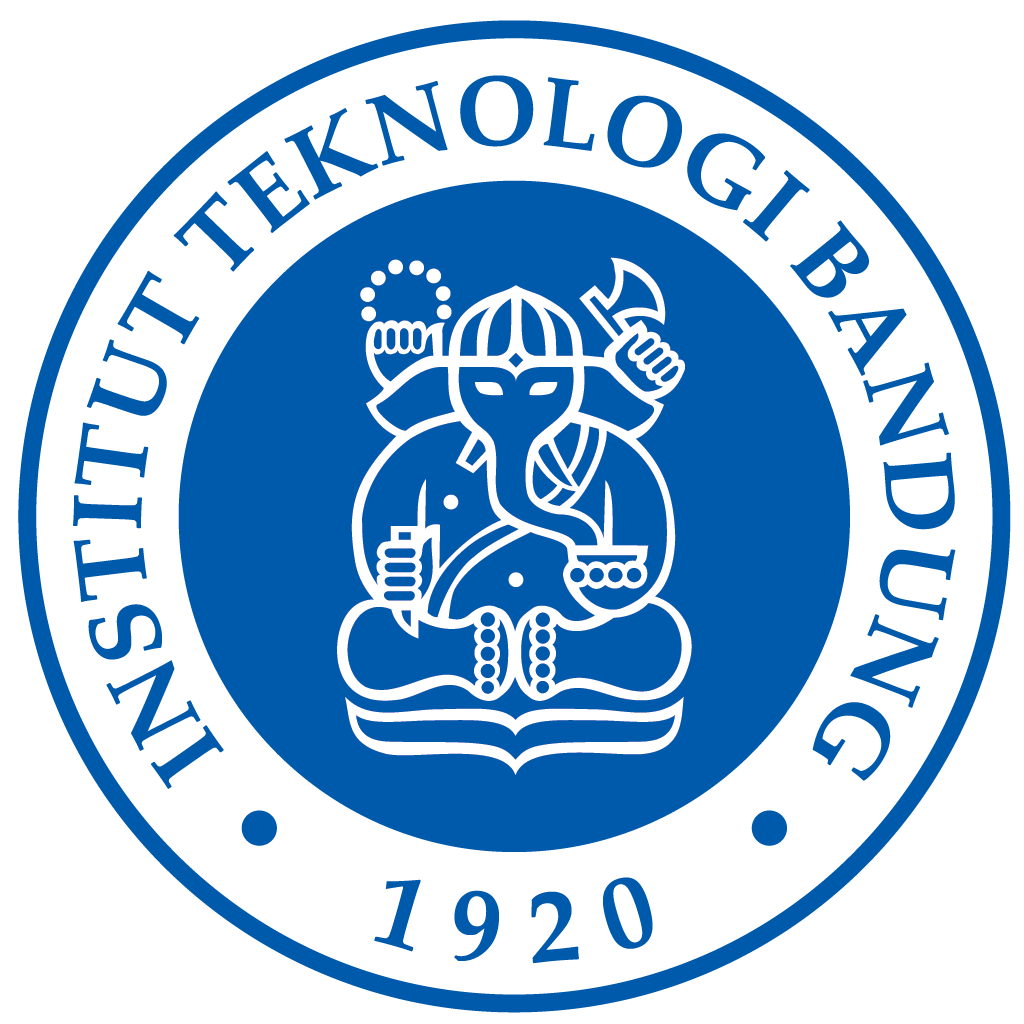
**TUGAS KECIL 1 IF2211 STRATEGI ALGORITMA: WORD SEARCH PUZZLE DENGAN ALGORITMA *BRUTE FORCE***



Dibuat oleh :

Rizky Akbar Asmaran (13520111)

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

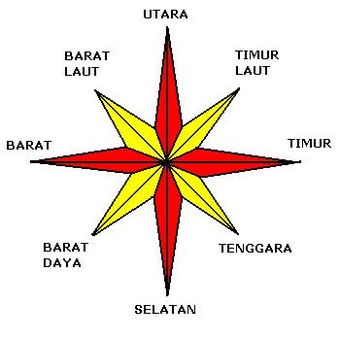
2022

1. PENJELASAN ALGORITMA BRUTE FORCE

Definisi brute force berdasarkan Cambridge Dictionary yaitu menyelesaikan sesuatu dengan kekuatan yang besar. Dalam algoritma brute force memiliki prinsip yang sama, penyelesaian masalah dengan tidak cerdas dan tidak mangkus (butuh jumlah komputasi yang besar dan waktu yang lama).

