

LAPORAN TUGAS KECIL 1

IF2211 Strategi Algoritma

Penyelesaian *Word Search Puzzle* dengan Algoritma *Brute Force*



Disusun oleh:

Nama : Johannes Winson Sukiatmodjo

NIM : 13520123

PROGRAM STUDI TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2022

Algoritma *Brute Force*

Berdasarkan program yang telah saya buat, langkah-langkah pencarian solusinya adalah sebagai berikut:

1. Program akan mengecek setiap kata sampai ketemu. Jika kata sudah ketemu ke arah tertentu, proses pencarian dengan arah lain tidak akan dilakukan karena looping akan langsung berhenti dan akan mengecek kata-kata berikutnya.
2. Program akan mengecek setiap kata dengan mencocokkannya pada setiap karakter pada *puzzle* dengan urutan arah yang akan dicek terlebih dahulu adalah vertikal ke atas, vertikal ke bawah, horizontal ke kanan, horizontal ke kiri, diagonal ke kanan atas, diagonal ke kanan bawah, diagonal ke kiri atas, dan diagonal ke kiri bawah.
3. Sebelum masuk ke dalam proses pencocokan karakter, program akan mengecek terlebih dahulu apakah panjang kata tersebut melebihi batas *puzzle*. Jika panjang kata tersebut melebihi batas *puzzle*, pencocokan karakter melalui arah tersebut tidak akan dilakukan dan dilanjutkan dengan pencocokan ke arah yang lain.

Source Program

```
#include <iostream>
#include <fstream>
#include <chrono>

using namespace std;

int main() {
    string txt;
    int baris, kolom, keywords, check, i, j, k, l, m, a, b, c;
    int checkall = 0;
    bool found;
    cout << "Masukkan nama file : ";
    cin >> txt;
    cout << "Masukkan jumlah baris : ";
    cin >> baris;
    cout << "Masukkan jumlah kolom : ";
    cin >> kolom;
    cout << "Masukkan jumlah kata kunci : ";
    cin >> keywords;
    char character[baris][kolom];
    string word[keywords];
    string line;
    ifstream file;
    file.open("../test/" + txt);
    for (i = 0; i < baris; i++) {
        for (j = 0; j < kolom; j++) {
            file >> character[i][j];
        }
    }
    int idx = 0;
    while (getline(file, line)) {
        file >> word[idx];
        idx++;
    }
    file.close();
    auto started = std::chrono::high_resolution_clock::now();
    cout <<
    "=====
    ==> << endl;
    for (i = 0; i < keywords; i++) {
        check = 0;
        found = false;
        for (j = 0; j < baris; j++) {
            for (k = 0; k < kolom; k++) {
```

```

a = j;
b = k;
c = 0;
if (a - word[i].length() + 1 >= 0) { // vertikal ke atas
    while (character[a][b] == word[i][c]) {
        a--;
        c++;
        check++;
        checkall++;
        if (word[i].length() == c) {
            found = true;
            a = j - word[i].length() + 1;
            b = k;
            c = 0;
            for (l = 0; l < baris; l++) {
                for (m = 0; m < kolom; m++) {
                    if ((l == a) && (m == b) && (c <
word[i].length())) {
                        cout << character[l][m] << ' ';
                        a++;
                        c++;
                    } else {
                        cout << '-' << ' ';
                    }
                }
                cout << endl;
            }
            break;
        }
    }
    if (found) {
        break;
    }
    check += 1;
    checkall += 1;
    a = j;
    b = k;
    c = 0;
}
if (a + word[i].length() - 1 < baris) { // vertikal ke bawah
    while (character[a][b] == word[i][c]) {
        a++;
        c++;
        check++;
        checkall++;
        if (word[i].length() == c) {
            found = true;
            a = j;

