

LAPORAN TUGAS KECIL I

PENYELESAIAN *WORD SEARCH PUZZLE*

DENGAN ALGORITMA *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	1
Algoritma Brute Force	2
Source Program	3
Screenshot Input dan Output	6
Link Drive Kode Program	30
Checklist	30

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 bernilai 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

Menggunakan bahasa pemrograman C++ dengan pertimbangan library yang sudah ada dan runtime yang cepat.

```
#include <iostream>
#include <string>
#include <fstream>
#include <sstream>
#include <vector>
#include <chrono>
using namespace std;
using namespace std::chrono;

typedef vector<vector<char>> matrix;

matrix loadMatrix(ifstream& file);
vector<string> loadAnswers(ifstream& file);
void printMatrix(matrix mat);
void check(matrix mat, string answer, int r, int c, int dr, int dc);
string clean(string s);

int counter;

int main() {
    string filename;
    char isPuzzleDone;
    do {
        cout << "======" << endl;
        cout << "Masukkan nama file : ";
        cin >> filename;
        string filedir = "../test/" + filename;
        ifstream file(filedir.c_str());
        matrix board;
        board = loadMatrix(file);
        printMatrix(board);
        vector<string> answers = loadAnswers(file);
        auto start = high_resolution_clock::now();
        for (const string& answer : answers) {
            for (int i = 0; i < board.size(); i++) {
                for (int j = 0; j < board[i].size(); j++) {
                    for (int dr = -1; dr <= 1; dr++) {
                        for (int dc = -1; dc <= 1; dc++) {
                            if (dr == 0 && dc == 0) continue;
                            check(board, answer, i, j, dr, dc);
                        }
                    }
                }
            }
        }
        auto stop = high_resolution_clock::now();
        auto duration = duration_cast<microseconds>(stop - start);
        cout << "Time taken by program: " << duration.count() << "seconds" << endl;
        cout << "Jumlah total perbandingan huruf yang dilakukan untuk menemukan kata di dalam puzzle: " << counter << " kali" << "\n\n";
        file.close();

        isPuzzleDone = 'n';
        cout << "Apakah anda ingin memecahkan puzzle lain? (y/n)" << endl;
        cin >> isPuzzleDone;
    } while ((isPuzzleDone == 'Y') || (isPuzzleDone == 'y'));
    cout << "======" << endl;
    cout << "\nWord search puzzle solver by Kristo Abdi 13520058." << endl;
```

```

matrix loadMatrix(istream& file) {
    string line;
    matrix mat;
    while (getline(file, line)) {
        line = clean(line);
        if (line.length() == 0) break; //EOF

        vector<char> row;
        for (const char& cc : line) {
            if (cc >= 'A' && cc <= 'Z') {
                row.push_back(cc);
            }
        }
        mat.push_back(row);
    }

    return mat;
}

vector<string> loadAnswers(istream& file) {
    vector<string> answers;
    string line;
    while (getline(file, line)) {
        if (line.length() == 0) break;
        answers.push_back(clean(line));
    }
    return answers;
};

```

```

void printMatrix(matrix mat) {
    for (int i = 0; i < mat.size(); i++) {
        for (int j = 0; j < mat[i].size(); j++) {
            cout << mat[i][j] << " ";
        }
        cout << endl;
    }
}

string clean(string s) {
    // sometimes the file can have CRLF ending, so we need to clean the string first before using it.
    if (s.length() == 0) return s;
    if (s[s.length()-1] != '\x0D') return s;
    return s.substr(0, s.length()-1);
}

```

```

void check(matrix mat, string answer, int r, int c, int dr, int dc) {
    bool correct = true;
    matrix res = mat;
    matrix temp = mat;
    for (int i = 0; i < answer.length(); i++) {
        int ri = r + dr * i;
        int ci = c + dc * i;
        if (
            ri < 0 || ri >= mat.size() || // overflow vertically
            ci < 0 || ci >= mat[ri].size() || // overflow horizontally
            mat[ri][ci] != answer[i]
        ) {
            correct = false;
            break;
        } else {
            temp[ri][ci] = '1';
        }
        counter++;
    }

    if (!correct) {
        return;
    }

    for (int i = 0; i < mat.size(); i++) {
        for (int j = 0; j < mat[i].size(); j++) {
            if (temp[i][j] != '1') {
                res[i][j] = '-';
            }
        }
    }
    printMatrix(res);
    cout << endl;
}

```

Screenshot Input dan Output

Test 1. (Kecil 12x18)

```
D:\Tucil\word-search-puzzle-solver\src (master -> origin)
$ main
=====
Masukkan nama file : small.txt
YTIKBBARBARANOELLEQ
VIJBSSSMROSARIADQMf
BYVESNKRONYHUTAOAS
IOQILUIIYXIANGLINGH
PIKDINCNABUYREULAE
KM SOSBARGNQ TUDUZKN
EIGUAUSMOGFCNHIDMH
QYAWVZHA BSUEEQKJORE
IANBAALCYEEAIIILINS
NPYNLJJY KQURONKKQENA
GIUSJOZF FISCHLCIEF
JEANSKKKOMIA YAKAZ

-----
- A -
- M -
- B -
- E -
R
-----
- AYAKA -
```

