# LAPORAN

# TUGAS BESAR

Disusun untuk memenuhi tugas

Mata Kuliah: Strategi Algoritma
Dosen Pengampu: Dr. EMA RACHMAWATI, S.T., M.T.

Oleh:

MUHAMMAD RAFI ANDEO PRAJA (1301200278)
M NAUFAL RIFQI RAMDHANI (1301201572)
RIZKY FERDIAN PRASETYO (1301204229)

**KELAS IF-44-01**
**JURUSAN S1 INFORMATIKA**
**FAKULTAS INFORMATIKA**
**UNIVERSITAS TELKOM**

# DAFTAR ISI

# ABSTRAK

Muhammad Rafi Andeo Praja. M Naufal Rifqi Ramdhani. Rizky Ferdian Prasetyo 2022. Analisis mengenai permasalahan kasus Policeman Catch Thieves menggunakan strategi Brute Force dan Greedy

Analisis berikut bertujuan untuk mempelajari ilmu dasar dari strategi Brute Force dan Greedy lalu diimplementasikan kedalam suatu permasalahan yaitu Policeman Catch Thieves serta diberikan sebuah kesimpulan bahwa manakah strategi yang lebih baik berdasarkan waktu kompleksitas yang dihasilkan

Hasil analisis menunjukkan bahwa penggunaan strategi greedy tentu lebih optimal dibandingkan dengan strategi brute force, tetapi strategi greedy tidak selalu menghasilkan output yang benar. Perbedaan dari waktu kompleksitas dapat diketahui cukup terlihat jelas bahwa strategi greedy memiliki waktu yang lebih cepat untuk menyelesaikan permasalahan Policeman Catch Thieves berikut. Maka dari itu, permasalahan policeman catch thieves lebih baik dipecahkan menggunakan strategi brute force dibandingkan dengan greedy dari segi hasil

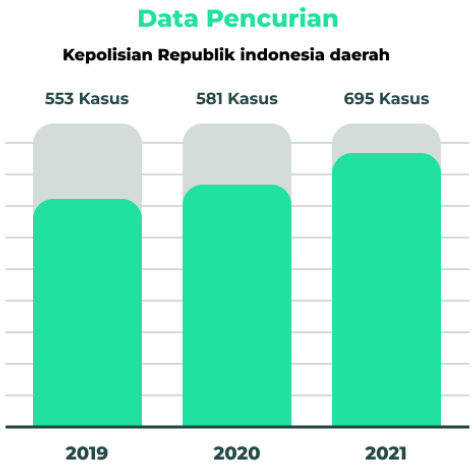
**Kata Kunci: Greedy, Brute Force, Policeman Catch Thieves, Algoritma, Solusi.**

# BAB I

# PENDAHULUAN

## A. Latar Belakang Masalah

Tindak pidana secara sederhana dapat dikatakan sebagai perbuatan yang pelakunya seharusnya dapat dipidana. Tindak pidana pencurian adalah pengambilan properti miliki orang lain secara tidak sah tanpa seizin pemilik. Perkembangan tindak kejahatan terutama tindak pencurian selama 3 tahun terakhir telah mengalami peningkatan.




Sumber *:* *http://bappeda.jogjaprov.go.id/dataku/data_dasar/index/447-jumlah-kasus-pencurian?id_skpd=39*

Dari data yang telah di dapat di kepolisian republik indonesia daerah dalam 3 tahun telah mengalami peningkatan. Hal ini dapat disebabkan oleh beberapa faktor diantaranya adalah pengaruh lingkungan, adanya kesempatan untuk melakukan tindak pidana tersebut dan kurangnya kesadaran terhadap hukum dari pencuri serta dapat disebabkan oleh faktor sosial lainnya.

# BAB II

# DASAR TEORI

## A. Pengertian Strategi Brute Force

Algoritma Brute Force merupakan sebuah pendekatan yang lempang (straightforward) untuk memecahkan suatu masalah, biasanya didasarkan pada pernyataan masalah (problem statement) dan definisi konsep yang dilibatkan. Algoritma Brute Force memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang jelas (obvious way).

