

Penerapan Algoritma *Brute-Force* pada *Instagram Friend Suggestions*

Rania Dwi Fadhilah - 13520142

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail (gmail): 13520142@std.stei.itb.ac.id

Abstract—Instagram merupakan salah satu jejaring sosial yang masih sering digunakan oleh berbagai kalangan. Banyaknya pengguna yang menggunakan aplikasi ini membuat sang *developer* mengembangkan banyak fitur baru. Salah satu fitur yang terdapat di Instagram adalah fitur *friend suggestions* yang memberikan rekomendasi akun untuk *di-follow*. Terdapat banyak algoritma yang dapat digunakan untuk mengimplementasikan fitur ini. Pada pelajaran Strategi Algoritma IF2211, mahasiswa diajarkan tentang materi algoritma *Brute-Force*. Materi *Brute Force* ini dapat diaplikasikan pada pembuatan Instagram *Friend Suggestions* ini.

Keywords—brute force, stima, instagram.

I. PENDAHULUAN

Tidak dapat dipungkiri bahwa dunia maya kini menjadi wadah yang bermanfaat untuk banyak hal, terutama untuk berkoneksi dengan orang lain dan mendapatkan teman baru. Namun, apabila tidak digunakan dengan baik, maka hal ini dapat menimbulkan efek negatif bagi penggunanya.

Instagram merupakan salah satu sosial media yang sering digunakan oleh masyarakat sekarang. Sosial media ini memperbolehkan penggunanya untuk menyebarkan pesan dalam bentuk gambar/video. Seiring bertambahnya pengguna, bertambah banyak pula fitur yang disuguhkan oleh Instagram.

Salah satu fitur yang dibuat oleh Instagram adalah *friend suggestions*. Fitur ini menyediakan rekomendasi akun Instagram untuk *di-follow*. Rekomendasi yang disediakan ini merupakan akun yang banyak *di-follow* oleh akun-akun yang diikuti oleh si pengguna atau yang disebut *mutual*. Fitur ini menjadi salah satu opsi untuk meminimalisir efek negatif yang mungkin muncul karena mencari teman melalui jejaring sosial.

Terdapat berbagai macam algoritma yang dapat digunakan dalam mengimplementasikan fitur *Friend Suggestions* pada Instagram, salah satunya adalah algoritma *Brute-Force*. Oleh karena itu, dengan mempelajari *Brute-Force* secara lebih mendalam, akan dilakukan sebuah penelitian untuk menerapkan algoritma *Brute-Force* pada fitur Instagram ini.

II. LANDASAN TEORI

A. Algoritma *Brute-Force*

Algoritma *Brute Force* atau *exhaustive search* merupakan algoritma yang menggunakan pendekatan *straightforward* dalam memecahkan suatu persoalan. Algoritma ini dapat menyelesaikan suatu persoalan dengan sistematis, sederhana, mudah dimengerti, dan langsung. Namun, algoritma *Brute Force* cenderung kurang mangkus dan kurang “cerdas”. Selain itu, ketika menerima data masukan berukuran besar, algoritma ini berjalan dengan sangat lambat.

Algoritma ini biasanya didasarkan pada pernyataan pada persoalan dan definisi/konsep yang dilibatkan. Algoritma ini menyelesaikan permasalahan dengan sangat sederhana, langsung, dan cara yang jelas.

Brute Force merupakan algoritma yang membutuhkan volume komputasi yang besar dan waktu yang lama dalam menyelesaikan masalah sesuai dengan namanya yaitu “Force”. Tidak jarang algoritma *Brute Force* disebut sebagai algoritma yang naif.

Untuk memperbaiki kinerja pencarian, dapat diterapkan fungsi *heuristic* sehingga pencarian kemungkinan solusi tidak perlu dilakukan secara penuh. Heuristik sendiri mengacu pada teknik pemecahan masalah persoalan berbasis pengalaman. Namun, keoptimalan hasil tetap tidak dapat terjamin.