В
Е
И
О
О
О
У

д
- и -
О
и
А

EULA

FISCHL

G
A
N
Y
U

HUTAO

NIGUGUAN

KELLE

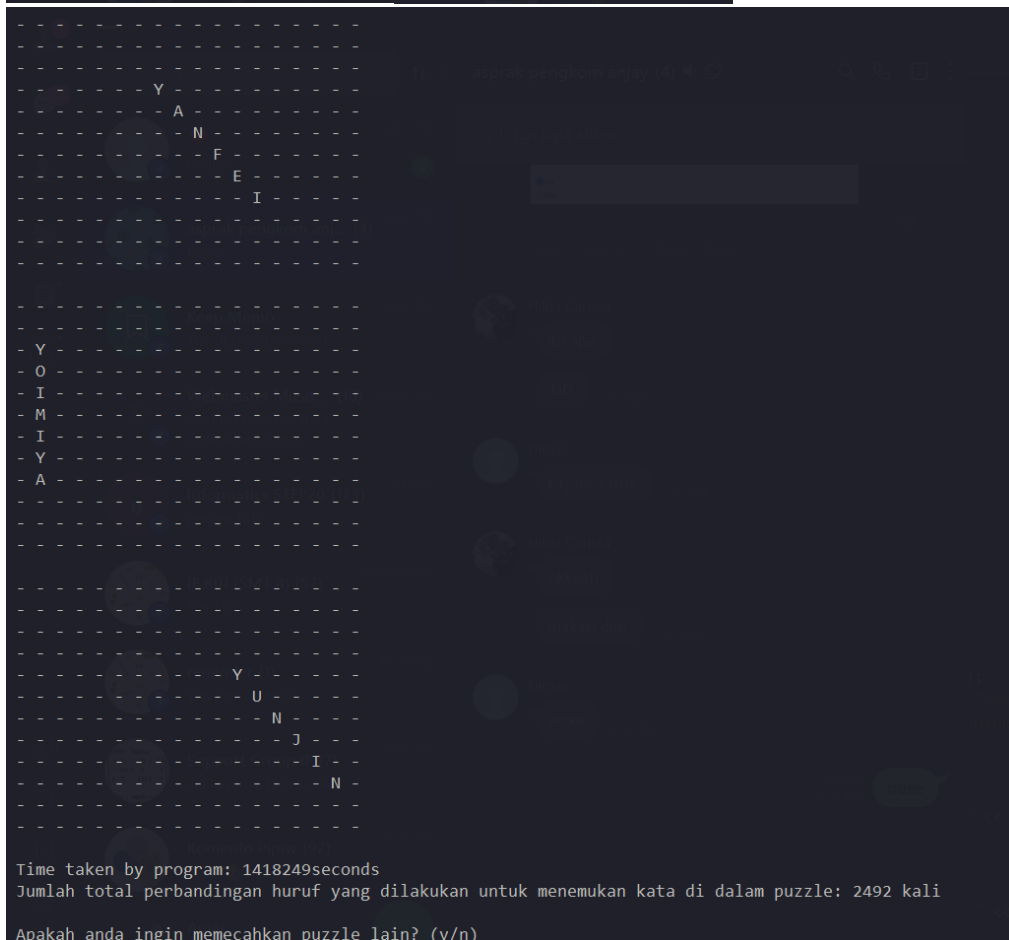
JEAN

KOKOMI

LISSA

QUEEN

QIQI



Test 2. (Kecil 14x14)

[illegible]

[illegible]

```
- - - - -  
- - - - -  
- - - - -  
- W A R P E D - - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
Time taken by program: 751485seconds  
Jumlah total perbandingan huruf yang dilakukan untuk menemukan kata di dalam puzzle: 10859 kali
```