Algoritma Brute Force ini jarang menghasilkan algoritma yang mangkus. Beberapa algoritmanya juga sangat lambat hingga tidak dapat diterima, karena algoritma brute force ini akan mencari semua kemungkinan yang ada, lalu melakukan evaluasi terhadap semua kemungkinan tersebut dan menyimpan solusi terbaik yang didapatkan, lalu solusi terbaik itu akan diumumkan.

## B. Pengertian Strategi Greedy

Algoritma Greedy merupakan algoritma yang membuat pilihan yang optimal secara lokal dengan harapan bahwa algoritma ini akan mengarah pada solusi yang optimal secara global. Jelas, algoritma greedy tidak selalu menghasilkan solusi optimal. Tapi untuk berbagai masalah penting algoritma greedy cukup kuat. Algoritma greedy selalu memecahkan masalah optimasi untuk setiap fungsi tujuan linear. Namun, ada situasi di mana algoritma greedy bekerja untuk banyak tetapi tidak semua fungsi tujuan linear.

Algoritma greedy hanya mencari suatu titik atau variabel permasalahan yang dimana memiliki nilai atau bobot paling besar. Algoritma greedy akan terus mencari dan menyimpan variabel dari nilai terbesar sampai nilai terkecil tanpa melanggar suatu constraint atau aturan yang sudah ditetapkan dalam suatu permasalahan atau kasus.

## C. Pengertian Permasalahan Policeman catch Thieves

Permasalahan Policeman catch Thieves atau polisi menangkap pencuri adalah sebuah permasalahan yang dimana terdapat 2 kelompok yaitu polisi dan pencuri dengan tujuan kelompok polisi harus menangkap banyaknya pencuri yang bisa ditangkap. Permasalahan tersebut memiliki beberapa aturan atau constraint yaitu:

1. 1 polisi hanya bisa menangkap 1 pencuri
2. Kelompok polisi tidak bisa menangkap semua pencuri jika nilai n pencuri lebih besar daripada kelompok polisi
3. Polisi hanya bisa menangkap pencuri dengan jarak k yang ditentukan

# BAB III

# IMPLEMENTASI

## A. Penerapan Permasalahan Policeman Catch Thieves dengan Strategi Brute Force

Penerapan dengan strategi brute force untuk permasalahan policeman catch thieves berikut melibatkan program untuk mencari seluruh kemungkinan yang tersedia bagi polisi agar dapat menangkap pencuri. Polisi dapat menangkap pencuri jika jarak dari polisi dan pencuri tidak melebihi nilai k yang ditentukan, berikut adalah contoh penerapan brute force.

Array = [‘P’, ‘T’, ‘P’, ‘P’, ‘T’, ‘T’]
K = 2

Solusi:

1) [‘P’, ‘T’], [‘P’, ‘T’], [‘P’, ‘T’]
2) [‘T’, ‘P’], [‘P’, ‘T’]
3) [‘P’, ‘T’], [‘P’, ‘T’]

Solusi brute force selalu menghasilkan solusi yang benar tetapi tidak dapat menghasilkan solusi yang optimal

## B. Penerapan Permasalahan Policeman Catch Thieves dengan Strategi Greedy

Penerapan dengan strategi greedy untuk permasalahan policeman catch thieves berikut melibatkan program untuk mencari seluruh kemungkinan dengan suatu ketentuan tertentu. Polisi dari paling kiri menangkap pencuri yang berada paling dekat dengan ketentuan index k masih diterima, konsep tersebut dapat dikatakan cukup efektif untuk beberapa kondisi saja, dan tidak semua kondisi dapat dipecahkan menggunakan konsep tersebut, berikut adalah contoh penerapan greedy.