```

```

        b = k;
        c = 0;
        for (l = 0; l < baris; l++) {
            for (m = 0; m < kolom; m++) {
                if ((l == a) && (m == b) && (c <
word[i].length())) {
                    cout << character[l][m] << ' ';
                    a++;
                    c++;
                } else {
                    cout << '-' << ' ';
                }
            }
            cout << endl;
        }
        break;
    }
}
if (found) {
    break;
}
check += 1;
checkall += 1;
a = j;
b = k;
c = 0;
}
if (b + word[i].length() - 1 < kolom) { // horizontal ke kanan
    while (character[a][b] == word[i][c]) {
        b++;
        c++;
        check++;
        checkall++;
        if (word[i].length() == c) {
            found = true;
            a = j;
            b = k;
            c = 0;
            for (l = 0; l < baris; l++) {
                for (m = 0; m < kolom; m++) {
                    if ((l == a) && (m == b) && (c <
word[i].length())) {
                        cout << character[l][m] << ' ';
                        b++;
                        c++;
                    } else {
                        cout << '-' << ' ';
                    }
                }
            }
        }
    }
}

```

```

        }
        cout << endl;
    }
    break;
}
}
if (found) {
    break;
}
check += 1;
checkall += 1;
a = j;
b = k;
c = 0;
}
if (b - word[i].length() + 1 >= 0) { // horizontal ke kiri
    while (character[a][b] == word[i][c]) {
        b--;
        c++;
        check++;
        checkall++;
        if (word[i].length() == c) {
            found = true;
            a = j;
            b = k - word[i].length() + 1;
            c = 0;
            for (l = 0; l < baris; l++) {
                for (m = 0; m < kolom; m++) {
                    if ((l == a) && (m == b) && (c <
word[i].length())) {
                        cout << character[l][m] << ' ';
                        b++;
                        c++;
                    } else {
                        cout << '-' << ' ';
                    }
                }
            }
            cout << endl;
        }
        break;
    }
}
if (found) {
    break;
}
check += 1;
checkall += 1;
a = j;

```

```

        b = k;
        c = 0;
    }
    if ((a - word[i].length() + 1 >= 0) && (b + word[i].length() -
1 < kolom)) { // diagonal ke kanan atas
        while (character[a][b] == word[i][c]) {
            a--;
            b++;
            c++;
            check++;
            checkall++;
            if (word[i].length() == c) {
                found = true;
                a = j - word[i].length() + 1;
                b = k + word[i].length() - 1;
                c = 0;
                for (l = 0; l < baris; l++) {
                    for (m = 0; m < kolom; m++) {
                        if ((l == a) && (m == b) && (c <
word[i].length())) {
                            cout << character[l][m] << ' ';
                            a++;
                            b--;
                            c++;
                        } else {
                            cout << '-' << ' ';
                        }
                    }
                }
                cout << endl;
            }
            break;
        }
    }
    if (found) {
        break;
    }
    check += 1;
    checkall += 1;
    a = j;
    b = k;
    c = 0;
}
if ((a + word[i].length() - 1 < baris) && (b +
word[i].length() - 1 < kolom)) { // diagonal ke kanan bawah
    while (character[a][b] == word[i][c]) {
        a++;
        b++;
        c++;
    }
}

```

```

        check++;
        checkall++;
        if (word[i].length() == c) {
            found = true;
            a = j;
            b = k;
            c = 0;
            for (l = 0; l < baris; l++) {
                for (m = 0; m < kolom; m++) {
                    if ((l == a) && (m == b) && (c <
word[i].length())) {
                        cout << character[l][m] << ' ';
                        a++;
                        b++;
                        c++;
                    } else {
                        cout << '-' << ' ';
                    }
                }
                cout << endl;
            }
            break;
        }
    }
    if (found) {
        break;
    }
    check += 1;
    checkall += 1;
    a = j;
    b = k;
    c = 0;
}
if ((a - word[i].length() + 1 >= 0) && (b - word[i].length() +
1 >= 0)) { // diagonal ke kiri atas
    while (character[a][b] == word[i][c]) {
        a--;
        b--;
        c++;
        check++;
        checkall++;
        if (word[i].length() == c) {
            found = true;
            a = j - word[i].length() + 1;
            b = k - word[i].length() + 1;
            c = 0;
            for (l = 0; l < baris; l++) {
                for (m = 0; m < kolom; m++) {