Test 3. (Kecil 16x16)

```

Apakah anda ingin memecahkan puzzle lain? (y/n)
y
=====
Masukkan nama file : small13.txt
H Z D E N Q N R E F U J E Y K R
U C Y P H E R A N U T Z Y N E F
V R A T W L M W O U A D I B F S
K Q O E T X Q O T R W F M R D J
Q G U Y R D R A S B G A B K D Q
R E Y N A B N M E H U M C W K
O S R E P I V G I C P R H M I X
C K T J M Q W D R F B X D L P I
J Y E I G O M A B Q K V L U C N
F E T Y D R L V C T Y J J I C R E
E S T G Q B S N V S O H J R T O
Q K A T X O K X O Y C N Q A J H
H A Z G V D J N O E N Q U A L P
B Y Y A E D V H P F H X O W V O
S O F S M M W O C N Z G F C D C
J Z A R T S A I I Q P E O U Q Y

- - - - - A R T S A - - - - -

H C A E R B
- - - - -

- - - - - E R C Y P H E R - - - - -
- - - - - B - - - - -
- - - - - M - - - - -
- - - - - A - - - - -
- - - - - H - - - - -
- - - - - C - - - - -

- - - - - I L L I
- - - - - J L
- - - - - O J
- - - - - Y

- - - - - K A Y O - - - - -

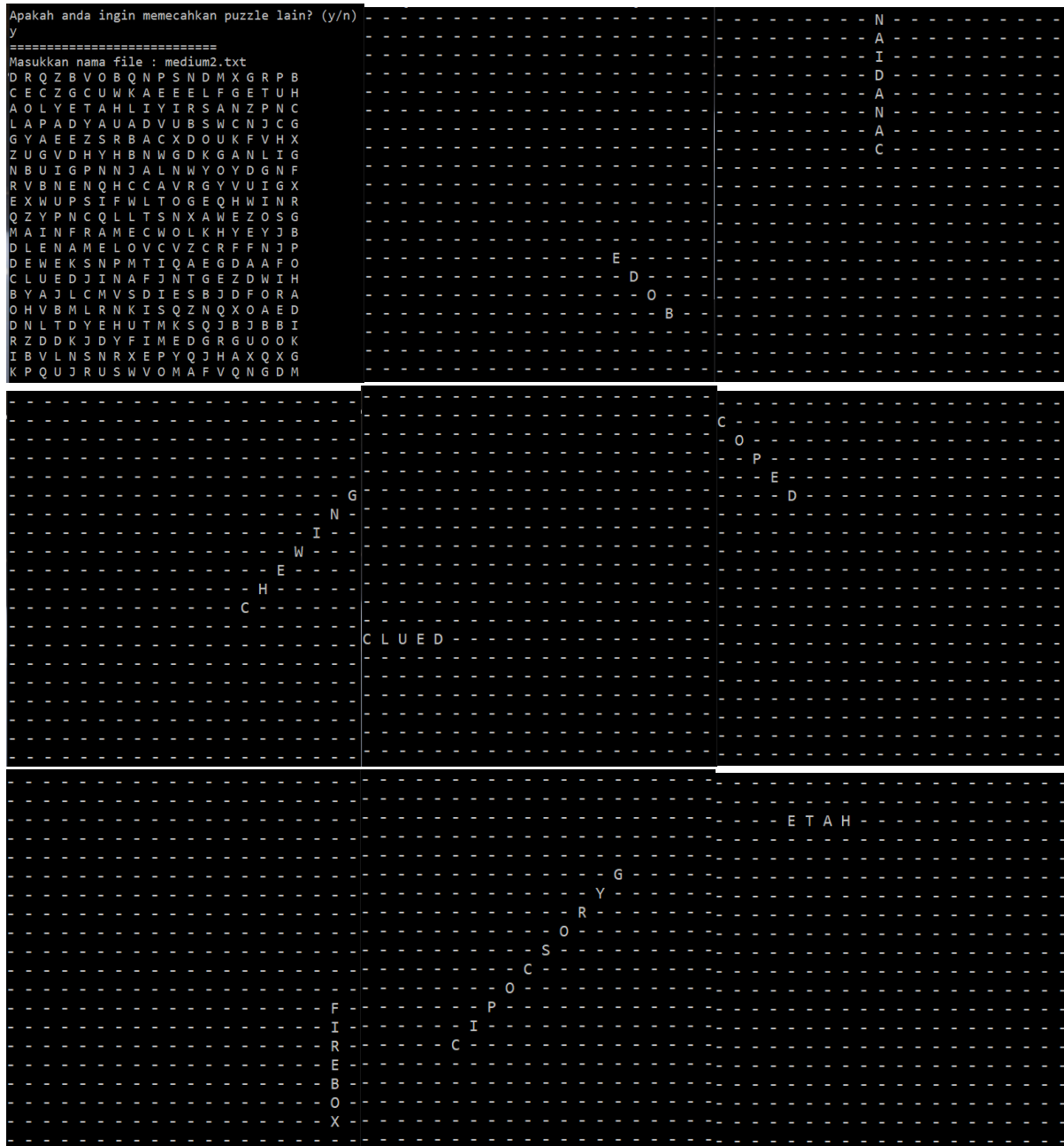
- - - - - N E M O - - - - -
- - - - - X I N E O H P
- - - - - N O E N - - - - -

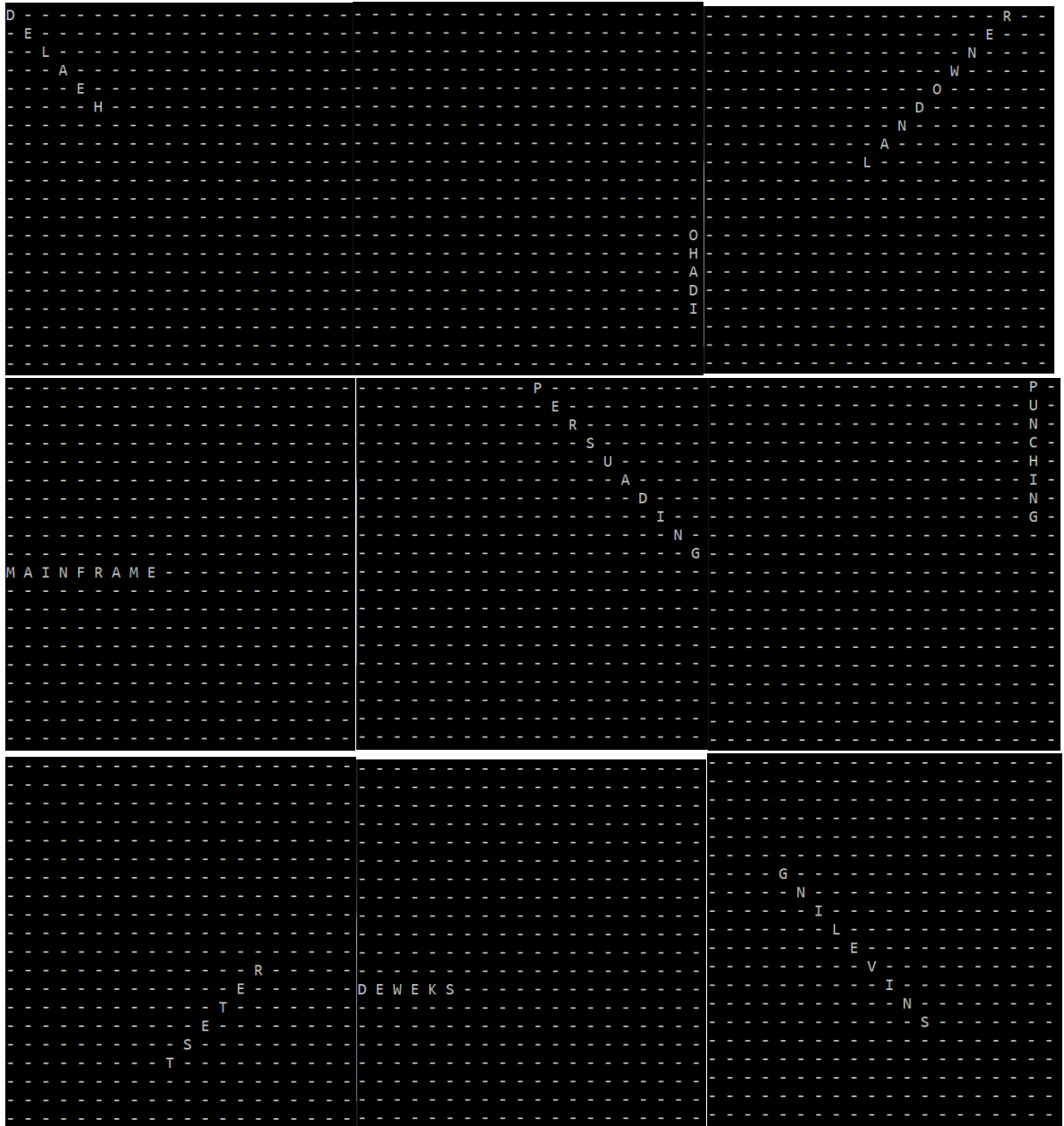
```


Test 4. (Sedang 20x22)

[illegible]