Algoritma strategi greedy:

1) Temukan pasangan polisi dan pencuri yang berada di paling kiri array
2) Kasus 1: Jika jarak polisi dan pencuri <= index k yang diberikan, maka pencuri dapat ditangkap
3) Kasus 2: Jika jarak polisi dan pencuri > index k yang diberikan, maka tidak dapat ditangkap polisi yang sedang dipilih
    1. Jika polisi terletak di belakang pencuri, maka perlu dicari polisi berikutnya dan dicek apakah bisa menangkap pencuri tersebut atau tidak

2. Jika pencuri terletak di belakang polisi, maka perlu dicari pencuri berikutnya dan dicek apakah bisa ditangkap polisi tersebut atau tidak

4) Proses tersebut diulang kembali sampai program mencapai di titik 2 yang dimana berada pada kasus 1

Array = [‘P’, ‘T’, ‘P’, ‘P’, ‘T’, ‘T’]
K = 2

Konsep 1:

“Polisi paling kiri menangkap pencuri yang berada paling dekat”

Solusi:

1) [‘P’, ‘T’], [‘P’, ‘T’]

Konsep 2:

“Polisi paling kiri menangkap pencuri yang berada paling jauh”

Solusi:

1) [‘P’, ‘T’], [‘P’, ‘T’], [‘P’, ‘T’]

Strategi greedy tidak selalu menghasilkan solusi yang benar tetapi dapat menghasilkan solusi yang optimal.

## C. Perbandingan Kedua Strategi Penerapan Permasalahan Policeman Catch Thieves

Perbandingan menggunakan kedua strategi berikut dapat terlihat jelas tanpa memandang waktu kompleksitas bahwa strategi brute force selalu menghasilkan hasil yang benar karena strategi brute force selalu mengecek setiap kemungkinan jawaban atau solusi yang dihasilkan. Sedangkan strategi greedy tentu lebih optimal karena strategi greedy hanya mencari solusi berdasarkan suatu konsep atau aturan yang diterapkan.

# BAB IV

# ANALISIS

## A. Kompleksitas Waktu Permasalahan dengan Strategi Brute Force

Kompleksitas Waktu:

$$T(n) = \sum_{i=0}^{n-1} 1 + \sum_{\substack{l=0 \\ r=0}}^{n-1} 1 = n + n = O(n)$$

Ruang Bantu:
O(n)

Kompleksitas Ruang:
O(1)

## B. Kompleksitas Waktu Permasalahan dengan Strategi Greedy

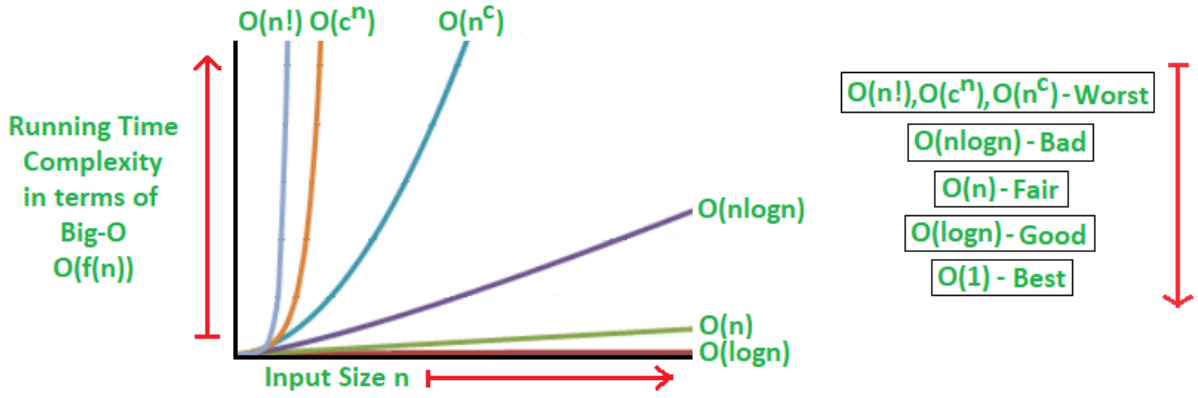
Kompleksitas Waktu:

$$T(n) = \sum_{i=0}^{n} 1 + \sum_{i=0}^{n} 1 + \sum_{\substack{pol=-1 \\ thi=-1}}^{n-1} 1 + \sum_{pol=-1}^{n-1} 1 + \sum_{thi=0}^{n-1} 1 = n + n + (n-2) + (n-1) + (n-1) = O(n)$$

