**LAPORAN TUGAS KECIL I**

**PENYELESAIAN *WORD SEARCH PUZZLE*DENGANALGORITMA *BRUTE FORCE***

Laporan dibuat untuk memenuhi salah satu tugas mata kuliah

IF2211 Strategi Algoritma



Disusun oleh:

**Kristo Abdi Wiguna 13520058**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2022**

## DAFTAR ISI

[**DAFTAR ISI**](#_7slfdo6f9jtp)1

[**Algoritma Brute Force**](#_4xts2hm1ww7x)1

[**Source Program**](#_xvo6myk30nxo)1

[**Screenshot Input dan Output**](#_fzbbr2qzzegr)1

[**Link Drive Kode Program**](#_yg06o7lkq2kb)1

[**Checklist**](#_cvoiy6o0l2s4)1

[**Daftar Referensi**](#_qvnsja9a8h9s)1

## Algoritma Brute Force

Algoritma penyelesaian word puzzle pertama dilakukan dengan membaca file yang berisi puzzle dan daftar jawaban yang perlu dicari. Pada saat membaca file teks tersebut, program akan membaca puzzle menjadi sebuah *vector of vector of char* dan membaca daftar jawaban menjadi *vector of string*. Lalu, terdapat 3 nested loop untuk mengiterasi karakter satu per satu dalam matriks huruf puzzle tersebut dan dicocokkan dengan urutan daftar jawaban. Ketika mengiterasi sebuah karakter, akan dicek setiap arah dengan menggunakan perulangan perhitungan delta dr dan dc dengan alur sebagai berikut.

* Ketika dr bernilai -1 dan dc bernilai -1, maka akan memanggil prosedur check untuk mengecek arah diagonal kiri bawah dari sebuah karakter,
* Ketika dr bernilai -1 dan dc bernila 0, maka akan memanggil prosedur check untuk mengecek arah horizontal kiri dari sebuah karakter,
* Ketika dr bernilai 0 dan dc bernilai 1, maka akan memanggil prosedur check untuk mengecek arah vertikal atas dari sebuah karakter, dan seterusnya.

Prosedur check akan mengecek kesamaan karakter puzzle dengan karakter jawaban ke-i dengan menggunakan perulangan. Jika out of bounds, maka perulangan akan terhenti dan akan kembali pada program utama untuk lanjut mengecek karakter selanjutnya. Jika tidak out of bounds dan sama, maka akan lanjut mengecek karakter selanjutnya sesuai arah yang sudah ditentukan oleh variabel dr dan dc serta mengubah karakter yang benar sementara menjadi ‘1’ untuk penanda.

Jika benar semua akan lanjut pada perulangan dimana karakter yang bukan ‘1’ akan diubah menjadi ‘-‘ dan mengubah ‘1’ menggunakan karakter asli pada matriks huruf salinan dari matriks awal. Setelah itu, matriks yang hanya menampilkan huruf di puzzle yang sesuai jawaban akan ditampilkan di output.

## Source Program

## Screenshot Input dan Output

Test 1. (Kecil 6x6)

Test 2. (Kecil 6x6)

Test 3. (Kecil 6x6)

Test 4. (Sedang 6x6)

Test 5. (Sedang 6x6)

Test 6. (Sedang 6x6)

Test 7. (Besar 6x6)

Test 8. (Besar 6x6)

Test 9. (Besar 6x6)

## Link Drive Kode Program

[*https://drive.google.com/drive/folders/1Z3QIil6msTq7ytK-AV4v5LDk-feMmBO\_?usp=sharing*](https://drive.google.com/drive/folders/1Z3QIil6msTq7ytK-AV4v5LDk-feMmBO_?usp=sharing)

## Checklist

|  |  |  |
| --- | --- | --- |
| Poin | Ya | Tidak |
| 1. Program berhasil dikompilasi tanpa kesalahan (no syntax error) | v |  |
| 2. Program berhasil running | v |  |
| 3. Program dapat membaca file masukan dan menuliskan luaran. | v |  |
| 4. Program berhasil menemukan semua kata di dalam puzzle | v |  |

## Daftar Referensi