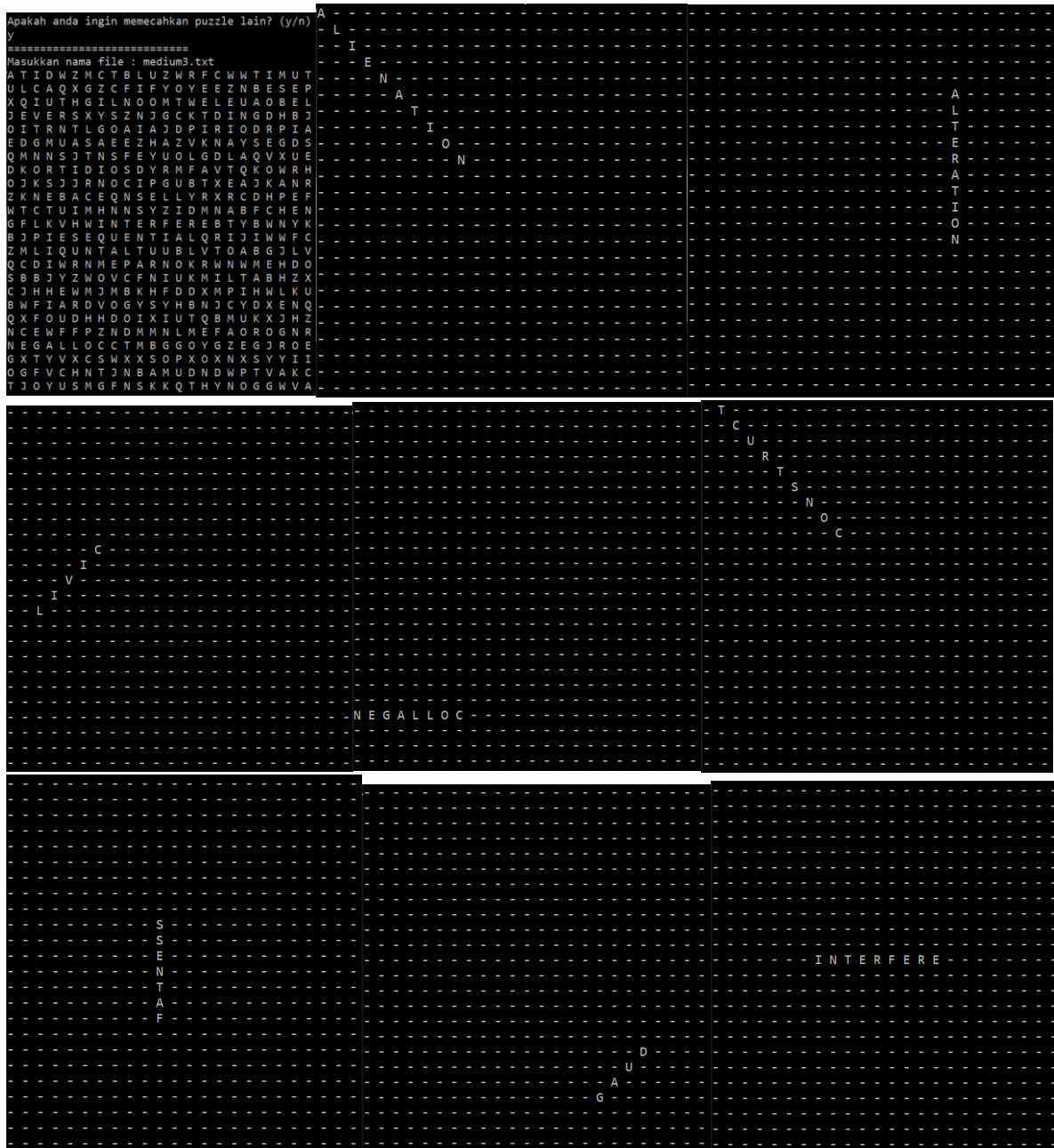
Test 5. (Sedang 20x20)

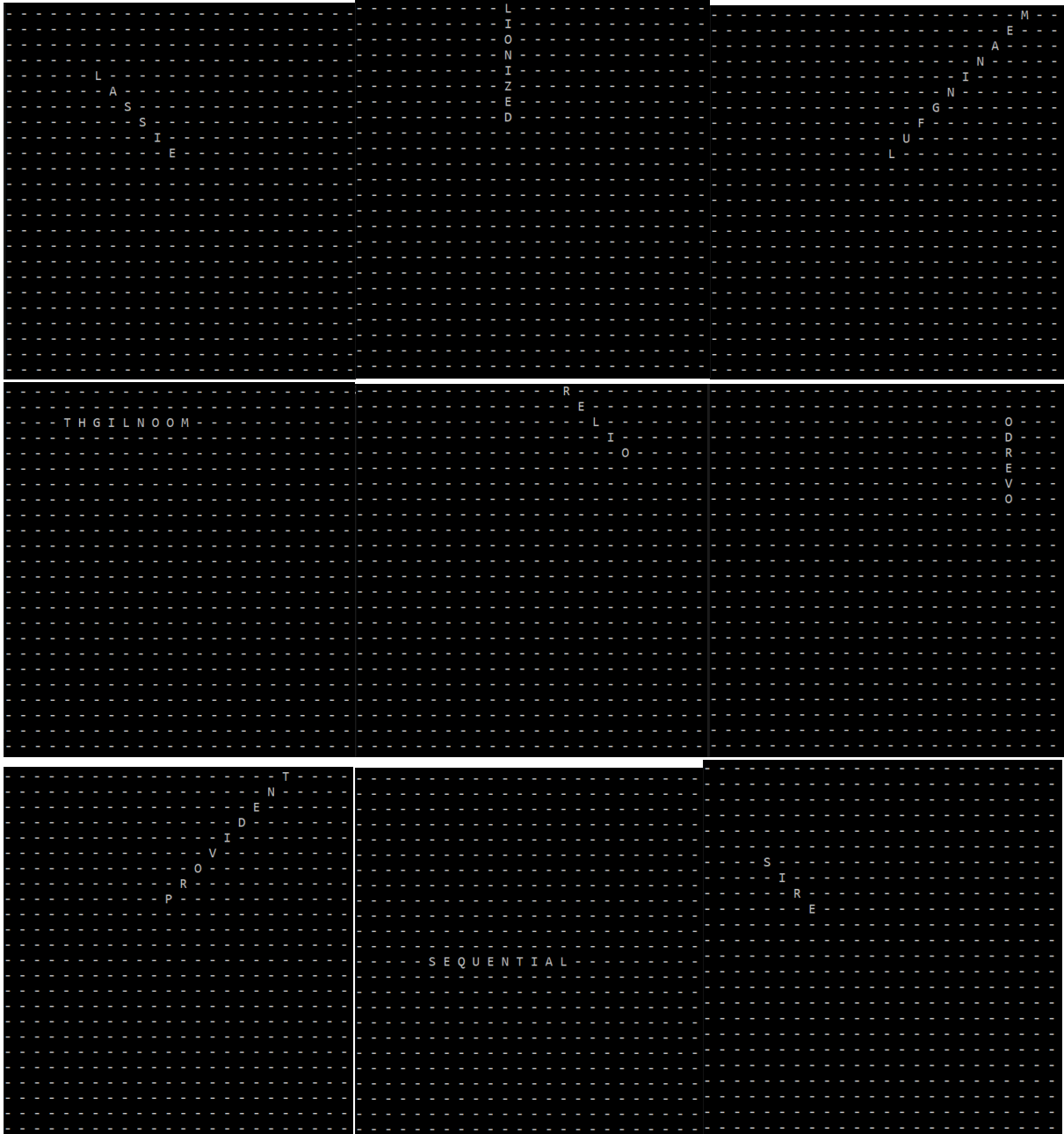





```
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- E N A M E L - - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
Time taken by program: 1954958seconds  
Jumlah total perbandingan huruf yang dilakukan untuk menemukan kata di dalam puzzle: 15162 kali
```