Ruang Bantu:
O(1)

## C. Perbandingan Kompleksitas Waktu Menggunakan Kedua Strategi

Kompleksitas waktu strategi Brute Force dengan strategi Greedy dapat dibilang cukup sama dari segi waktu saja tetapi dari ruang bantu dapat dikatakan bahwa strategi greedy lebih unggul yang dimana O(1) lebih cepat dibandingkan dengan O(n), dapat dilihat pada gambar berikut mengenai penjelasan time complexity.




https://www.geeksforgeeks.org/analysis-algorithms-big-o-analysis/

# BAB V

# PENUTUP

## A. Kesimpulan

Kesimpulan yang dapat kami berikan adalah strategi yang cocok dari segi optimal untuk menyelesaikan permasalahan policeman catch thieves adalah strategi greedy yang dimana membuat program berjalan lebih cepat dibandingkan dengan strategi brute force. Strategi brute force dapat dikatakan sebagai strategi yang selalu menghasilkan output yang benar dikarenakan seluruh kemungkinan solusi akan dicari oleh program, tetapi tentu saja jika program mencari kemungkinan solusi, maka akan memakan waktu yang lebih lama dibandingkan dengan program yang hanya menggunakan 1 aturan atau konsep dalam mencari solusi. Jika kedua strategi dianalisis berdasarkan waktu kompleksitas, maka dapat dilihat memiliki waktu kompleksitas yang sama, maka dari itu penilaian suatu strategi juga dapat dilihat dari kompleksitas ruang yang dihasilkan yang dimana strategi greedy menang.

## B. Saran

Kami memahami bahwa program dan laporan yang kami buat masih sangat jauh dari kata sempurna. Kami sangat berterima kasih kepada ibu Dr. EMA RACHMAWATI, S.T., M.T. yang sudah membimbing dan mengajari kami dengan sangat sabar dengan tujuan supaya kami dapat memahami dan belajar mengenai beberapa strategi algoritma. Kami juga sangat berterima kasih karena sudah diberi kesempatan untuk mencoba belajar dan berkolaborasi ide kami dengan mengadakan tugas besar ini yang dimana kami tidak hanya belajar teori strategi algoritma tetapi juga belajar mengenai kerja sama kelompok atau tim yang dapat membantu kami menjadi pribadi yang lebih baik. Kami berharap bahwa laporan dan program yang kami buat dapat bermanfaat bagi siapapun yang membaca dan mencoba program kami.

# DAFTAR PUSTAKA

A. Vince. (2001). A Framework for the Greedy Algorithm. Diambil 29 Mei 2022, dari https://www.sciencedirect.com/science/article/pii/S0166218X01003626

Diambil 29 Mei 2022, dari http://bappeda.jogjaprov.go.id/dataku/data_dasar/index/447-jumlah-kasus-pencurian?id_skpd=39

SoumyadeepDebnath. (2022). Analysis of Algorithms | Big-O Analysis. Diambil 13 Juni 2022, dari https://www.geeksforgeeks.org/analysis-algorithms-big-o-analysis/

