Implementasi Sederhana Algorithm Brute Force pada pencarian parkiran kosong

Naufal Rotif Dewanto 120140107 (*Author*)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Sumatera, Jl. Terusan Ryacudu Lampung
E-mail (gmail): naufal.120140107@student.itera.ac.id

***Abstract*—Parkir merupakan kondisi dimana suatu kendaraan berhenti dan pengemudinya meninggalkan kendaraan, setiap fasilitas umum mempunyai parkiran untuk para pengendar jika ingin mengunjungi fasilitas tersebut. Parkir mempunyai beberapa jenis untuk berbagai kendaraan dan bagaimana status parkiran tersebut, apakah terisi suatu kendaraan atau tidak, kita bisa mencarinya dengan menggunakan algoritma ini yaitu algoritma brute force.**

Keywords—brute force; parkiran; algoritma; kendaraan

I. PENDAHULUAN'

Kendaraan adalah sebuah transportasi yang dapat mengangkut orang ataupun benda, kendaraan banyak jenisnya bisa dibedakan dari banyak roda, dimana kendaraan itu dikendara, dan hingga fungsinya. Seiring perkembangan zaman kendaraan lebih maju yang berawal dibantu oleh tenaga hewan seperti kuda lalu tenaga uap dan sekarang tenaga elektrik.

Parkir adalah dimana kendaraan berhenti karena pengemudinya meninggalkan kendaraanya, setiap fasilitas mempunyai parkiran untuk para pengunjungnya ingin memarkirkan kendaraannya supaya lebih aman dan mudah dicapai.

Algorima adalah sekumpulan instruksi terstruktur yang akan diimplementasikan kedalam program suatu komputer untuk menyelesaikan suatu permasalahan komputasi tertentu. Algoritma merupakan prosedur atau sebuah langkah – langkah dalam perhitungan.

Algorimat Brute Force adalah algorima yang mencari solusi dari sebuah larik dengan cara terseluruh tanpa meninggalkan satupun.

Kita bisa mencari parkiran kosong menggunakan algoritma brute force dimana kita melihat apakah status dari sebuah parkiran itu terisi atau tidak terisi. Menelusuri semua parkiran yang ada.

II. TEORI DASAR

A. Brute Force

Algoritma *brute force* merupakan yang biasa digunakan untuk mencari sebuah solusi dari berbagai kasus, Algoritma brute force bersifat pendekatan yang lempeng atau straightforward. Memecahkan suatu masalah dengan sangat sederhana, langsung, jelas caranya. Algoritma brute force umum dikatakan tidak cerdas dan tidak mangkus, karena membutuhkan komputasi yang besar dan waktu yang relative lama untuk menyelesaikan masalah biasa disebut algoritma naïve (naif).

Algoritma brute force cocok jika ada persoalan yang berukuran kecil karena sederhana dan implementasinya mudah, sering digunakan untuk basis pembanding pada algoritma lainnya yang lebih mangkus. Walaupun problem solvingnya yang mangkus tetapi banyak persoalan yang bisa diselesaikan oleh algoritma brute force, itu merupakan kelebihan brute force.

Kekuatan atau kelebihan algoritma brute force

- Algoritma brute force bisa diimplementasikan untuk memecahkan hampir seluruh masalah yang ada.
- Brute force merupakan algoritma sederhana dan mudah dimengerti
- Merupakan algoritma yang layak untuk berbagai masalah penting contohnya pencarian, pencocokan string, perkalian matriks, penguruan.
- Menghasilkan algoritma baku atau standard untuk tugas komputasi layaknya penjumlahan/perkalian n buah bilangan, mencari elemen minimum dan maksimum dalam sebuah set atau larik

Kelemahan algoritma brute force

- Jarang menghasilkan algoritma yang mangkus atau efektif.
- Jika sebuah persoalan yang dicari berukuran besar maka tidak disarankan menggunakan algoritma ini karena bersifat lambat dalam penyelesaiannya..
- Algoritma ini tidak sekreatif atau sekontruktif untuk strategi pemecahan masalah lainnya.

