

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

```
File Edit Selection View Go Run Terminal Help
main.cpp - huB 1-0ima - Visual Studio Code
PROBLEMS 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 : idc12x16.txt
Masukkan jumlah baris : 16
Masukkan jumlah kolom : 12
Masukkan jumlah kata kunci : 16

-----
- A D A -
-----
Total perbandingan huruf yang dilakukan : 747
-----
- F -
- O -
- R -
- T -
- R -
- A -
- N -
-----
Total perbandingan huruf yang dilakukan : 67
-----
```

[illegible]

2. 14x14x8.txt

The image shows a screenshot of a Windows desktop with Visual Studio Code open. The code editor displays a C++ program named 'main.cpp' which implements a letter frequency analysis. The program prompts the user for a file name, the number of lines, and the number of columns. It then reads the file and counts the occurrences of each letter (A-Z) in the text. The results are displayed as a table with columns for the letter and its count. The program also calculates the total number of letter comparisons performed. The output shows that for the file '14-144d.txt', there were 1315 comparisons. The status bar at the bottom indicates the file is 'main.cpp', the cursor is at line 1, column 1, and the file is UTF-8 encoded. The system tray at the bottom shows the date and time as 4:29 on 25/01/2022.

The image shows a Visual Studio Code editor window with a C program open. The program is designed to count the frequency of each letter in a given string. It uses an array of 26 integers to store the counts for each letter of the alphabet. The string being processed is "SALAM GUJUNG". The terminal output shows the following results:

```
-----  
S  
A  
L  
M  
-----  
Total perbandingan huruf yang dilakukan : 107  
-----  
S  
L  
I  
M  
E  
-----  
Total perbandingan huruf yang dilakukan : 183  
-----  
S  
L  
I  
M  
S  
-----  
Total perbandingan huruf yang dilakukan : 802  
-----  
Waktu eksekusi program : 136 ms  
Jumlah total perbandingan huruf yang dilakukan : 5053  
PS C:\Users\johan\OneDrive\Documents\GitHub\Tucil-1-Stima>
```


The status bar at the bottom indicates the file is named "main.c", the encoding is UTF-8, and the language is C++. The system tray shows the date and time as 4:29 on 25/01/2022.

3. 16x16x6.txt

[illegible][illegible]

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 : 00-180d.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, a search icon, 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 for character frequency analysis. The program reads a string 'YANFEI' and counts the frequency of each character. The output shows the total number of comparisons (2246 and 2185) and the frequency of each character (Y: 5, A: 0, N: 8, F: 6, E: 1, I: 5). The program also displays the execution time (172 ms) and the total number of comparisons (8899).

```
main.cpp - Tucil 1-Stima - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
- l -
Total perbandingan huruf yang dilakukan : 2246
-----
Y - 5 -
A - 0 -
N - 8 -
F - 6 -
E - 1 -
I - 5 -
-----
Total perbandingan huruf yang dilakukan : 2185
-----
Y A N F E I
-----
Total perbandingan huruf yang dilakukan : 1925
-----
Waktu eksekusi program : 172 ms
Jumlah total perbandingan huruf yang dilakukan : 8899
PS C:\Users\Johan\OneDrive\Documents\GitHub\Tucil 1-Stima> src
```

Ln 1, Col 1 | Spaces: 4 | UTF-8 | CRLF | C++ | Win32 | 4:34 25/01/2022

5. 22x22x50.txt

The image shows a Windows 10 desktop environment. In the foreground, a Visual Studio Code (VS Code) editor window is open, displaying a C++ source file named 'mah.cpp'. The code is a simple program that prompts the user for a filename, the number of lines, and the number of words, and then prints the count of words. The terminal output shows the program running successfully, with the user inputting '22' for the number of lines and '50' for the number of words. The VS Code interface includes a menu bar at the top with options like File, Edit, Selection, View, Go, Run, Terminal, and Help. The bottom status bar of VS Code shows the current line and column (Ln 1, Col 1), the number of spaces (4), and the active language (C++). The Windows taskbar at the bottom of the screen shows various icons, including the Start button, Search, File Explorer, Google Chrome, WhatsApp, Telegram, Discord, and several other applications. The system tray in the bottom right corner shows the date and time as 4:38 on 25/01/2022.