Pada Program penyelesaian *Word Search Puzzle* ini digunakan algoritma *Brute Force* menggunakan Bahasa pemograman C++. Adapun berikut merupakan Langkah-Langkah yang dilakukan dalam menyelesaikan program *Word Search Puzzle* dengan menggunakan Algoritma *Brute Force* :

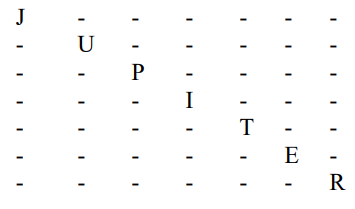
1. Membuat sebuah matriks karakter untuk menyimpan sebuah puzzle yang didapatkan dari input berupa file. Lalu membuat sebuah list kata yang berupa jawaban dari puzzle tersebut yang didapatkan dari input dari pengguna berupa file.
2. Lalu dilakukan pencarian kata pada list kata dengan mencocokan huruf pertama kata yang akan dicari dengan karakter pertama pada matriks.
3. Apabila karakter tersebut sama, proses pencocokan dilanjutkan ke arah yang telah ditentukan sebelumnya. Dalam proses pencarian dan pencocokan ini terdapat 8 arah yang bisa ditempuh seperti arah mata angin.



Gambar 1.1 Arah Mata Angin

Mulai dari arah utara (atas), arah timur laut, arah timur, arah tenggara, arah selatan, arah barat daya, arah barat hingga arah barat laut.