B. Parkiran dan kendaraan

Parkiran sebuah tempat dimana kendaraan bisa berhenti dan pengemudi bisa meninggalkan kendaraanya secara aman. Parkir tidak secara sembarangan harus membutuhkan tempat dan dalam tempat itu harus mempunyai lahan kosong untuk bisa ditempati kendaraan.

Gambar 2.1 Tempat Parkir

https://www.parkpca.com/help-improve-communities-with-first-rate-parking-management-services/

Kendaraan sendiri merupakan alat transportasi untuk menangkut manusia maupun barang bisa digunakan untuk mengantar dari satu lokasi ke lokasi lainnya, untuk mengunjungi suatu tempat kita membutuhkan tempat untuk menyimpan kendaraan secara umum maka kita memerlukan sebuah tempat parkiran kosong.

Kita bisa mencari parkiran disebuah lokasi yang kita kunjungi dengan fasilitas yang sudah disedia dengan menggunakan berbagai cara dari valet hingga manual. Kita juga bisa membantu pencarian menggunakan sebuah alat yang akan membantu kita.

Gambar 2.2 Detection Parking Lot

https://www.behance.net/gallery/29828109/Deep-Learning-Automatic-Parking-Lot-Classification/modules/192024701

Sequential Search merupakan salah satu algoritma pencarian pada metode brute force dimana dia akan mencari variable pada suatu larik yang disediakan, dari larik pertama sampai larik terakhir dan jika tidak ditemukan di awal, algoritma ini bisa kita gunakan untuk membantu dalam penyelesaiaan masalah pencarian lahan kosong tersebut.

III. ANALISIS DAN PEMBAHASAN

Pada tugas kali saya mengambil topik pencarian parkiran kosong menggunakan metode algoritma brute force salah satunya yaitu sequential search, dimana setiap larai di bandingkan dengan value yang ingin di cari, pencarian akan selesai atau memberitahu jika value atau variable ditemukan pada larik tersebut.

Saya akan melakukan beberapa uji coba dengan program kode yang saya buat, program akan mencari value pada larik yang sudah disediakan, dan akan melakukan aksi apa bila value ditemukan.

A. Sequential Search

Mencari sebuah variable pada larik bisa kita lakukan dengan banyak algoritma search, tetapi kali ini kita akan mencarinya menggunakan algoritma brute force yang bernama sequential search.

Untuk menggunakannya kita akan membuat sebuah larik atau array pada program, larik atau array ini nanti kita isi menggunakan fungsi untuk membuat array teresebut terisi dengan angka random 0 sampai 1.

Disini saya membuat permisalan dimana sebuah parkiran adalah larik atau array, dan status parikira terisi atau kosong yaitu dengan angka 0 dan 1. Angka 1 adalah dimana status parkiran terisi dan angka 0 dimana status parkiran kosong.

Jika sudah berhasil membuat dan mengisi array, kita akan melakukan pencarian menggunakan algoritma sequential search, dimana kita akan melihat semua isi dari array, dan menampilkan penjelasannya. Setelah semua array berhasil di telusuri kita dapat mengetahui status dari sebuah parkiran tersebut.

Berikut Pseucode dari sequential search

```
procedure PencarianBeruntun(input a1, a2, ..., an : integer, x : integer; output idx : integer)
{ Mencari elemen bernilai x di dalam senarai a1, a2, ..., an. Lokasi (indeks elemen) tempat x
ditemukan diisi ke dalam idx. Jika x tidak ditemukan, maka idx diisi dengan 0.
  Masukan: a1, a2, ..., an
  Luaran: idx
}
Deklarasi
  k : integer

Algoritma:
  k←1
  while (k < n) and (ak ≠ x) do
    k ← k + 1
  endwhile
  { k = n or ak = x }

  if ak = x then   { x ditemukan }
    idx ← k
  else
    idx ← –1       { x tidak ditemukan }
  endif
```

Gambar 3.1 Pseudo Code

https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Brute-Force-(2021)-Bag1.pdf

B. Python

Python adalah salah satu Bahasa pemrograman yang saya gunakan kali ini gunakan, Python merupakan salah satu Bahasa yang mendekati Bahasa manusia aslinya, yang artinya Bahasa python mudah dimengerti untuk pemula.