# LAMPIRAN

1. Program Policeman Catch Thieves dengan strategi Brute Force

```
"""_____________________________________________________________________
_
Permasalahan Policeman Catch Thieves Menggunakan Strategi Brute
Force|
Kelompok 8:
|
MUHAMMAD RAFI ANDEO PRAJA (1301200278)
|
M NAUFAL RIFQI RAMDHANI (1301201572)
|
RIZKY FERDIAN PRASETYO (1301204229)
|
_____________________________________________________________________
_|"""
# Program Python untuk menentukan jumlah maksimum pencuri yang dapat
ditangkap.

# Function untuk menginisialisasi pencuri dan polisi kedalam array
baru
def inisialisasi(arr, n):
    i = 0
    pencuri = []
    polisi = []

    # Menginisialisasikan polisi dan pencuri kedalam list
    while i < n:
        if arr[i] == 'P':
            polisi.append(i)
        elif arr[i] == 'T':
            pencuri.append(i)
        i += 1
    return polisi, pencuri

# Function untuk menentukan jumlah maksimum pencuri yang dapat
ditangkap
def tangkapPencuri(polisi, pencuri, k):
    # Minimum indeks untuk pencuri : pencuri[x], polisi: polisi[y]
```

```
    x = 0
    y = 0
    tertangkap = 0

    while x < len(pencuri) and y < len(polisi):
        # Jika pencuri dapat ditangkap
        if (abs( pencuri[x] - polisi[y] ) <= k):
            tertangkap += 1
            x += 1
            y += 1

        # Increment untuk minimum indeks
        elif pencuri[x] < polisi[y]:
            x += 1
        else:
            y += 1

    return tertangkap

# Main Program
if __name__=='__main__':
    arr1 = ['P', 'T', 'T', 'P', 'T']
    k = 2
    n = len(arr1)
    polisi, pencuri = inisialisasi(arr1, n)
    print(("Maksimum pencuri yang dapat ditangkap adalah: {}".
         format(tangkapPencuri(polisi, pencuri, k))))
```

2. Program Policeman Catch Thieves dengan strategi Greedy

```
"""_____________________________________________________________
Permasalahan Policeman Catch Thieves Menggunakan Strategi Greedy|
Kelompok 8:                                                     |
MUHAMMAD RAFI ANDEO PRAJA (1301200278)                          |
M NAUFAL RIFQI RAMDHANI (1301201572)                            |
RIZKY FERDIAN PRASETYO (1301204229)                             |
                                                                |"""
_________________________________________________________________

# Function untuk menentukan index awal dari polisi
def polisipertama(arr, n, polisi):
    for i in range(n):
```

```
        if (arr[i] == 'P'):
            polisi = i
            return polisi

# Function untuk menentukan index awal dari pencuri
def pencuripertama(arr, n, pencuri):
    for i in range(n):
        if (arr[i] == 'T'):
            pencuri = i
            return pencuri

# Function untuk menentukan jumlah maksimum pencuri yang dapat
ditangkap
def tangkapPencuri(arr, n, k):
    index_polisi = -1
    index_pencuri = -1
    tertangkap = 0
    polisi = polisipertama(arr, n, index_polisi)
    pencuri = pencuripertama(arr, n, index_pencuri)
    if (pencuri == -1 or polisi == -1):
        return 0
    while (polisi < n and pencuri < n):
        if (abs(polisi - pencuri) <= k):

            polisi += 1
            while (polisi < n and arr[polisi] != 'P'):
                polisi += 1

            pencuri += 1
            while (pencuri < n and arr[pencuri] != 'T'):
                pencuri += 1

            tertangkap += 1
        elif (pencuri < polisi):
            pencuri += 1
            while (pencuri < n and arr[pencuri] != 'T'):
                pencuri += 1
        else:
            polisi += 1
            while (polisi < n and arr[polisi] != 'P'):
```

```
                polisi += 1
    return tertangkap

# Main program
if __name__=='__main__':
    arr1 = ['P', 'T', 'P', 'P', 'T', 'T']
    k = 2
    n = len(arr1)
    print("Maksimum pencuri yang dapat ditangkap adalah: {}".
format(tangkapPencuri(arr1, n, k)))
```

3. Link Program Github
   https://github.com/RafiAndeo/tugas_besar_strategi_algoritma