```



```

                                if ((l == a) && (m == b) && (c <
word[i].length())) {
                                cout << character[l][m] << ' ';
                                a++;
                                b++;
                                c++;
                                } else {
                                    cout << '-' << ' ';
                                }
                            }
                        cout << endl;
                    }
                break;
            }
        }
    if (found) {
        break;
    }
    check += 1;
    checkall += 1;
    a = j;
    b = k;
    c = 0;
}
if ((a + word[i].length() - 1 < baris) && (b -
word[i].length() + 1 >= 0)) { // diagonal ke kiri bawah
    while (character[a][b] == word[i][c]) {
        a++;
        b--;
        c++;
        check++;
        checkall++;
        if (word[i].length() == c) {
            found = true;
            a = j;
            b = k;
            c = 0;
            for (l = 0; l < baris; l++) {
                for (m = 0; m < kolom; m++) {
                    if ((l == a) && (m == b) && (c <
word[i].length())) {
                        cout << character[l][m] << ' ';
                        a++;
                        b--;
                        c++;
                    } else {
                        cout << '-' << ' ';
                    }
                }
            }
        }
    }
}

```

```

        }
        cout << endl;
    }
    break;
}
}
if (found) {
    break;
}
check += 1;
checkall += 1;
a = j;
b = k;
c = 0;
}
}
if (found) {
    cout << "Total perbandingan huruf yang dilakukan : " << check
<< endl;

    cout <<
"=====
==" << endl;

    break;
}
}
}
auto done = std::chrono::high_resolution_clock::now();
std::cout << "Waktu eksekusi program : " <<
std::chrono::duration_cast<std::chrono::milliseconds>(done-started).count() <<
" ms" << endl;
    cout << "Jumlah total perbandingan huruf yang dilakukan : " << checkall <<
endl;
    return 0;
}

```

Screenshots

1. 14x12x16.txt

The image shows a Visual Studio Code editor window with a C++ file named 'main.cpp'. The code is a simple program that reads a file '14x12x16.txt' and compares its contents with the string 'ADA' and 'N'. The program counts the number of comparisons made for each string. The output of the program is displayed in the terminal window at the bottom of the editor. The output shows that the string 'ADA' was compared 747 times, and the string 'N' was compared 67 times. The terminal window also shows the command prompt and the execution of the program. The Visual Studio Code interface includes a sidebar on the left with icons for Explorer, Search, and Run and Debug. The top of the window has a menu bar with options like File, Edit, Selection, View, Go, Run, Terminal, and Help. The bottom of the window shows a status bar with information about the current file, line, column, and encoding.

[illegible]

2. 14x14x8.txt

The screenshot shows the Visual Studio Code interface with a file named 'main.cpp' open. The code implements a program to analyze the frequency of letters in a text file. It uses arrays to store counts for each letter (A-Z) and calculates the total number of occurrences for each letter. The output shows that the letter 'H' appears 1315 times and the letter 'U' appears 721 times. The interface includes a sidebar with Explorer, Search, and Run and Debug views, and a bottom status bar showing the current file, line, column, and encoding information.

[illegible]

3. 16x16x6.txt

The image shows a Windows desktop environment. The primary focus is a Visual Studio Code (VS Code) window titled 'mahapp - Tucil-1-Stima - Visual Studio Code'. The 'TERMINAL' pane is active, displaying the output of a PowerShell script. The script, located at 'C:\Users\johan\OneDrive\Documents\GitHub\Tucil-1-Stima\src\c', performs a letter frequency analysis on a file named 'Ho-Hoed.txt'. It reports a total of 654 comparisons for uppercase letters and 538 for lowercase letters. The script also lists the letters and their respective counts, such as 'R' with 10 counts and 'E' with 10 counts. The Windows taskbar at the bottom shows various application icons, including the Start menu, File Explorer, Google Chrome, WhatsApp, Telegram, Discord, and several other utility and communication apps. The system clock in the bottom right corner indicates the time is 4:31 on 25/01/2022.