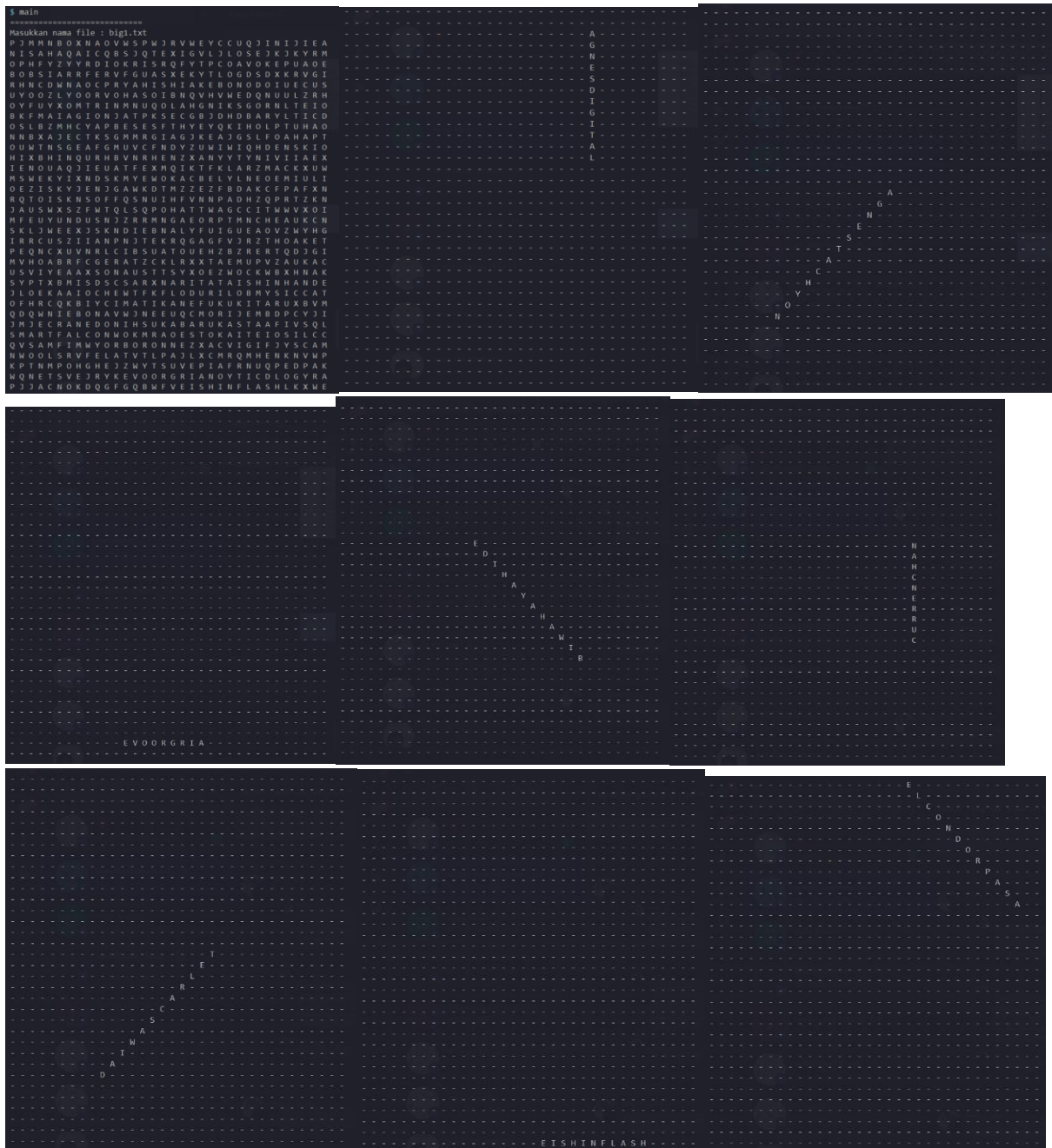
Test 6. (Sedang 24x24)