a. Kekuatan *Brute Force*

- Dapat diimplementasikan pada hampir Sebagian besar masalah (*wide applicability*)
- Algoritmanya sederhana dan mudah untuk dimengerti
- Dapat menghasilkan algoritma yang layak untuk beberapa masalah krusial seperti *searching*, *sorting*, *matching*, dan perkalian matriks
- Menghasilkan algoritma standar untuk tugas-tugas komputasi seperti penjumlahan/perkalian n buah bilangan, dan menentukan minimum atau maksimum dalam sebuah larik

a. Kelemahan Brute Force

- Algoritma yang dihasilkan kurang mangkus
- Umumnya bergerak lambat untuk masukan berukuran besar
- Tidak sekonstruktif/sekreatif strategi pemecahan masalah lainnya

c. Contoh Penggunaan Algoritma

Algoritma *Brute Force* dapat digunakan untuk mencari elemen terbesar dalam sebuah senarai. Implementasinya sendiri adalah sebagai berikut :

```
procedure CariElemenTerbesar(input  $a_1, a_2, \dots, a_n$  : integer, output maks : integer)
{ Mencari elemen terbesar di antara elemen  $a_1, a_2, \dots, a_n$ 
  Elemen terbesar disimpan di dalam maks.
  Masukan:  $a_1, a_2, \dots, a_n$ 
  Luaran: maks
}
Deklarasi
k : integer

Algoritma:
maks  $\leftarrow a_1$ 
for k  $\leftarrow 2$  to n do
  if  $a_k > maks$  then
    maks  $\leftarrow a_k$ 
  endif
endfor
```

Gambar 1 Contoh Penggunaan Algoritma

d. Penerapan Algoritma Brute-Force

Beberapa contoh penerapan algoritma *Brute Force* dalam kehidupan sehari-hari diantara lain adalah evaluasi polinom, pencocokan *string*, penyelesaian permainan sudoku, penyelesaian *Word Search Puzzle*, dan penyelesaian *Crossword Puzzle*.

B. Instagram Friend Suggestions

a) Instagram

Instagram merupakan sebuah aplikasi jejaring sosial berbagi foto dan video yang memungkinkan penggunanya membagikannya ke orang lain. Ciri khas dari Instagram adalah memotong foto menjadi berbentuk persegi seperti layaknya kamera polaroid. Instagram diakuisisi oleh Facebook dengan nilai sekitar \$1 miliar.



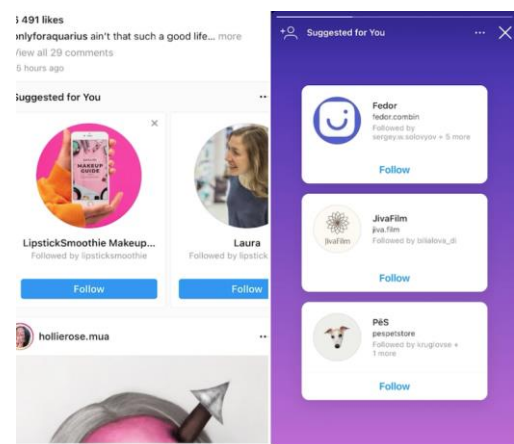
Gambar 2 Instagram

Terdapat banyak fitur yang diberikan oleh Instagram, antara lain pengikut sebagai target audiens, mengunggah foto (dengan menyediakan kamera, efek foto, judul foto, label foto, arroba, publikasi, dan sebagainya), tanda suka untuk menyukai akun milik pengguna lain, *explore* untuk melihat rekomendasi unggahan pengguna lain yang sekiranya cocok dengan pemilik akun, *friend suggestions* yang memberikan pilihan akun untuk di-follow, dan masih banyak lagi.

Saat ini, sudah ada lebih dari 8 juta pengguna dari seluruh belahan dunia yang memiliki Instagram. Seiring dengan bertambahnya pengguna Instagram, bertambah banyak pula fitur yang dikembangkan oleh aplikasi ini.

b) Friend Suggestions Feature

Salah satu fitur yang dimiliki oleh Instagram adalah *Friend Suggestions*. Fitur ini memperbolehkan penggunanya untuk mendapat rekomendasi akun untuk diikuti berdasarkan keadaan akun tersebut. Keadaan akun tersebut dapat berupa akun di sekitarnya maupun unggahan yang sering dilihat oleh pengguna. Instagram akan memberikan rekomendasi akun yang banyak diikuti oleh orang-orang yang sang pengguna ikuti.



Gambar 3 Friend Suggestions Instagram

Pengembangan dan penyediaan fitur ini mempermudah pengguna untuk mencari akun untuk diikuti. Selain itu, apabila pengguna berniat untuk memakai aplikasi jejaring sosial ini untuk mencari teman, maka fitur ini juga dapat meminimalisir adanya bahaya yang mungkin muncul dari mendapat teman baru melalui jejaring sosial. Hal ini disebabkan karena Instagram memberikan rekomendasi berupa akun yang diikuti pula juga oleh akun-akun lainnya atau kerap disebut *mutual*. Walaupun tidak menjamin, namun dengan adanya banyak akun sekitar yang juga mengikuti rekomendasi akun-akun tersebut, tentu lebih mudah pula untuk si pengguna akun utama mengenal lebih jauh terkait pengguna akun yang direkomendasikan dengan me-refer akun tersebut ke mereka yang sudah mengikuti sebelumnya.