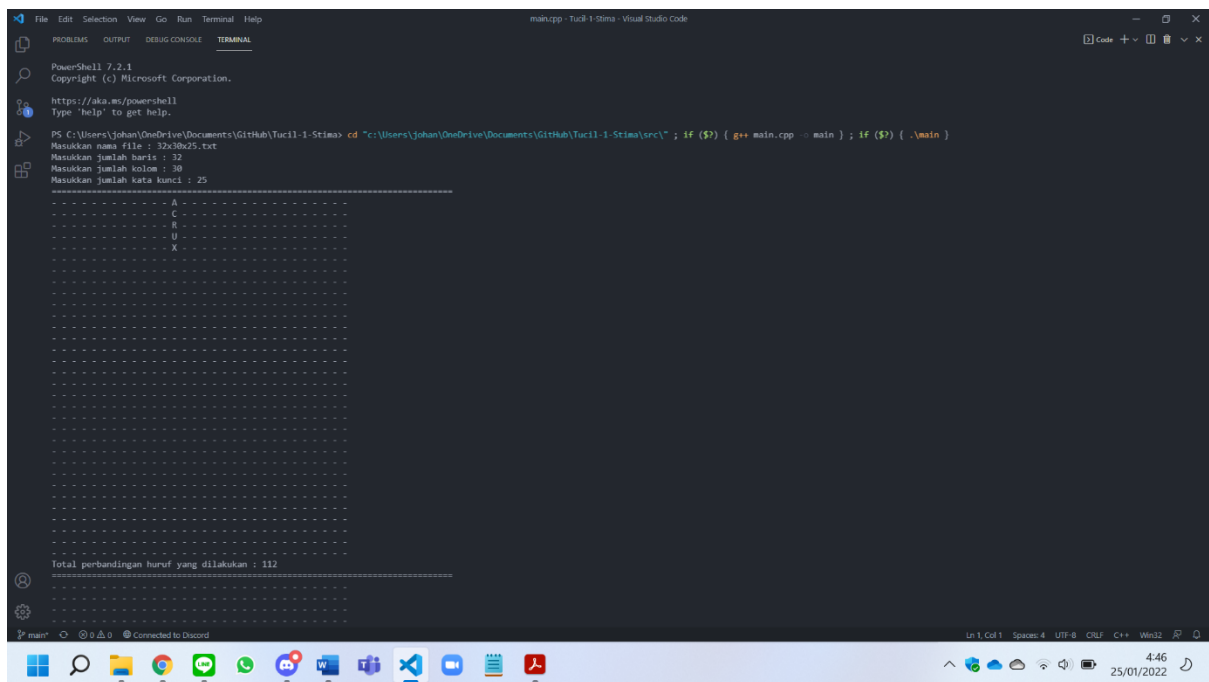
[illegible]

6. 24x24x50.txt

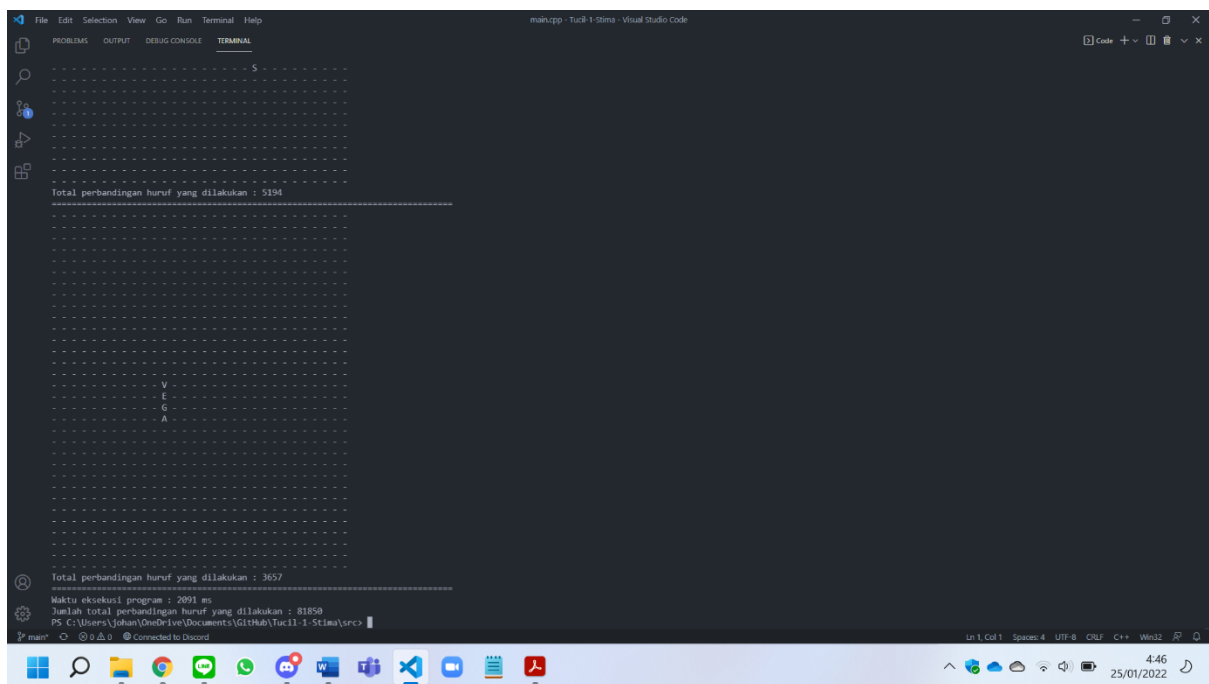
The screenshot shows the Visual Studio Code editor with a file named 'mah.cpp' open. The code defines a struct 'Mahasiswa' with fields 'nama' (string), 'baris' (int), 'kolom' (int), and 'kata_kunci' (string). It includes functions for inputting student data ('MasukkanData'), calculating row and column indices ('HitungBarisKolom'), and printing the data ('PrintData'). The main function prompts the user for a filename, reads it, calls 'MasukkanData', calculates indices, and prints the result. The output window at the bottom shows the successful compilation of 'mah.exe' and its execution, displaying the prompt 'Masukkan nama file :'. The status bar at the very bottom indicates the current position is Line 1, Column 1, with 4 spaces, UTF-8 encoding, C++ language, and Win32 architecture.