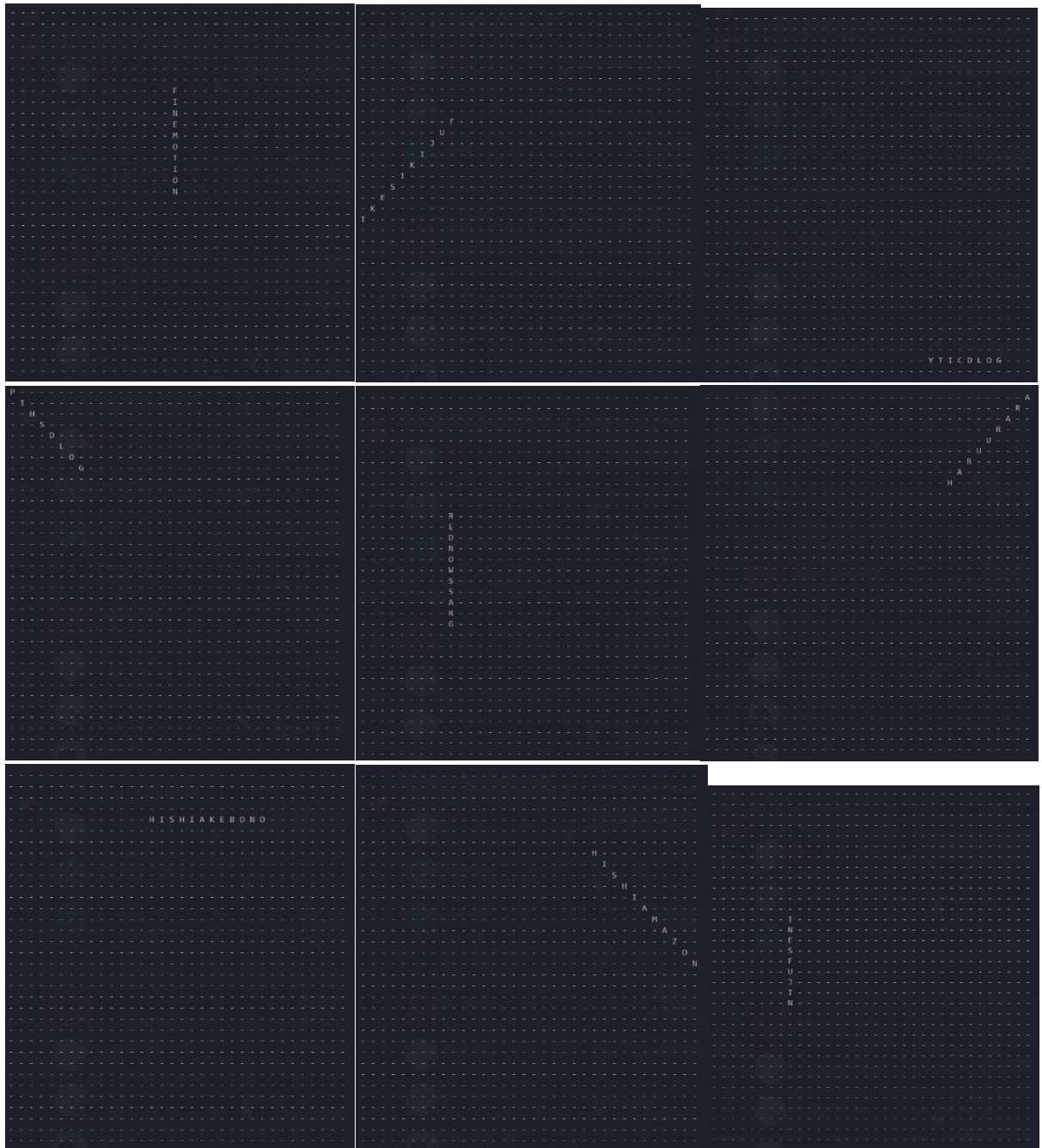


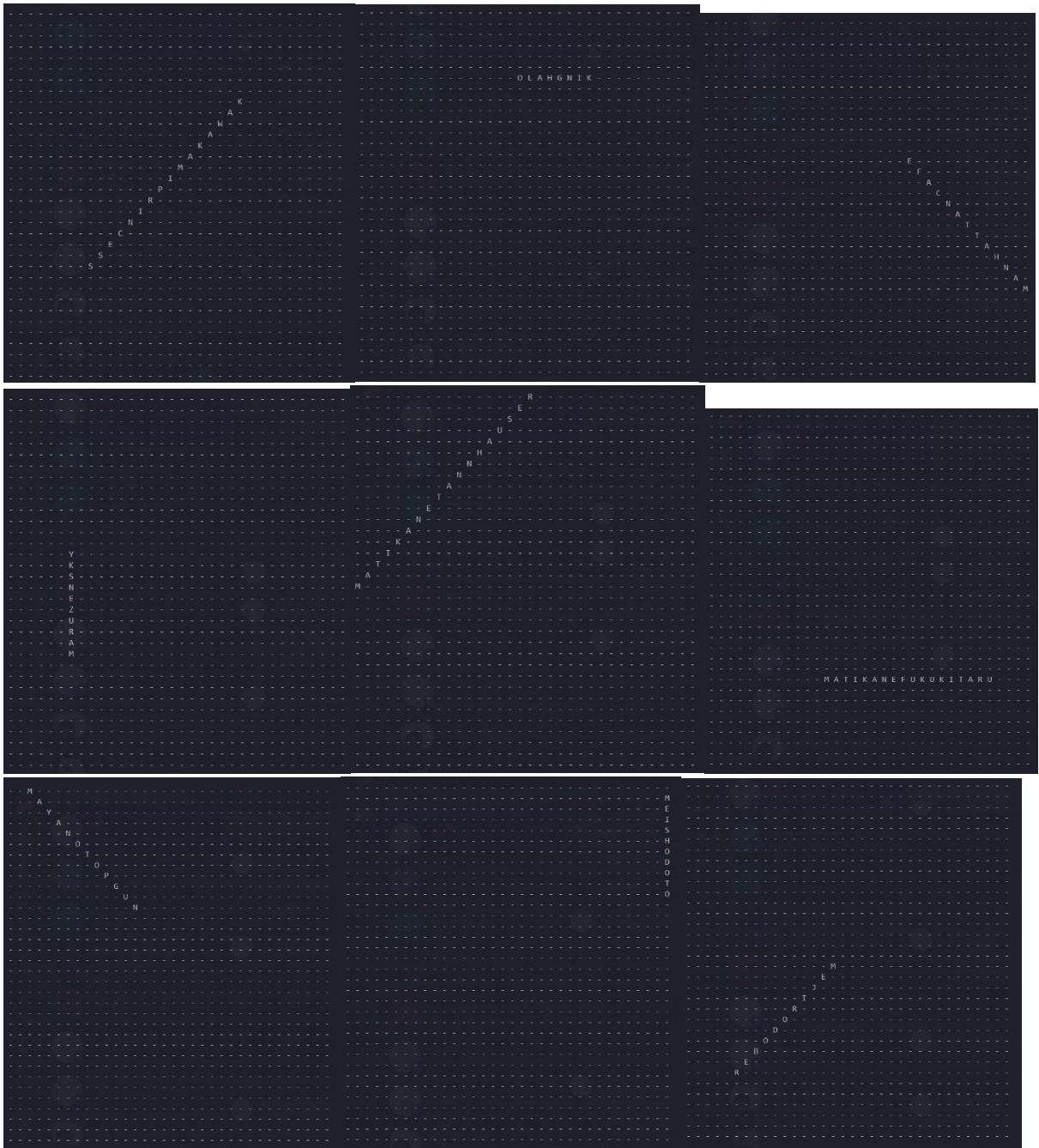


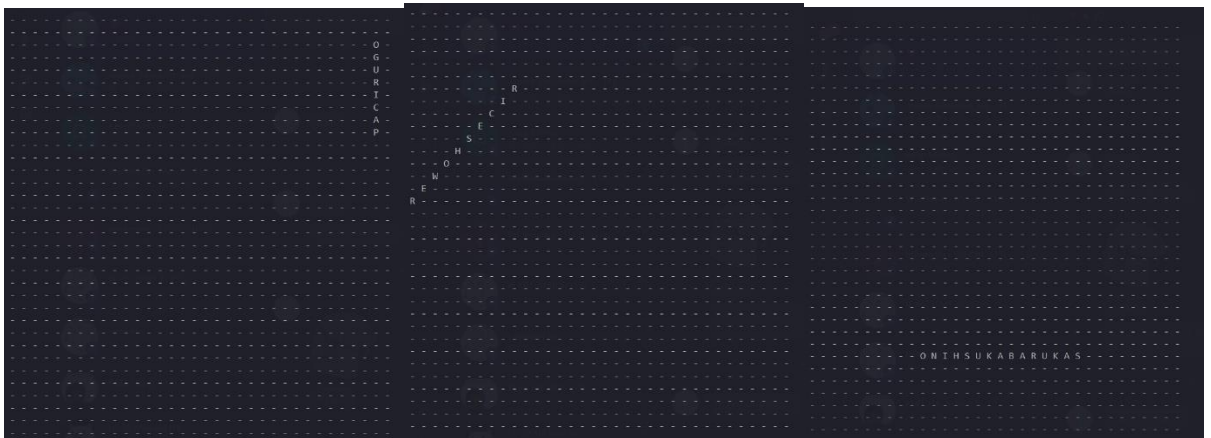
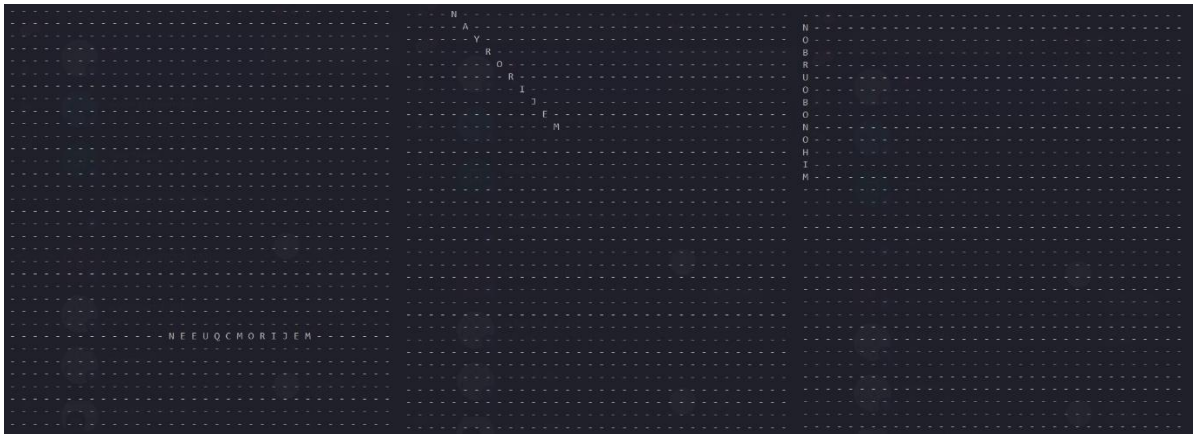
```
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - P I H W - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
Time taken by program: 3153324seconds  
Jumlah total perbandingan huruf yang dilakukan untuk menemukan kata di dalam puzzle: 23353 kali
```