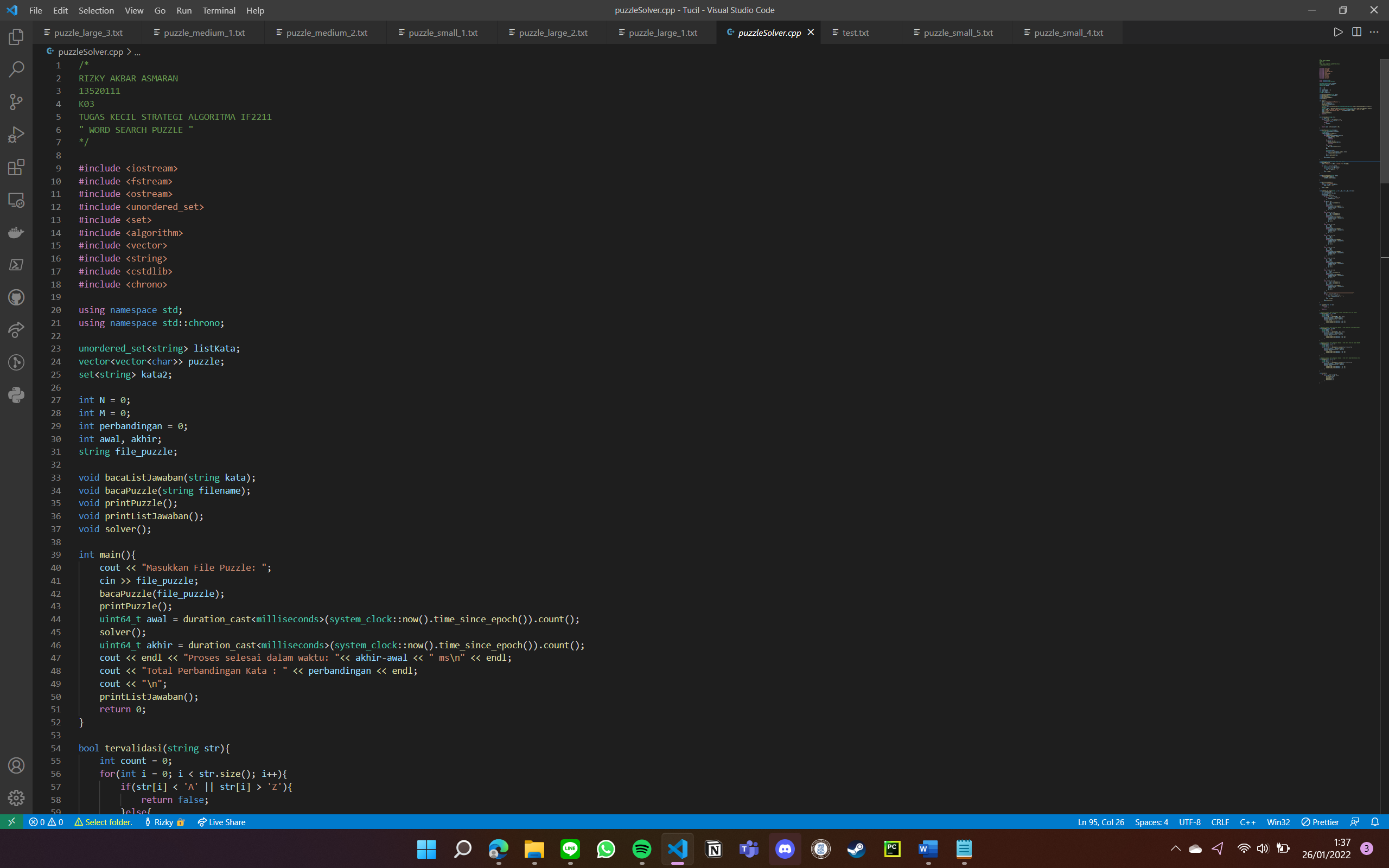
1. Dan sebaliknya, apabila karakter yang dijumpai tidak sama, akan terjadi pergantian arah dan melanjutkan pencarian Kembali, dan apabila semua arah telah di lewati maka akan dilanjutkan ke huruf selanjutnya pada puzzle bisa ke huruf kanan atau bawah dari kolom pertama.
2. Apabila ditemukan kata yang cocok sesuai dengan pencarian, maka karakter sudah cocok dengan kata yang dicari sehingga setelah itu program akan mengeluarkan luaran berupa matriks puzzle dengan bentuk puzzle dengan posisi kata yang dicari seperti berikut.