Banyak modul pada python yang bisa kita gunakan untuk membantu pemrograman yang sudah ada pada python, salah satu contohnya adalah math, random , numpy dan lain lain. Kali ini saya menggunakan modul random dan numpy.

C. Random

Modul Random merupakan salah satu modul pada python yang berfungsi sebagai menciptakan random nomor yang tercipta dari pseudo random. Saya menggunakannya untuk mengisikan list yang sudah dibuat, dengan angka 0 dan 1.

Dengan menuliskan import random kita sudah bisa mengakses modul ini dan bisa menggunakannya pada python. Ada banyak method pada random module dari seed,getstate,setstate,randint, randrange dan lain lain. Tiap method mempunyai fungsi dan ciri khas tersendiri dari berbagai methodnya, masih banyak lagi yang bisa kita bahas tentang module random tetapi kali ini kita menggunakan method random randint.

D. Numpy

Numpy adalah python library yang biasa digunakan sebagai arrays, juga berfungsi sebagai domain linear algebra dan matrix. Saya menggunakan numpy karena untuk bertujuan array harus diperlukan flekseblitas lebih dari pada tradisional python list.

Untuk menggunakan fungsi numpy untuk pertama kali kita diharuskan untuk menginstal modul ini agar bisa digunakan nantinya, setelah itu kita bisa menuliskan import numpy untuk membuat array.

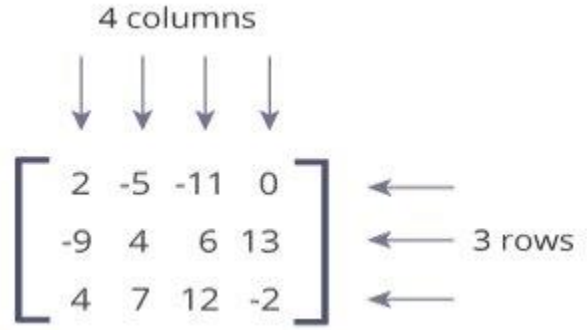
Gambar 3.2

https://numpy.org/images/logo.svg

Untuk membuat array kita bisa menuliskan nump array zeros(n) dimana sebuah array matrix berukuran n berisikan angka 0 yang nantinya bisa kita ubah menggunakan fungsi random.

E. Matriks

Matriks merupakan susunan bilangan yang terdiri dari kolom dan baris. Setiap matriks berdimensi M x N dimana membentuk persegi atau persegi panjang. Dengan menggunakan matriks bisa menambah serta membantu variasi dari pengerjaan tugas ini.



Gambar 3.3 Matrix

https://www.programiz.com/python-programming/matrix

IV. IMPLEMENTASI

Dengan awal membuat modul array atau larik menggunakan numpy, yaitu dengan cara memasukan import numpy ke dalam program, setelah itu kita akan membuat menu pilihan dimana dimensi dari parkiran atau arraynya berdimensi 1 dimensi atau 2 dimensi user akan memilih antara kedua pilihan tersebut.

Setelah itu kita akan membuat inputan dimana nanti user akan memasukan banyaknya kolom maupun baris sesuai pilihan yang dipilih tadi.

Jika banyaknya array sudah dimasukan kita akan memasukan angka antara 0 dan 1 dengan modul random. Kita menggunakan perulangan for dengan range array, lalu menginisialisasikan array satu persatu menggunakan fungsi random.randint(0,1) program akan memasukan angka 0 hingga 1 secara random sesuai pseudo random.

Setelah semua array berhasil terisi kita bisa mulai mencari status dari isi parkiran tersebut dengan menyatakan bahwa angka 1 dimana sebuah parkiran terisi dan angka 0 dimana sebuah parkiran kosong.

Dengan menggunakan perulangan kita bisa menelusuri array 1 per 1 ditambah fungsi if yang dimana mengecek isinya apakah isi aray itu berisi satu atau bukan. Jika suatu array berisi angka 1 akan mengeluarkan outpot “Parkiran [1...2…n] terisi” dan akan melakkukan fungsi while jika fungsi if tidak terlaksanakan dan mengeluarkan outpot “Parkiran[1…2...n] kosong”.