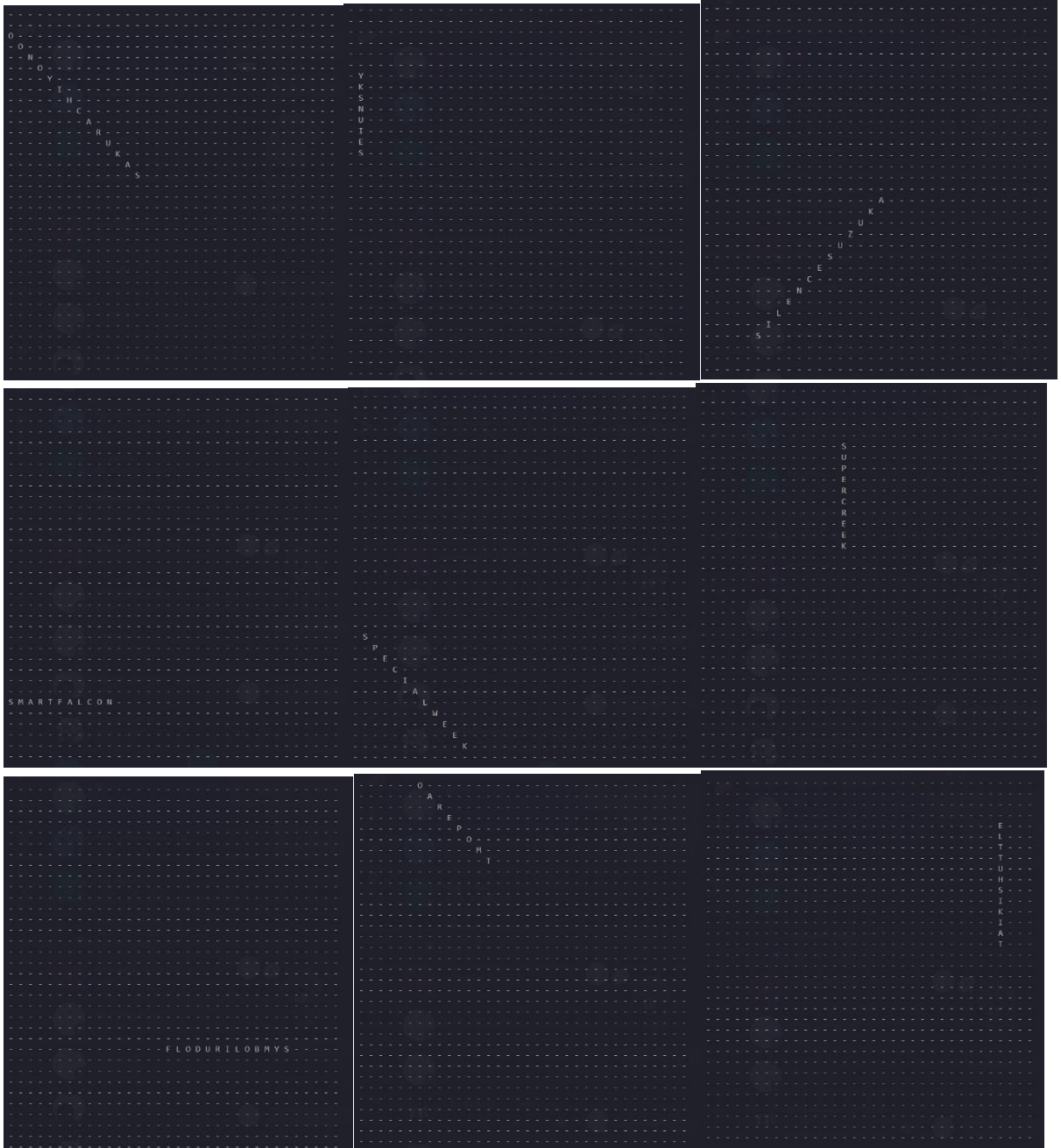
Test 7. (Besar 34x34)