The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal displays the output of a C++ program that compares letters from 'A' to 'Z' using nested loops. The output is as follows:

```
main.cpp - Tucil-1-Stima - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
-----
Total perbandingan huruf yang dilakukan : 649
-----
      G - - - - -
    - A - - - - -
      N - - - - -
    - E - - - - -
      T - - - - -
    - M - - - - -
      L - - - - -
    - A - - - - -
      I - - - - -
    - O - - - - -
      U - - - - -
    - D - - - - -
      P - - - - -
    - F - - - - -
      B - - - - -
    - H - - - - -
      K - - - - -
    - J - - - - -
      V - - - - -
    - W - - - - -
      X - - - - -
    - Y - - - - -
      Z - - - - -
-----
Total perbandingan huruf yang dilakukan : 81
-----
      M - - - - -
    - O - - - - -
      E - - - - -
    - A - - - - -
      I - - - - -
    - D - - - - -
      P - - - - -
    - F - - - - -
      B - - - - -
    - H - - - - -
      K - - - - -
    - J - - - - -
      V - - - - -
    - W - - - - -
      X - - - - -
    - Y - - - - -
      Z - - - - -
-----
Total perbandingan huruf yang dilakukan : 820
-----
Waktu eksekusi program : 135 ms
Jumlah total perbandingan huruf yang dilakukan : 3486
PS C:\Users\johan\OneDrive\Documents\GitHub\Tucil-1-Stima>src>
```

The status bar at the bottom indicates the file is named "main.cpp", the encoding is UTF-8, and the language is C++. The system clock shows 4:31 on 25/01/2022.

4. 20x18x6.txt

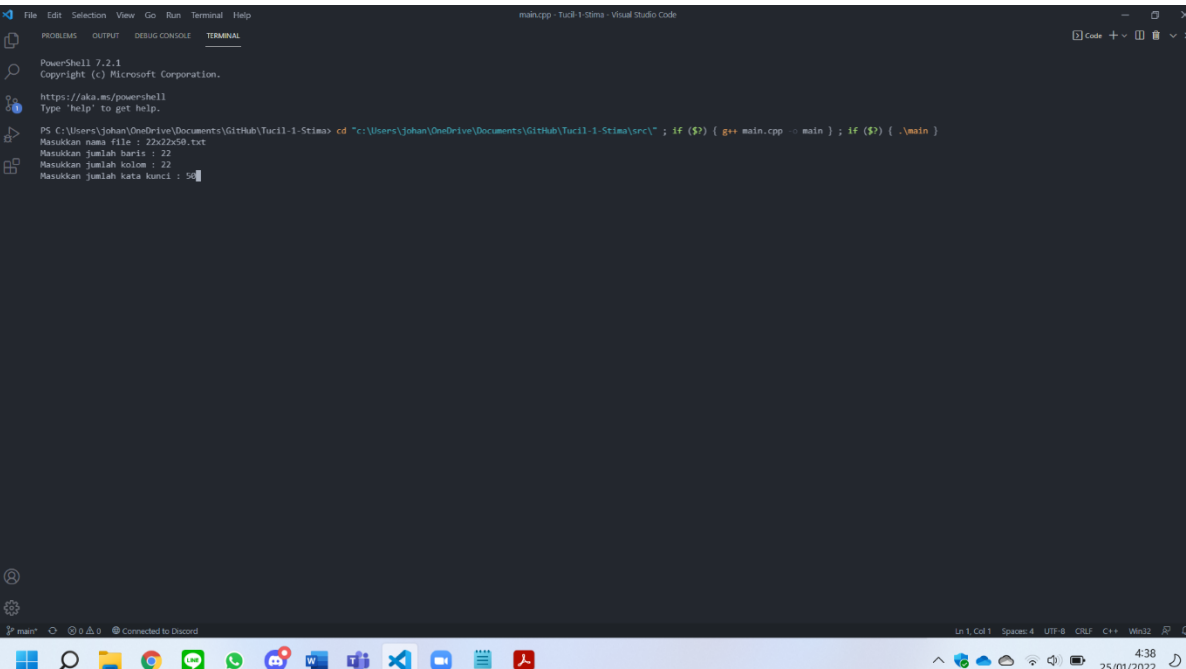
The image shows a Windows desktop environment. In the foreground, a Visual Studio Code window is open, displaying a PowerShell terminal. The terminal window has a title bar that reads "muhapp - Tucil-1-Stima - Visual Studio Code". The terminal output shows the following sequence of commands and results:

```
PowerShell 7.2.1  
Copyright (c) Microsoft Corporation.  
  
https://aka.ms/powershell  
Type 'help' to get help.  
  
PS C:\Users\johan\OneDrive\Documents\GitHub\Tucil-1-Stima> cd "c:\Users\johan\OneDrive\Documents\GitHub\Tucil-1-Stima\src\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }  
Masukkan nama file : 00-18bd.txt  
Masukkan jumlah baris : 20  
Masukkan jumlah kolom : 18  
Masukkan jumlah kata kunci : 6  
-----  
- A -  
- B -  
- E -  
- R -  
-----  
Total perbandingan huruf yang dilakukan : 708  
-----  
- )  
- E  
- A  
- R  
-----
```