Setelah semua array berhasil ditelusuri program akan meminta inputan apakah ingin melakukan uji coba ulang, jika iya maka bisa memilih pilihan kedua yang tersedia, jika tidak bisa menginputkan angka 3 atau keluar dari program maka program selesai. Berikut adalah implementasi simple saya untuk mengecek apakah sebuah parkiran tersebut terisi atau tidak.

V. KODE PROGRAM

```
import random
import numpy as np

print("1. Parkiran / Array 1D")
print("2. Parkiran / Array 2D")
print("3. Exit")
pilih = int(input("Pilih Parkiran: "))

while(pilih != 3):
   if pilih == 1:
      n = int(input("Banyak Kolom: "))
      parkiran = np.zeros(n)

      for i in range (n):
         parkiran[i] = random.randint(0,1)
      print("")
      for i in range (n):
         if parkiran[i] == 1:
            print("Parkiran[",i+1,"] Terisi")
         else:
            print("Parkiran[",i+1,"] Kosong")
   elif pilih  == 2:
      n = int(input("Banyak Kolom: "))
      m = int(input("Banyak Baris: "))
      parkiran = np.zeros((n,m))

      for i in range (n):
         for j in range (m):
            parkiran[i][j] = random.randint(0,1)
      print("")
      for i in range (n):
         for j in range (n):
            if parkiran[i][j] == 1:
               print("Parkiran[",i+1,"]","[",j+1,"] Terisi")
            else:
               print("Parkiran[",i+1,"]","[",j+1,"] Kosong")
   print("")
   print("1. Parkiran / Array 1D")
   print("2. Parkiran / Array 2D")
   print("3. Exit")
   pilih = int(input("Pilih Parkiran: "))
```

VI. UJI COBA DAN HASIL

A. Uji Coba Pertama

Pada uji coba yang pertama saya ingin melakukan percobaan menggunakan array atau matriks dimensi satu.

Disini saya memasukan banyak array berkolom sebanyak 5 array.

Terlihat array pertama berisi angka 1 yang menyebabkan output 1 lalu untuk array kedua berisikan angka 1 juga yang mengoutputkan terisi, pada array ketiga berisikan angka 0 yang mengoutputkan kosong, pada array keempat

```
1. Parkiran / Array 1D
2. Parkiran / Array 2D
3. Exit
Pilih Parkiran: 1
Banyak Kolom: 5

Parkiran[ 1 ] Terisi
Parkiran[ 2 ] Terisi
Parkiran[ 3 ] Kosong
Parkiran[ 4 ] Kosong
Parkiran[ 5 ] Terisi
```

Gambar 6.1 Uji Coba Pertama

B. Uji Coba Kedua

Pada Uji coba yang kedua saya tetap mencoba percobaan menggunakan array berdimensi satu, tetapi menggunakan jumlah array yang berbeda yaitu 9 array

```
1. Parkiran / Array 1D
2. Parkiran / Array 2D
3. Exit
Pilih Parkiran: 1
Banyak Kolom: 9

Parkiran[ 1 ] Terisi
Parkiran[ 2 ] Terisi
Parkiran[ 3 ] Kosong
Parkiran[ 4 ] Terisi
Parkiran[ 5 ] Terisi
Parkiran[ 6 ] Terisi
Parkiran[ 7 ] Kosong
Parkiran[ 8 ] Kosong
Parkiran[ 9 ] Kosong
```

Gambar 6.2 Percobaan Kedua

Pada uji coba kali ini array pertama berisikan 1 yang berarti parkiran terisi dan array kedua juga berisikan 2 parkiran terisi, selanjutnya pada parkiran ketiga adalah parkiran kosong, dan array 4, 5, serta array 6 berisi angka 1 yang berarti parkiran terisi, , untuk array 7, 8 dan 9 terisi angka 0 berarti parkiran kosong dan bisa ditempati.