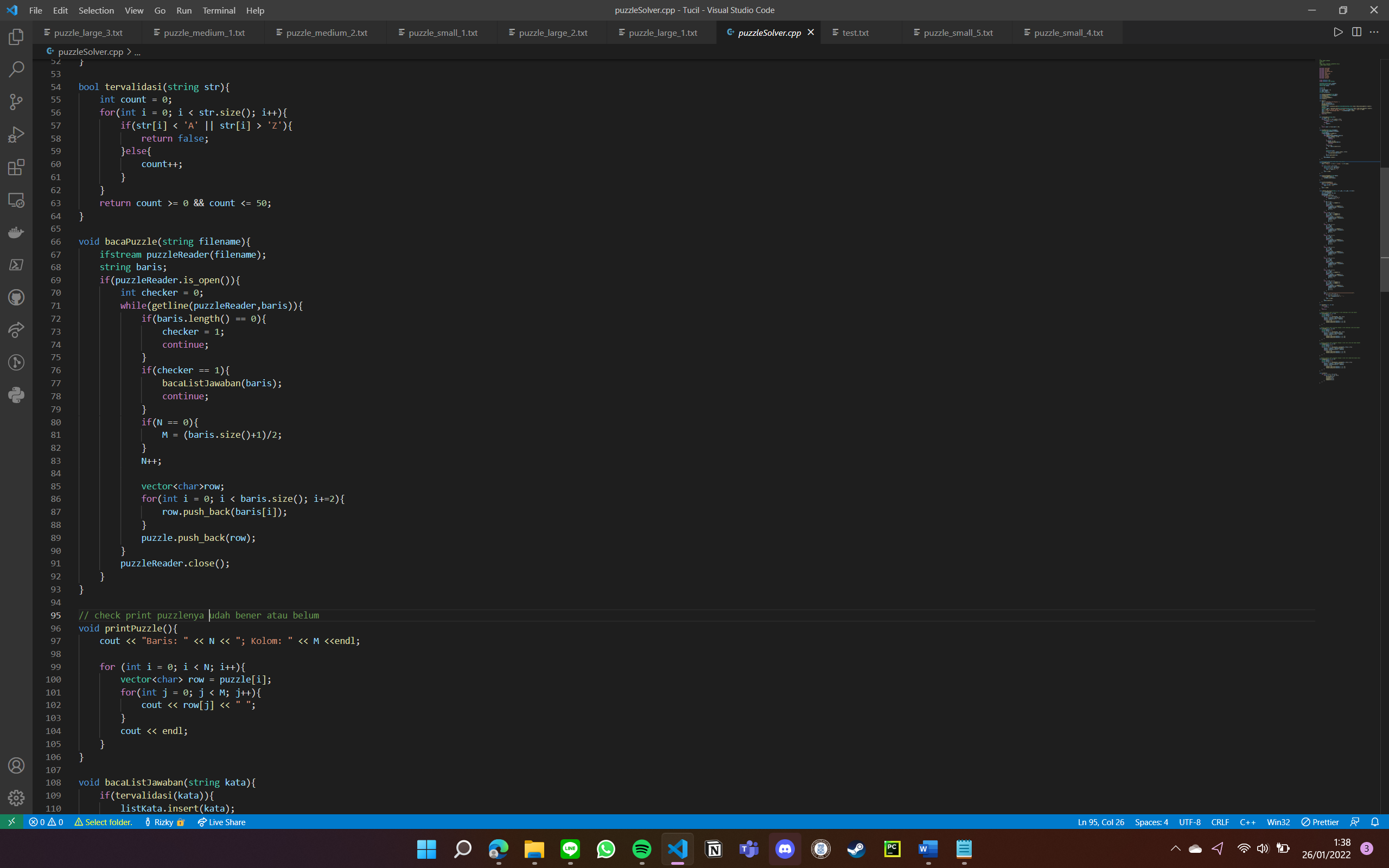


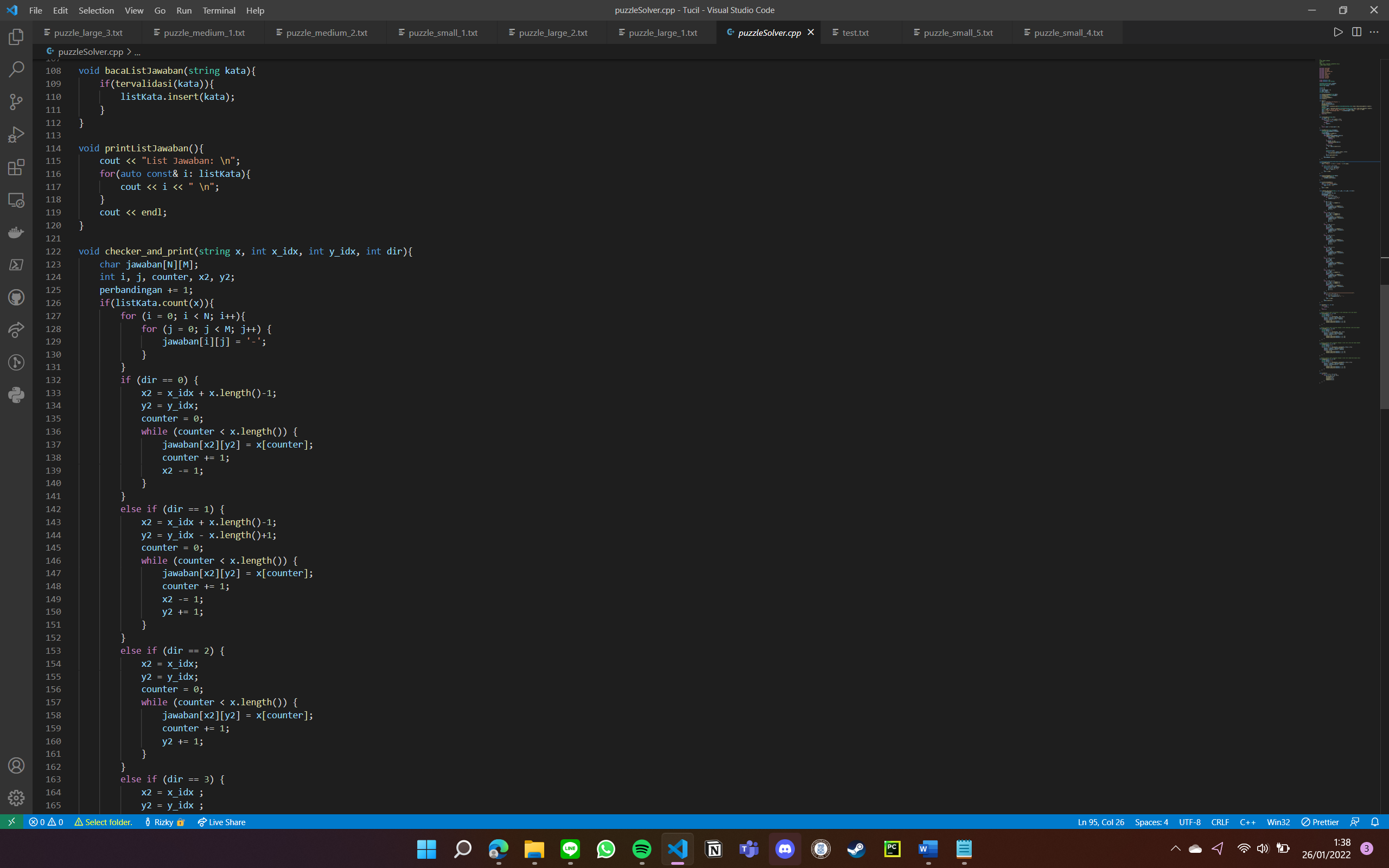
Gambar 1.2 Contoh Output

1. Apabila semua arah telah diperiksa, maka pencarian dilanjutkan ke kata berikutnya dalam daftar jawaban puzzle yang harus dicari, akan tetapi masih membandingkan dengan huruf pada puzzle.
2. Apabila seluruh huruf telah diperiksa, pencarian sudah selesai. Luaran akhir pada program ini akan berupa jumlah waktu yang dibutuhkan, total kata yang dibandingkan dan list kata yang ditemukan pada puzzle.
3. SOURCE CODE PROGRAM