At the bottom of the screen, the Windows taskbar is visible, showing the Start button and several pinned application icons including File Explorer, Google Chrome, WhatsApp, Telegram, Discord, and others. The system tray in the bottom right corner displays the time as 4:34 and the date as 25/01/2022.

The image shows a Visual Studio Code editor window with a C++ program named 'main.cpp' open. The program is a letter frequency analyzer. It starts by calculating the total number of letters in a string and then counts the frequency of each letter from 'A' to 'Z'. The output is displayed in the terminal window at the bottom. The first part of the program shows a total of 2246 letters and a frequency distribution for letters A through Z. The second part of the program shows a total of 2185 letters and a frequency distribution for letters A through Z. The program is running on a Windows operating system, as indicated by the taskbar at the bottom. The taskbar shows various application icons, including the Start menu, File Explorer, Google Chrome, and several communication apps like WhatsApp, Telegram, and Discord. The system clock in the bottom right corner shows the time as 4:34 on 25/01/2022.

5. 22x22x50.txt



The screenshot shows a Windows 10 desktop with a Visual Studio Code editor window. The editor is open to a file named 'mah.cpp' in a project named 'Tucil-1-Stima'. The terminal window is active, showing the output of a C++ program. The program prompts the user to enter the number of rows (22), the number of columns (22), and the number of words (50). The output shows the program has successfully processed the input.

Visual Studio Code Interface:

- Menu Bar: File, Edit, Selection, View, Go, Run, Terminal, Help
- Activity Bar (Left): Explorer, Search, Source Control, Run and Debug, Extensions
- Editor Area:
 - File Explorer: mah.cpp - Tucil-1-Stima - Visual Studio Code
 - Terminal:


```
PowerShell 7.2.1
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.

PS C:\Users\johan\OneDrive\Documents\GitHub\Tucil-1-Stima> cd "c:\Users\johan\OneDrive\Documents\GitHub\Tucil-1-Stima\src\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
Masukkan nama file : 22x22x50.txt
Masukkan jumlah baris : 22
Masukkan jumlah kolom : 22
Masukkan jumlah kata kunci : 50
```
- Status Bar (Bottom): main.cpp, 43 Lines, 4 Columns, UTF-8, C++, Win32, 4:38 PM, 25/01/2022

[illegible]