III. IMPLEMENTASI DAN PENGUJIAN

A. Struktur Data dan Inisialisasi

Dalam makalah ini, bahasa pemrograman yang digunakan untuk menyelesaikan permasalahan adalah bahasa Python 3.x dan tidak ada *library* yang digunakan.

Dalam implementasinya sendiri, terdapat 2 struktur data yang digunakan, antara lain :

1. List

List merupakan struktur data yang dapat diubah dan terurut urutan elemennya. Setiap elemen yang terdapat pada list disebut sebagai *value*. Daftar seluruh *value* ini disimpan diantara tanda kurung siku [].

Penerapan list ini sendiri digunakan untuk menyimpan data dari file, menyimpan nama-nama akun yang sudah di cek, menyimpan *pair*, dan menyimpan nama-nama rekomendasi akun.

2. Pair

Pair merupakan struktur data yang memiliki 2 nilai yang saling berkorespondensi. Struktur data ini digunakan untuk menyimpan data-data akun yang saling mengikuti dan juga data jumlah *mutual* akun.

B. Implementasi Algoritma

Dalam mengimplementasikan algoritma *Brute-Force* untuk mencari *friend suggestions*, diperlukan beberapa tahap, yaitu :

1. Menerima sebuah list berisi daftar akun yang saling mengikuti satu sama lain. List ini didapatkan dari sebuah file dan akan diekstrak dengan cara berikut :

```
function handleFile(fileName : string) → list of Pair
{ I.S. File terdefinisi dan tidak kosong }
{ F.S. Mengembalikan list berupa Pair yang memiliki 2
elemen yang merupakan nama akun yang saling mengikuti
satu sama lain }
```

KAMUS LOKAL

List : list of Pair
Line : String
temp : list of String

ALGORITMA

```
if exists(fileName) then
    open input(fileName)
    List ← []
    repeat
        line ← readLine(fileName)
        temp ← line.split
        append temp to List
    until eof
→ List
```

2. Menerima nama sebuah *user* yang ingin dijadikan acuan untuk dicarikan *friend suggestions*-nya.

3. Membuat sebuah list kosong untuk menyimpan daftar akun yang diikuti oleh teman-teman akun acuan dan juga list kosong untuk menyimpan nama akun yang difollow oleh akun acuan

4. Melakukan iterasi pada seluruh elemen berbentuk *pair* yang terdapat pada list yang dihasilkan oleh file

a. Mencari elemen yang memiliki *value* berupa nama akun acuan

b. Menyimpan *value* dari pasangan elemen sebelumnya ke dalam list kosong penyimpanan akun yang difollow oleh akun acuan

c. Melakukan iterasi terhadap list lagi untuk mencari elemen yang salah satu *value*-nya adalah pasangan dari akun acuan yang didapatkan pada tahap sebelumnya. dan menyimpan seluruh elemen pada list kosong untuk menyimpan nama akun rekomendasi

5. Menghilangkan akun *mutual* (sudah diikuti oleh akun acuan) dari list yang menyimpan nama-nama akun rekomendasi.

6. Mencari jumlah *mutual* yang dimiliki oleh list dengan akun acuan. Implementasinya adalah sebagai berikut :

```
function notInList(name : String, List : list of Pair) →
boolean
{ I.S. List terdefinisi dan tidak kosong, nama terdefinisi }
{ F.S. Mengembalikan true apabila name tidak ada di list dan
false apabila nama ada di list }
```

KAMUS LOKAL

ALGORITMA

```
repeat
    if (name = List Element) then
        → false
    until end of list
→ true
```

```
function notInPair(name : String, List : list of Pair) →
boolean
{ I.S. List terdefinisi dan tidak kosong, nama terdefinisi }
{ F.S. Mengembalikan true apabila name tidak ada di list dan
false apabila nama ada di list }
```

KAMUS LOKAL

ALGORITMA

```
repeat
```

```

    if (name = List Element (0)) then
        → false
    until end of list
    → true