C. UJI COBA KETIGA

Pada Uji Coba ketiga kita akan mencoba menggunakan array 2 dimensi dengan kolom 2 dan baris 2 yang berarti matriks 2 x 2.

```
1. Parkiran / Array 1D
2. Parkiran / Array 2D
3. Exit
Pilih Parkiran: 2
Banyak Kolom: 2
Banyak Baris: 2

Parkiran[ 1 ] [ 1 ] Terisi
Parkiran[ 1 ] [ 2 ] Kosong
Parkiran[ 2 ] [ 1 ] Kosong
Parkiran[ 2 ] [ 2 ] Kosong
```

Gambar 6.3 Uji Coba ketiga

Pada kali ini array (1,1) berisi angka 1 yang berarti parkiran terisi dan array (1,2) berisi angka 0 yang berarti parkiran kosong selanjutnya pada array (2,1) berisi angka 0 dimana artinya parkiran kosong, begitu juga pada array (2,2) berisi angka 0 berarti parkirannya kosong tidak ada mobil terparkir.

D. Uji Coba Keempat

Pada Uji Coba Keempat kali ini sama dengan percobaan ke tiga, dengan banyak array 3 kolom dan 3 baris yang berarti 3 x 3.

```
1. Parkiran / Array 1D
2. Parkiran / Array 2D
3. Exit
Pilih Parkiran: 2
Banyak Kolom: 3
Banyak Baris: 3

Parkiran[ 1 ] [ 1 ] Terisi
Parkiran[ 1 ] [ 2 ] Terisi
Parkiran[ 1 ] [ 3 ] Kosong
Parkiran[ 2 ] [ 1 ] Terisi
Parkiran[ 2 ] [ 2 ] Terisi
Parkiran[ 2 ] [ 3 ] Terisi
Parkiran[ 3 ] [ 1 ] Kosong
Parkiran[ 3 ] [ 2 ] Terisi
Parkiran[ 3 ] [ 3 ] Terisi
```

Gambar 6.4 Percobaan keempat

Pada array (1,1) & (1,2) array terisi angka 1 yang berarti parkiran sudah terisi, selanjutnya array (1,3) array terisi angka 0 dimana parkiran kosong, lalu array (2,1), (2,2), (2,3) berisi angka 1 yang berarti parkiran terisi array selanjutnya yaitu array (3,1) array berisi angka 0 yang berarti parkiran kosong dan array (3,2) & (3,3) berisi angka 1 yang berarti parkiran terisi oleh mobil.

VIDEO LINK AT YOUTUBE

https://youtu.be/bblKqRT83W8.

KESIMPULAN

Dengan menggunakan algoritma Brute Force kita bisa menyelesaikan persoalan mencari parkiran kosong secara sederhana,, algoritma brute force menyelesaikan masalah dengan menyeluruh sesuai dengan sifatnya penyelesaian masalah dengan straightforward, sederhana dan simple.

Algoritma Sequntial Searach yang merupakan bagian dari algoritma brute force bisa melakukan pencarian yang secara detail dimana tidak meninggalkan satu array pun. Penggunaan sequential search bisa lebih dimaksimalkan jika sebuah permasalahannya tidak berinput besar dimana menyebabkan pencarian tidak begitu efisien dan membutuhkan waktu yang lama.

Saya berharap bisa lebih berkembang lagi dan bisa mengembangkannya lebih lagi dimana pencarian tidak menggunakan variable tetapi menggunakan suatu objek yang bisa membantu permasalahan ini, dengan bantuan dari algoritma brute force maupun algoritma lainnya

UCAPAN TERIMA KASIH

Begitulah implementasi sederhana brute force pada pencarian parkiran untuk tugas kali ini, semoga penjelasan yang saya sampaikan bisa mudah dicerna dan bisa digunakan untuk selanjutnya ataupun bisa dikembangkan lebih lanjut lagi pada penelitian saya kali ini, bila ada salah kata atau kurang kata mohon maaf. Sekian Terimakasih.

References

[1] "vehicle, n.", OED Online, Oxford University Press, November 2010

[2] "Global Parking Index 2019". Parkopedia. Retrieved 2021-01-21.

[3] "Definition of ALGORITHM". Merriam-Webster Online Dictionary. Archived from the original on February 14, 2020. Retrieved November 14, 2019.

[4] https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Brute-Force-(2021)-Bag1.pdf Diakses 24/03/2022

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, 24 Maret 2022

Naufal Rotif Dewanto
120140107