6. 24x24x50.txt

The screenshot shows a Visual Studio Code editor window with a C++ program. The program is titled 'main.cpp - Tucil 1-Stima - Visual Studio Code'. The code is a C++ program that compares two strings and counts the frequency of each letter (A-Z) in both. The program uses a 2D array to store the counts. The output of the program is displayed in the terminal, showing the total comparison count (2538) and the execution time (2553 ms). The status bar at the bottom indicates the file is 'main.cpp' and the editor is in 'main' mode.

```

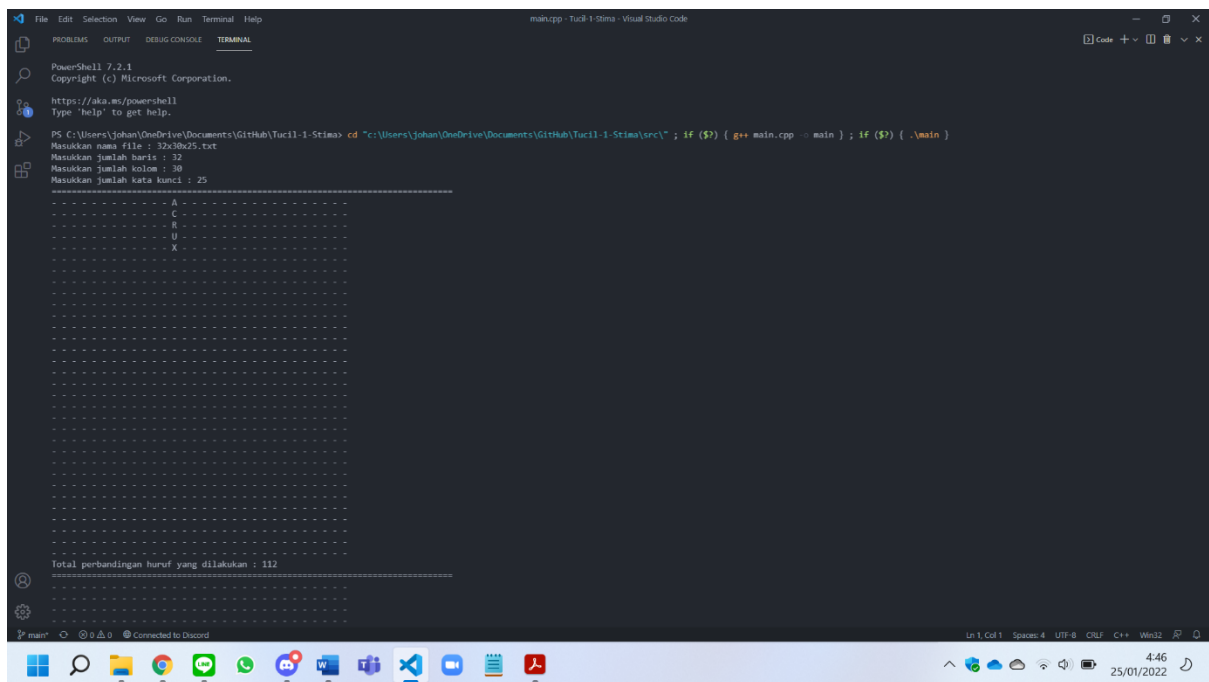
File Edit Selection View Go Run Terminal Help
main.cpp - Tucil 1-Stima - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
-----
- A -
- B -
- C -
- D -
- E -
- F -
- G -
- H -
- I -
- J -
- K -
- L -
- M -
- N -
- O -
- P -
- Q -
- R -
- S -
- T -
- U -
- V -
- W -
- X -
- Y -
- Z -

Total perbandingan huruf yang dilakukan : 2538
-----
- A -
- B -
- C -
- D -
- E -
- F -
- G -
- H -
- I -
- J -
- K -
- L -
- M -
- N -
- O -
- P -
- Q -
- R -
- S -
- T -
- U -
- V -
- W -
- X -
- Y -
- Z -

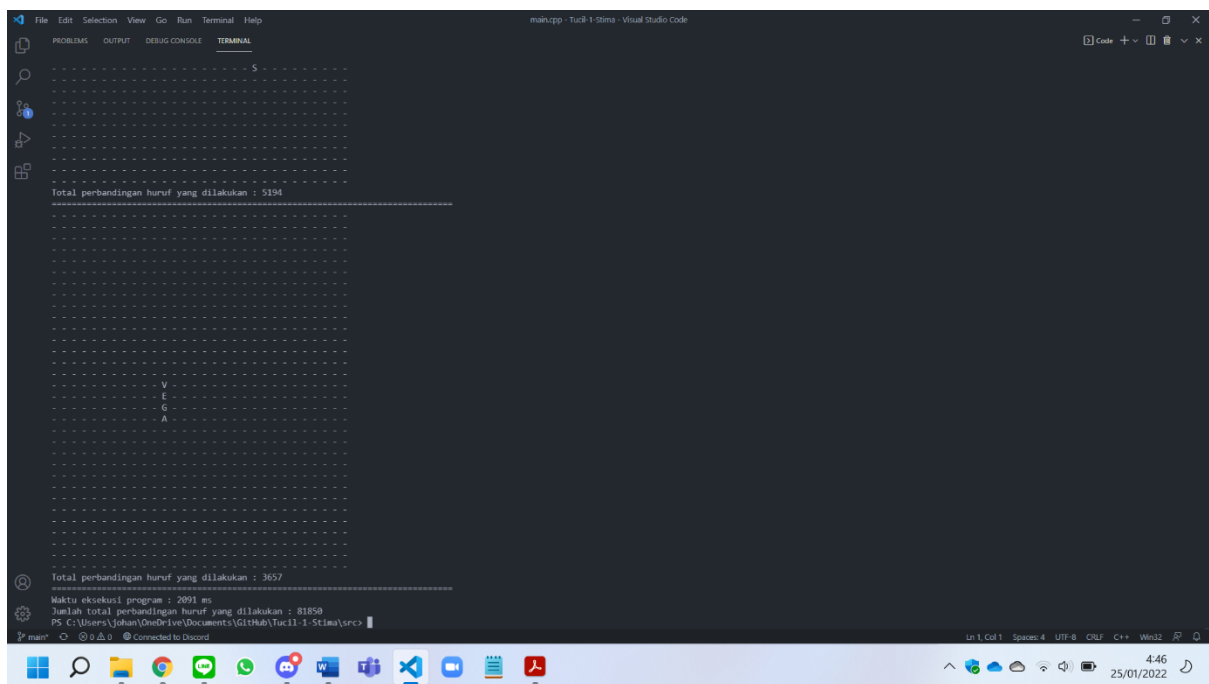
Total perbandingan huruf yang dilakukan : 3145
-----
Waktu eksekusi program : 2553 ms
Jumlah total perbandingan huruf yang dilakukan : 114539
PS C:\Users\johan\OneDrive\Documents\GitHub\Tucil 1-Stima> src
main
Ln 1, Col 1  Spaces: 4  UTF-8  CRLF  C++  Win32  443 25/01/2022