```

function mutualNum(List : list of String) → list of Pair
 { I.S. List terdefinisi dan tidak kosong }
 { F.S. Mengembalikan list berupa Pair yang memiliki 2 elemen dengan elemen berupa nama akun (bersifat unik) dan elemen ke-2 berupa jumlah kemunculan nama akun pada List }

KAMUS LOKAL

Pair : list of Pair
 Count : integer

ALGORITMA

```

Pair ← []
for i in List
    Count ← 0
    if (notInList (i, Pair) then
        for j in list
            if (j = i) then
                Count ← Count + 1
        append ([i, Count]) to Pair

→ Pair

```

7. Mengurutkan list jumlah mutual dari yang terbesar hingga terkecil dengan implementasi sebagai berikut :

procedure sortPair(input pair : list of Pair)
 { I.S. Pair terdefinisi dan tidak kosong }
 { F.S. Mengembalikan Pair yang elemen ke-2nya sudah terurut mengecil }

KAMUS LOKAL

Temp : Pair of String

ALGORITMA

```

for i in range (len(pair))
    for j in range (len(pair) - 1)
        if (pair[j][1] < pair[j+1][1]) then
            Temp ← pair[j]
            pair[j] ← pair[j+1]
            pair[j+1] ← Temp

```

Penggunaan algoritmanya sendiri dalam program ini adalah sebagai berikut :

```

# Menerima input
fileName = input("Enter the file name: ")
List = handleFile(fileName)
user = input("Enter the user: ")

```

```

start = time.time()

# Mencari hasil
Recom = []
Mutual = []
i = 1
for pair in List :
    # Cek apakah user ada di pair
    if (pair[0] == user or pair[1] == user) :

        # Kasus 1, user ada di pair[0]
        if (pair[0] == user) :
            if (notInList(pair[1], Mutual)):
                Mutual.append(pair[1])
            for following in List :
                if (following[0] != user and
                    following[1] != user) :
                    if (following[0] == pair[1]):

Recom.append(following[1])
                        elif (following[1] ==
pair[1]) :

Recom.append(following[0])

        # Kasus 2, user ada di pair[1]
        elif (pair[1] == user) :
            if (notInList(pair[0], Mutual)):
                Mutual.append(pair[0])
            for following in List :
                if (following[0] != user and
                    following[1] != user) :
                    if (following[0] == pair[0])

:

Recom.append(following[1])
                        elif (following[1] ==
pair[0]) :

Recom.append(following[0])

# Menghilangkan akun mutual (sudah diikuti oleh
user) dari recom
newRecom = []
for i in range(len(Recom)):
    if (notInList(Recom[i], Mutual)):
        newRecom.append(Recom[i])

# Memberikan output
print("\n" + user + " Instagram")
print("Suggested for you: ")
i = 1
RecomRes = mutualNum(newRecom)
sortPair(RecomRes)
for suggested in RecomRes :

```

```

print(str(i) + ". " + suggested[0])
i += 1
print(suggested[0] + " and you have " +
str(suggested[1]) + " mutual friends")

end = time.time()
print("\nTime: " + str(end - start) + " s")

```

C. Pengujian

Pengujian dilakukan dengan menggunakan kode-kode program yang telah terlampir pada sub-bab sebelumnya. Untuk dapat menguji kode, dibutuhkan sebuah file yang berisi list nama akun yang saling mengikuti satu sama lain.

Isi dari file yang akan dicek adalah sebagai berikut :

```

akun1 akun2
akun3 akun1
akun3 akun2
akun2 akun4
akun3 akun4
akun4 akun5
akun2 akun5
akun5 akun6
akun7 akun9
akun8 akun10
akun10 akun9
akun10 akun1
akun1 akun5
akun4 akun10
akun3 akun30
akun2 akun6

```

Kemudian, akan dipilih salah satu akun yang ingin dijadikan sebagai acuan untuk mencari rekomendasi pertemanan. Pada pengujian pertama, akan dipilih akun 1 sebagai acuan.

```

Enter the file name: 2.txt
Enter the user: akun1

akun1 Instagram
Suggested for you:
1. akun4
akun4 and you have 4 mutual friends
2. akun6
akun6 and you have 2 mutual friends
3. akun30
akun30 and you have 1 mutual friends
4. akun8
akun8 and you have 1 mutual friends
5. akun9
akun9 and you have 1 mutual friends

Time: 0.0035669803619384766 s