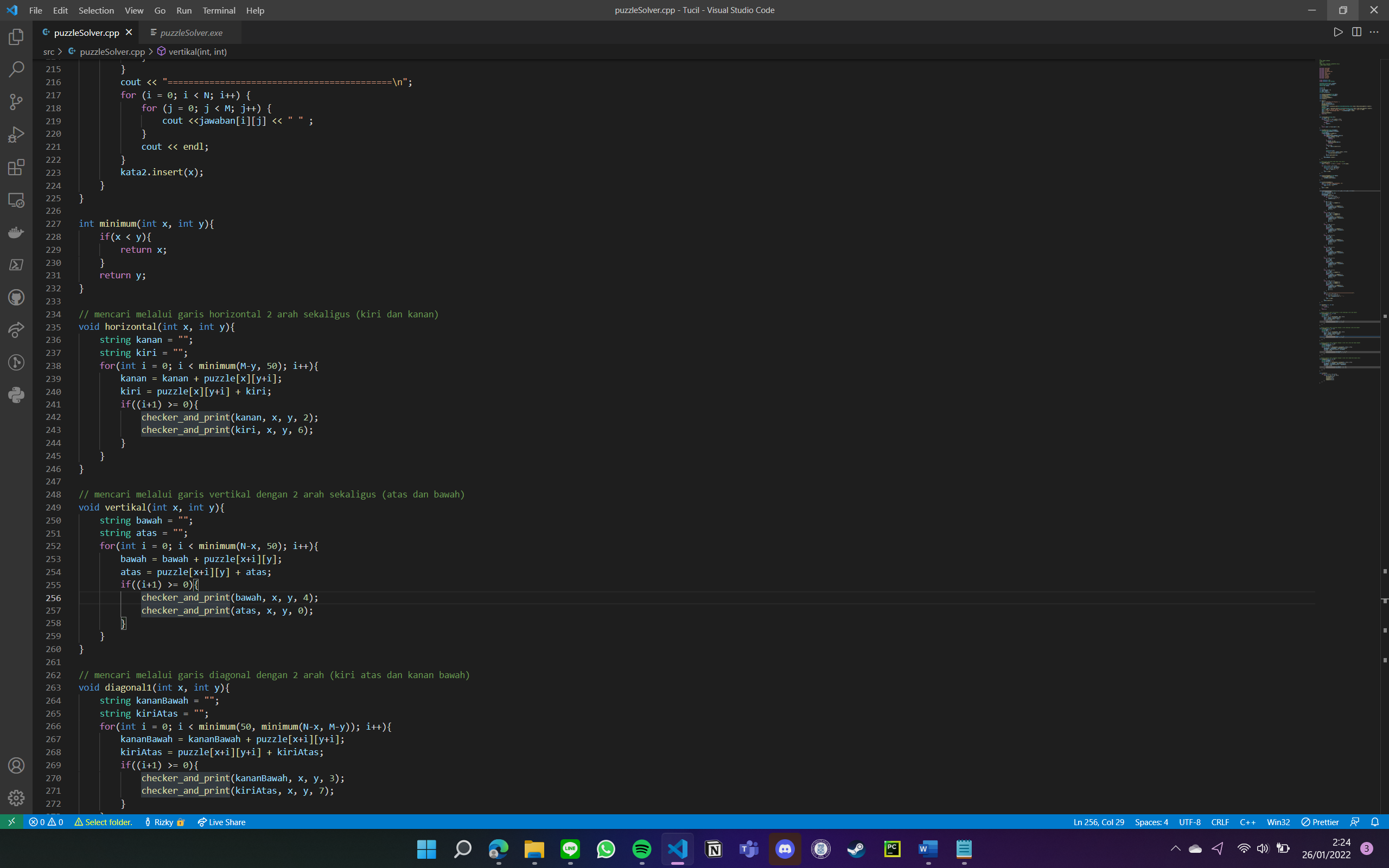
puzzleSolver.cpp

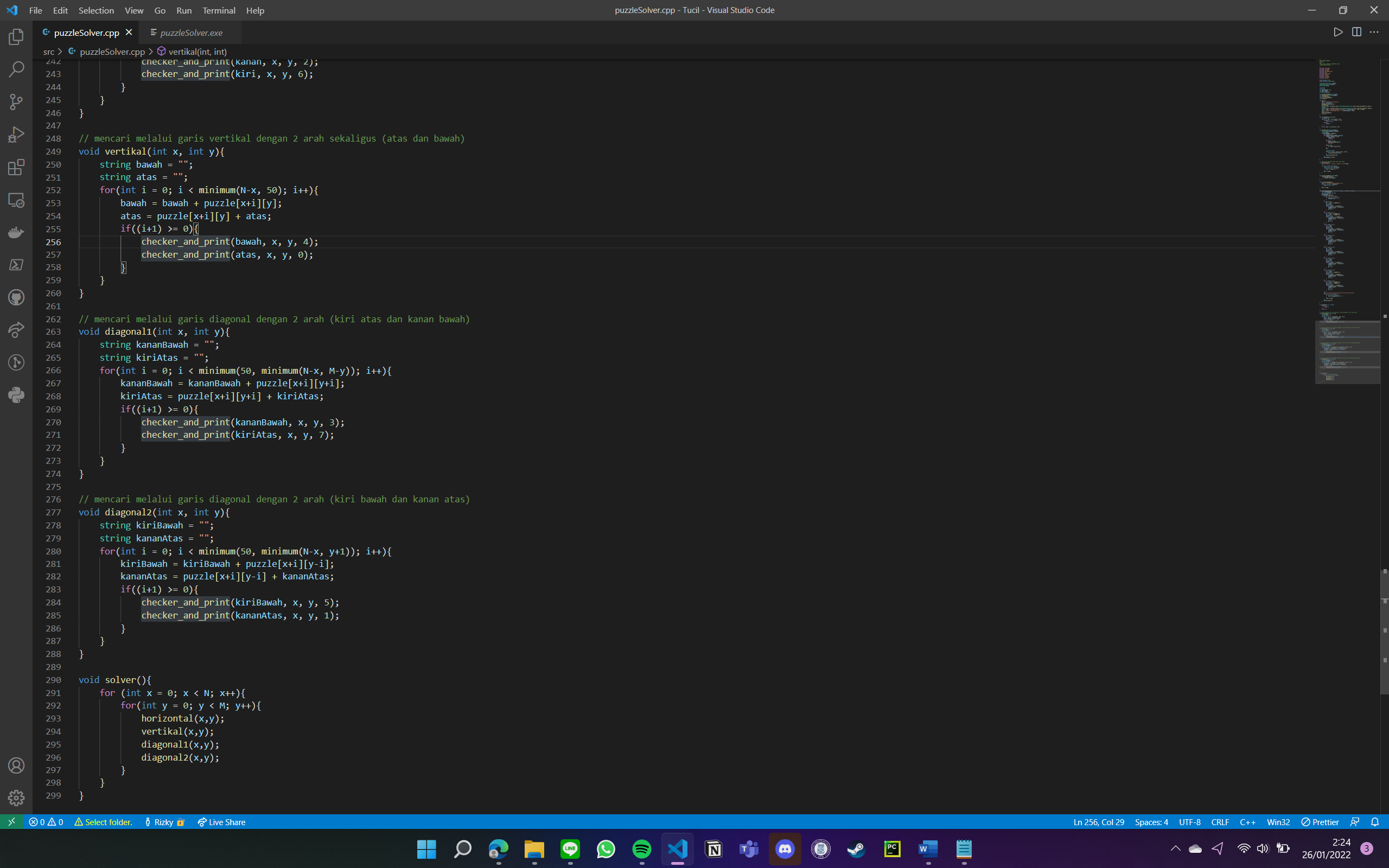












1. SCREENSHOT DARI INPUT DAN OUTPUT

|  |  |
| --- | --- |
| 1. Small Version | |
|  |  |
|  |  |
|  |  |
| 1. Medium Version | |
|  |  |
|  |  |
|  |  |
| 1. Large Version | |
|  | |
|  |  |
|  |  |
|  |  |
|  | |
|  | |
|  |  |
|  |  |
|  |  |
|  | |
|  |  |
|  |  |
|  | |

1. LINK SOURCE CODE

Link Github: <https://github.com/kibare/Tucil-Stima-1>

Or

Link Google Drive: https://drive.google.com/file/d/1IMG8Tx4iuyvjGOfY2CMjZx7hP0eGtbN9/view?usp=sharing

1. CHECKLIST

|  |  |  |
| --- | --- | --- |
| Poin | YA | TIDAK |
| 1. Program berhasil dikompilasi tanpa kesalahan (no syntax error) | √ |  |
| 1. Program berhasil *running* | √ |  |
| 1. Program dapat membaca berkas input dan menuliskan luaran | √ |  |
| 1. Program berhasil menemukan semua kata di dalam puzzle | √ |  |