```


7. 32x30x25.txt



```
main.cpp - Tucil-1-Stima - Visual Studio Code
PROBLEMAS OUTPUT DEBUG CONSOLE TERMINAL
PowerShell 7.2.1
Copyright (c) Microsoft Corporation.
https://aka.ms/powershell
Type 'help' to get help.
PS C:\Users\johan\OneDrive\Documents\GitHub\Tucil-1-Stima> cd "c:\Users\johan\OneDrive\Documents\GitHub\Tucil-1-Stima\src\"; if ($?) { g++ main.cpp -o main }; if ($?) { .\main }
Masukkan nama file : 32x30x25.txt
Masukkan jumlah baris : 32
Masukkan jumlah kolom : 30
Masukkan jumlah kata kunci : 25
-----
- A -
- C -
- R -
- U -
- X -
-----
Total perbandingan huruf yang dilakukan : 112
-----
```



```
main.cpp - Tucil-1-Stima - Visual Studio Code
PROBLEMAS OUTPUT DEBUG CONSOLE TERMINAL
-----
- S -
-----
Total perbandingan huruf yang dilakukan : 5194
-----
-----
- V -
- E -
- G -
- A -
-----
Total perbandingan huruf yang dilakukan : 3657
-----
Waktu eksekusi program : 2091 ms
Jumlah total perbandingan huruf yang dilakukan : 81858
PS C:\Users\johan\OneDrive\Documents\GitHub\Tucil-1-Stima\src>
```