```

Gambar 4 Pengujian 1

Dari gambar di atas, terlihat bahwa ada 5 rekomendasi akun yang diberikan algoritma untuk pengguna terurut mengecil sesuai jumlah teman yang sudah mengikuti akun tersebut. Berdasarkan file yang diterima, dapat ditelaah bahwa akun1 (akun acuan) mengikuti akun2, akun3, akun10, dan akun5. Sedangkan akun-akun tersebut mengikuti:

akun2 : akun3, akun4, akun5, akun6

akun3 : akun2, akun4, akun30

akun10 : akun8, akun9

akun5 : akun2, akun4, akun6

Dari list diatas, dengan mengkurasi akun-akun yang sudah diikuti oleh akun acuan juga akun-akun yang tertulis lebih dari 1x, maka dihasilkan list rekomendasi sebagai berikut :

1. akun4 = 3 mutual friends (akun2, akun3, akun5)
2. akun6 = 2 mutual friends (akun2, akun5)
3. akun30 = 1 mutual friends (akun3)
4. akun8 = 1 mutual friends (akun10)
5. akun9 = 1 mutual friends (akun10)

Waktu yang dibutuhkan untuk mencari data menggunakan implementasi algoritma *Brute-Force* adalah 0.003567 s. Durasi pencarian ini tergolong cukup lama untuk sebuah data yang berukuran kecil.

Pengujian ke-2 akan dilakukan dengan menggunakan akun10 sebagai akun acuan, sehingga hasil yang didapatkan adalah sebagai berikut :

```

Enter the file name: 2.txt
Enter the user: akun10

akun10 Instagram
Suggested for you:
1. akun2
akun2 and you have 2 mutual friends
2. akun3
akun3 and you have 2 mutual friends
3. akun5
akun5 and you have 2 mutual friends
4. akun7
akun7 and you have 1 mutual friends

Time: 0.003924369812011719 s

```

Gambar 5 Pengujian 2

Berdasarkan gambar di atas, terlihat bahwa ada 4 akun yang direkomendasikan oleh algoritma. Apabila dicocokkan dengan file, maka dapat diketahui bahwa akun10 mengikuti akun8, akun9, akun1, dan juga akun4. Kemudian, program merekomendasikan keempat akun tersebut karena akun2 diikuti oleh akun4 dan akun1, akun3 diikuti oleh akun1 dan akun4, akun5 diikuti oleh akun1 dan akun4, akun7 diikuti oleh akun9. Waktu yang dibutuhkan kurang lebih sama dengan pengujian sebelumnya.

IV. KESIMPULAN

Algoritma *Brute Force* dapat diimplementasikan dalam berbagai hal di kehidupan sehari-hari, contohnya adalah pada fitur *Friend Suggestions* di Instagram. Setiap pasangan akun yang saling mengikuti satu sama lain akan dikumpulkan dalam sebuah *List* yang kemudian akan diproses menggunakan algoritma *Brute Force* untuk mencari rekomendasi akun yang paling banyak difollow oleh teman-teman pengguna akun utama. Namun, apabila diperhatikan, algoritma membutuhkan waktu yang cukup lama untuk bisa menghasilkan list rekomendasi. Oleh sebab itu, diperlukan adaptasi lebih lanjut agar program bisa jalan dengan lebih cepat dan efisien.

UCAPAN TERIMA KASIH

Pertama-tama, penulis mengucapkan puji dan syukur kepada Tuhan yang Maha Esa, karena atas berkah dan rahmat yang diberikan-Nya, penulis dapat menyelesaikan masalah ini dengan tepat waktu. Penulis juga mengucapkan terima kasih yang sebesar-besarnya kepada seluruh dosen pengampu mata kuliah IF2211 Strategi Algoritma, terutama Ibu Dr. Masayu Leylia Khodra, S.T., M.T. sebagai guru yang mengajar kelas K01 atas bimbingannya selama satu semester ini. Tak lupa, penulis juga mengucapkan terima kasih kepada keluarga dan teman-teman yang telah memberikan dukungan, baik dari segi moral maupun material sehingga dapat membantu penulis dalam menyelesaikan makalah ini. Semoga makalah ini dapat memberi manfaat kepada para pembaca.

VIDEO LINK AT YOUTUBE (*Heading 5*)

<https://youtu.be/-HwyvQrjCto>

DAFTAR REFERENSI

- [1] [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-\(2022\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-(2022)-Bag1.pdf) diakses pada 20 Mei 2022
- [2] [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-\(2022\)-Bag2.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-(2022)-Bag2.pdf) diakses pada 20 Mei 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.

Bandung, 22 Mei 2022



Rania Dwi Fadhillah - 13520142