The image shows a Visual Studio Code editor window with a C++ program open. The program is named 'main.cpp' and is located in the 'Tucil 1 - Stima' project. The code is a C++ program that takes a string input and counts the frequency of each letter (A-Z) using an array. The program also calculates the total number of comparisons made. The output of the program is displayed in the terminal window, showing the total number of comparisons (2538) and the frequency of each letter. The program is run in a terminal window, and the output is displayed in the console. The status bar at the bottom indicates the file is encoded in UTF-8, the cursor is at line 1, column 1, and the editor is in C++ mode. The system tray at the bottom shows the date and time as 25/01/2022, 4:43.

7. 32x30x25.txt



```
main.cpp - Tucil-1-Stima - Visual Studio Code
PROBLEMS 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
PROBLEMS 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 image shows a Windows 10 desktop environment. In the foreground, a Visual Studio Code (VS Code) editor window is open, displaying a C++ file named 'mah.cpp'. The code is a simple program that prompts the user for a filename, the number of lines, and the number of words, then calculates the total number of characters. The terminal output shows the program running successfully, with the user inputting '442420.txt', '34', and '50'. The VS Code window title is 'mah.cpp - Tucil 1-Stima - Visual Studio Code'. The Windows taskbar at the bottom shows various icons including the Start button, Search, File Explorer, Chrome, Discord, and several other applications. The system tray shows the date and time as 4:48 on 25/01/2022.

The screenshot shows a Visual Studio Code editor with a terminal window open. The terminal displays the output of a C++ program that calculates the total number of comparisons for a given string. The input string is "YORBBORRONEZ", which has a length of 11. The program outputs "Total perbandingan huruf yang dilakukan : 6846" and "Waktu eksekusi program : 4663 ms". The terminal window has a dark theme and shows the file explorer on the left with a file named "main.cpp".

```

File Edit Selection View Go Run Terminal Help
main.cpp - Tuci11-Stima - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
-----
Total perbandingan huruf yang dilakukan : 496
-----
Total perbandingan huruf yang dilakukan : 6846
-----
Waktu eksekusi program : 4663 ms
Jumlah total perbandingan huruf yang dilakukan : 198687
PS C:\Users\johan\OneDrive\Documents\GitHub\Tuci11-Stima\src>

```

9. 35x35x47.txt

The screenshot shows the Visual Studio Code editor with a file named `main.cpp` open. The code is a C++ program that reads a filename, the number of rows, and the number of columns from the user, and then reads that many words from the input. The terminal window at the bottom shows the program's execution output.

```

File Edit Selection View Go Run Terminal Help
main.cpp - Tucil 1-Stima - Visual Studio Code

PROBLEMS 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 : 35x35d7.txt
Masukkan jumlah baris : 35
Masukkan jumlah kolom : 35
Masukkan jumlah kata kunci : 47
  
```

The status bar at the bottom indicates the current file is `main.cpp`, the cursor is at line 1, column 1, and the file is encoded in UTF-8 with CRLF line endings. The system clock shows 4:50 on 25/01/2022.

[illegible]

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