecil maka akan langsung di print.

Fungs convert pada program digunakan untuk menkonversi array menjadi sebuah string.

IV. IMPLEMENTASI

A. Implementasi brute force

Brute force diimplementasikan pada program yang saya buat, karena program ini akan mencari kesamaan pada tiap tiap karakter yang banyaknya bisa ratusan, ribuan, bahkan ratusan ribu per dokumen. Pada program ini dokumen sudah harus terdefinisi pada kode program.

B. Source code program

```

def convert(arr):
    temp = ""
    for i in arr:
        temp+=i
    return temp

def check(sentence, word):
    if len(sentence) < len(word):
        print(sentence, end=" ")
    else:
        temp = ""
        temp2 = ""
        for i in range (len(sentence)):
            temp = temp+sentence[i]
            if temp == word:
                print(temp.upper(), end="")
                #print(Style.RESET_ALL, end="")
                temp2=""
            else:
                temp2 += sentence[i]
                if i == len(sentence)-1:
                    print(temp2, end="")
                    temp2=""
                    if len(temp2) == len(word):
                        print(temp2[0], end="")
                        temp2 = list(temp2)
                        temp2.pop(0)
                        temp2 = convert(temp2)
                    if len(temp) + 1 > len(word):
                        temp = list(temp)
                        temp.pop(0)
                        temp = convert(temp)
        print(end=" ")

```

C. Uji coba

A. Uji coba pertama:

```

tang@sultan:~$ /usr/bin/python3 "/home/tan/Program/Algo
rithm strategy/Mandiri 2/bruteforcesearch.py"
Teks awal = andi di rumah sajaa. Seharian bermain game
dengan adiknya
search = a
Teks yang setelah diproses: Andi di rumAh sAjaA. SeHar
iAn berMAin gAMe dengAn AdiknyaA tang@sultan:~$ █

```

Gambar 4.2 Uji coba ketiga

Pada uji coba diatas, saya mencari karakter a, dimana karakter a tersebut terdapat pada beberapa bagian di array kalimat yang sudah dikonversi. Jika diperhatikan, tidak ada karakter a yang menjadi output dalam bentuk lowercase.

B. Uji coba kedua:

```

tan@sultan:~$ /usr/bin/python3 "/home/tan/Program/Algo
rithm strategy/Mandiri 2/bruteforcesearch.py"
Teks awal = dengan ini saya nyatakan anda luar biasa
search = atak
Teks yang setelah diproses: dengan ini saya nyATAKan a
nda luar biasa tan@sultan:~$ █

```

Gambar 4.3 Uji coba kedua

Uji coba kedua ini saya mencari string atak pada array kalimat yang sudah dikonversi. Hasilnya, karakter "atak" dikeluarkan menjadi uppercase character.

C. Uji coba ketiga

```
tan@sultan:~$ /usr/bin/python3 "/home/tan/Program/Algo
rithm strategy/Mandiri 2/bruteforcesearch.py"
Teks awal = saya ingin pergi dengan teman saya
search = saya
Teks yang setelah diproses: SAYA ingin pergi dengan te
man SAYA tan@sultan:~$ █
```

Gambar 4.4 Uji coba ketiga

Uji coba ketiga, mencari karakter/kata “saya” pada array kalimat yang sudah terkonversi. Hasilnya berhasil, dan kata “saya” dikeluarkan dalam bentuk uppercase.

Dari ketiga percobaan tersebut, dapat kita lihat bahwa implementasi bruteforce pada program berjalan dengan baik pada pencarian kata ada pola pada kalimat. Pola yang dimaksud adalah karakter pada kata atau susunan huruf pada sebuah kata yang dianggap sama. Untuk contohnya dapat dilihat pada uji coba pertama, dan kedua.

V. KESIMPULAN

Setelah percobaan dan pengimplementasian algoritma brute force pada program search kali ini, dapat disimpulkan bahwa algoritma bruteforce terbukti cocok untuk mencari pola(pattern) dan kata pada sebuah kalimat. Algoritma ini mudah dimengerti karena pengimplementasiannya yang cukup mudah diterapkan ditambah dengan penggunaan bahasa pemrograman python yang syntax/penulisannya tidak terlalu membingungkan, simple, dan mudah dimengerti oleh programmer maupun bukan programmer.

UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas karunia, rahmat, dan ridho-Nya, makalah ini dapat diselesaikan tepat waktu dengan hasil percobaan yang cukup memuaskan.

Pada kali ini, penulis ingin menyampaikan terima kasih kepada Imam Ekowicaksono, M. Si., dan Winda Yulita, M.Cs. Yang telah memberikan ilmu yang bermanfaat bagi kehidupan manusia dan masyarakat. Proses pembuatan makalah ini juga dibuat berdasarkan materi yang diajarkan oleh dosen pengampu. Penulis juga ingin menyampaikan terima kasih pada orang-orang terdekat, karena dukungannya makalah ini dapat diselesaikan dengan baik dan tepat waktu.

Penulis juga sadar bahwa makalah ini masih jauh dari kata sempurna. Penulis juga ingin menyampaikan permohonan maaf jika ada kesalahan dalam penulisan makalah ini. Penulis sangat berharap adanya kritik dan saran untuk penyempurnaan makalah atau karya tulis berikutnya. Besar harapan penulis, semoga makalah ini bermanfaat bagi masyarakat dan pembaca

REFERENCES

- [1] <https://www.freecodecamp.org/news/brute-force-algorithms-explained/> Diakses pada 29 maret 2022
- [2] <https://www.ruangguru.com/blog/mengenal-matriks-dalam-matematika-pengertian-jenis-dan-transpose> Diakses pada 29 maret 2022
- [3] <https://www.geeksforgeeks.org/searching-algorithms/> Diakses pada 29 maret 2022
- [4] [https://informatika.stei.itb.ac.id/~rinald.munir/Stmik/2020-2021/Algoritma-Brute-Force-\(2021\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinald.munir/Stmik/2020-2021/Algoritma-Brute-Force-(2021)-Bag1.pdf) Diakses 29 maret 2022

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.

Lampung, 29 April 2022



SultanAhmad Alfath 120140046