**LAPORAN AKHIR PRAKTIKUM**

Mata Praktikum : PERANCANGAN DAN ANALISIS ALGORITMA

Kelas : 3IA11

Praktikum ke- : 3

Tanggal : 7/1/2025

Materi : BRUTEFORCE

NPM : 51422161

Nama : MUHAMMAD TARMIDZI BARIQ

Ketua Asisten : MURAD

Jumlah Lembar : 5



**LABORATORIUM TEKNIK INFORMATIKA**

**UNIVERSITAS GUNADARMA**

**2025**

1. Jelaskan apa keuntungan dan kekurangan dari Bruteforce.

2. Buatlah program dengan pengimplementasian Bruteforce.

3. Jelaskan program yang anda telah buat baris per baris.

Jawab

1.

**Keuntungan**

Kesederhanaan: Implementasinya sangat sederhana. Tidak memerlukan pengetahuan mendalam tentang algoritma atau teknik canggih.

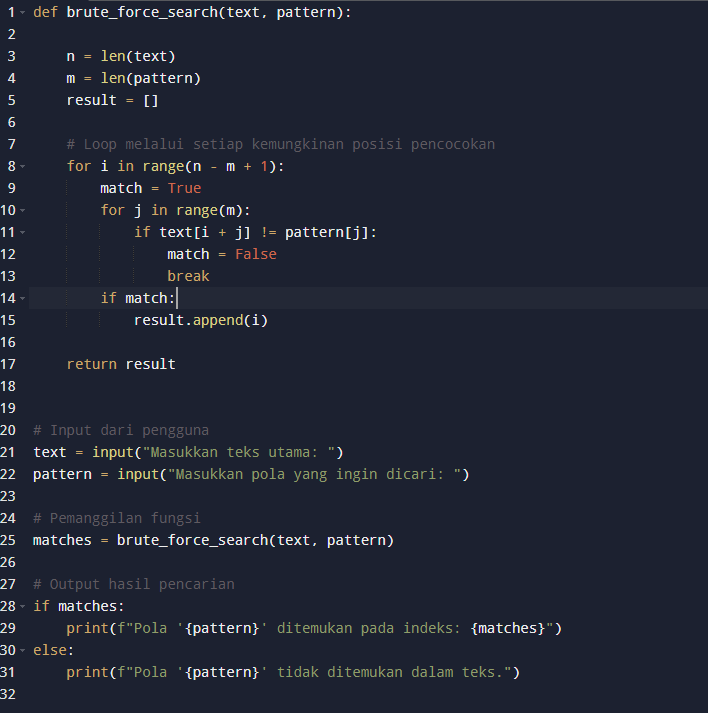
Ketepatan: Brute Force selalu menemukan solusi jika ada. Ini karena semua kemungkinan kombinasi dicoba.

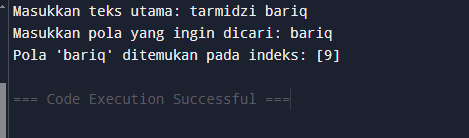
**Kekurangan**

Kecepatan: Proses ini cenderung sangat lambat ketika dihadapkan pada masalah yang kompleks dengan berbagai kemungkinan. Waktu yang dibutuhkan untuk eksekusi meningkat secara eksponensial seiring dengan bertambahnya jumlah input.

Efisiensi: Menggunakan sumber daya komputasi dalam jumlah yang besar. Hal ini dapat menjadi sangat boros energi dan tidak efisien, terutama ketika menangani masalah yang berskala besar.

2.





3

n = len(text)

m = len(pattern)

result = []

* n: Panjang teks utama.
* m: Panjang substring (pola) yang dicari.
* result: meyimpan array

for i in range(n - m + 1):

* Iterasi dari indeks pertama hingga indeks terakhir, n - m + 1: Batas iterasi, karena pencocokan pola tidak mungkin dimulai dari posisi yang terlalu dekat dengan akhir teks.

match = True

for j in range(m):

if text[i + j] != pattern[j]:

match = False

break

* i + j: Mengakses karakter teks yang sedang dicocokkan dengan karakter pada pola.
* Jika ada karakter yang tidak cocok (text[i + j] != pattern[j]), maka match diatur menjadi False dan pencocokan dihentikan untuk posisi ini (break).

if match:

result.append(i)

* Jika pola cocok dengan substring teks di posisi i, tambahkan indeks i ke dalam daftar result.

return result

* fungsi mengembalikan daftar result yang berisi semua indeks awal kemunculan pola.

text = input("Masukkan teks utama: ")

pattern = input("Masukkan pola yang ingin dicari: ")

* text: Input teks utama tempat pola akan dicari.
* pattern: Input substring atau pola yang akan dicocokkan.

matches = brute\_force\_search(text, pattern)

if matches:

print(f"Pola '{pattern}' ditemukan pada indeks: {matches}")

else:

print(f"Pola '{pattern}' tidak ditemukan dalam teks.")

* Jika matches tidak kosong, berarti pola ditemukan di beberapa indeks, dan indeks-indeks tersebut ditampilkan.
* Jika matches kosong, berarti pola tidak ditemukan dalam teks.