8. 34x34x50.txt

The screenshot shows the Visual Studio Code editor with a terminal window open. The terminal displays the following text:

```
PowerShell 7.2.1
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.

PS C:\Users\johan\OneDrive\Documents\GitHub\Tucil-1-Stima> cd "c:\Users\johan\OneDrive\Documents\GitHub\Tucil-1-Stima\src\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
Masukkan nama file : 3434cd0.txt
Masukkan jumlah baris : 34
Masukkan jumlah kolom : 34
Masukkan jumlah kata kunci : 50
```

The status bar at the bottom indicates the file is named 'main', is in the 'main' directory, and is using the 'UTF-8' encoding. The system clock shows 4:48 on 25/01/2022.

The image shows a Visual Studio Code editor window with a C++ file named 'main.cpp'. The editor is displaying a recursive function 'perbandingan' that counts the number of letters in a string. The function is defined as follows:

```
int perbandingan(string s) {
    if (s.length() == 0) return 0;
    return 1 + perbandingan(s.substr(1));
}
```

The main function calls 'perbandingan' twice: first with 'BORNEO' and then with 'BORBORNEZ'. The output of the program is shown in the terminal window at the bottom of the editor. The output for 'BORNEO' is 496, and for 'BORBORNEZ' it is 6846. The status bar at the bottom of the editor shows 'Ln 1, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', 'C++', 'Win32', and 'P'. The taskbar at the bottom of the screen shows various application icons, including the Windows Start button, File Explorer, Google Chrome, and several communication apps like WhatsApp, Telegram, and Discord. The system clock in the bottom right corner shows the time as 4:48 on 25/01/2022.

9. 35x35x47.txt

The screenshot shows the Visual Studio Code editor with a terminal window open. The terminal displays the output of a C++ program that reads a file named '35x35od7.txt' and prints the number of lines, columns, and words. The program output is as follows:

```
PowerShell 7.2.1
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.

PS C:\Users\johan\OneDrive\Documents\GitHub\Tucil-1-Stima> cd "c:\Users\johan\OneDrive\Documents\GitHub\Tucil-1-Stima\src\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
Masukkan nama file : 35x35od7.txt
Masukkan jumlah baris : 35
Masukkan jumlah kolom : 35
Masukkan jumlah kata kunci : 47
```

The Windows taskbar at the bottom shows the time as 4:50 on 25/01/2022. The system tray includes icons for network, volume, and battery status.

```

main.cpp - TUCIL 1-Stima - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
-----
A
D
A
Y
A

Total perbandingan huruf yang dilakukan : 7442

-----
A
D
A
Y
A

Total perbandingan huruf yang dilakukan : 7397

-----
Waktu eksekusi program : 4570 ms
Jumlah total perbandingan huruf yang dilakukan : 18846
PS C:\Users\johan\OneDrive\Documents\GitHub\Tucil-1-Stima> src

```

No.	Poin	Ya	Tidak
1.	Program berhasil dikompilasi tanpa kesalahan (<i>no syntax error</i>)	✓	
2.	Program berhasil <i>running</i>	✓	
3.	Program dapat membaca file masukan dan menuliskan luaran	✓	
4.	Program berhasil menemukan semua kata di dalam <i>puzzle</i>	✓	

Alamat repository GitHub : <https://github.com/johannes-ws/Tucil-1-Stima>