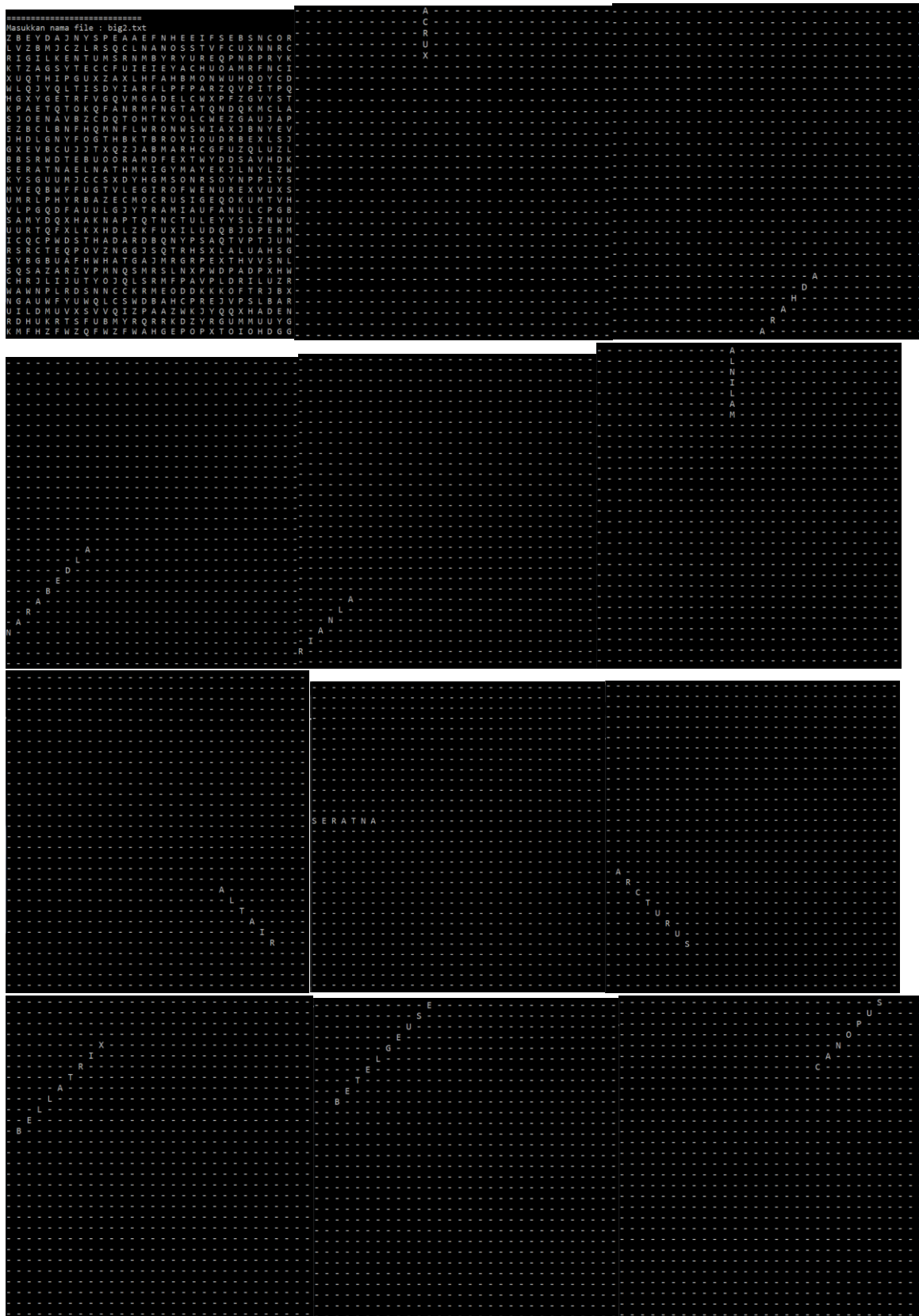


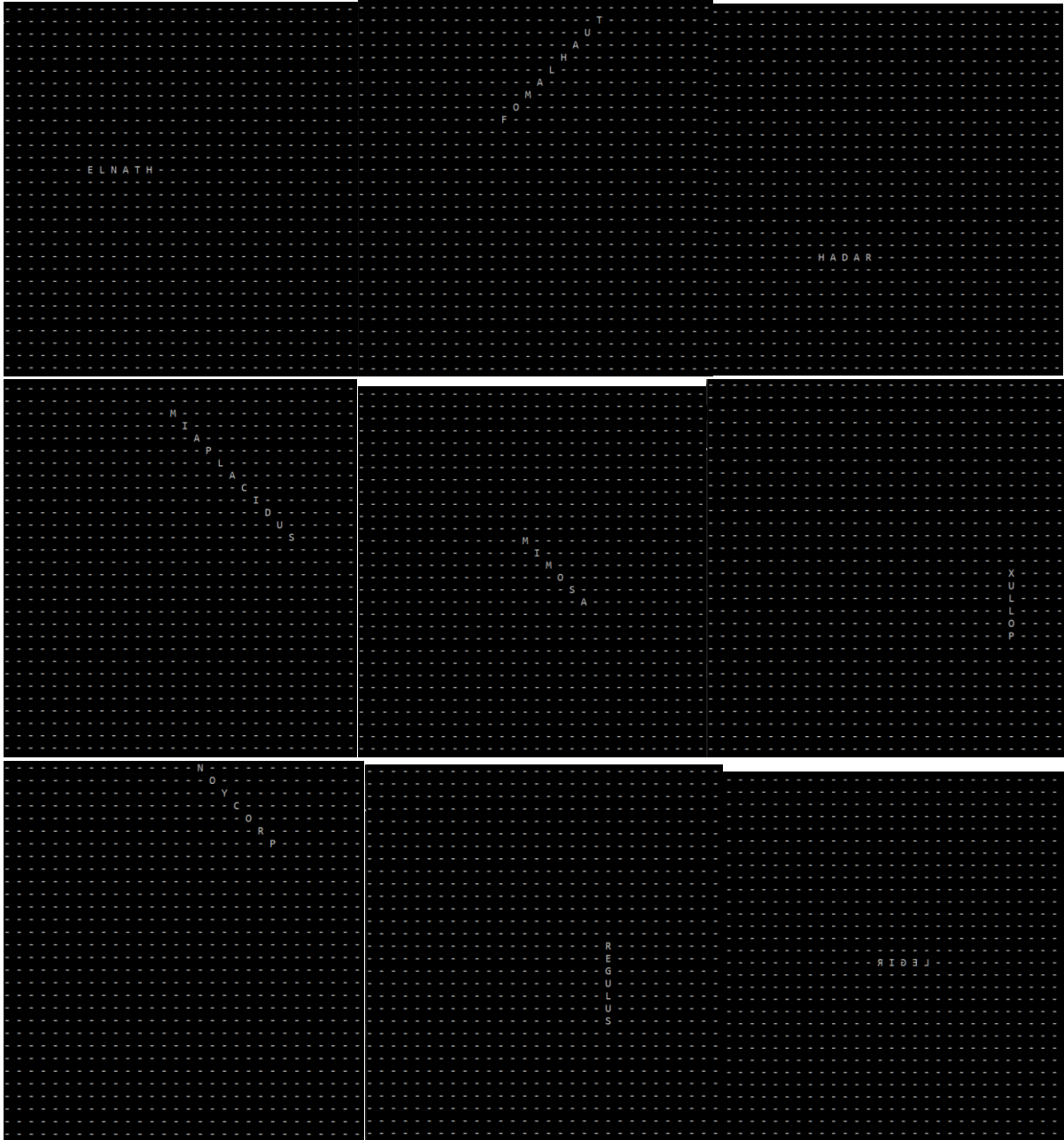


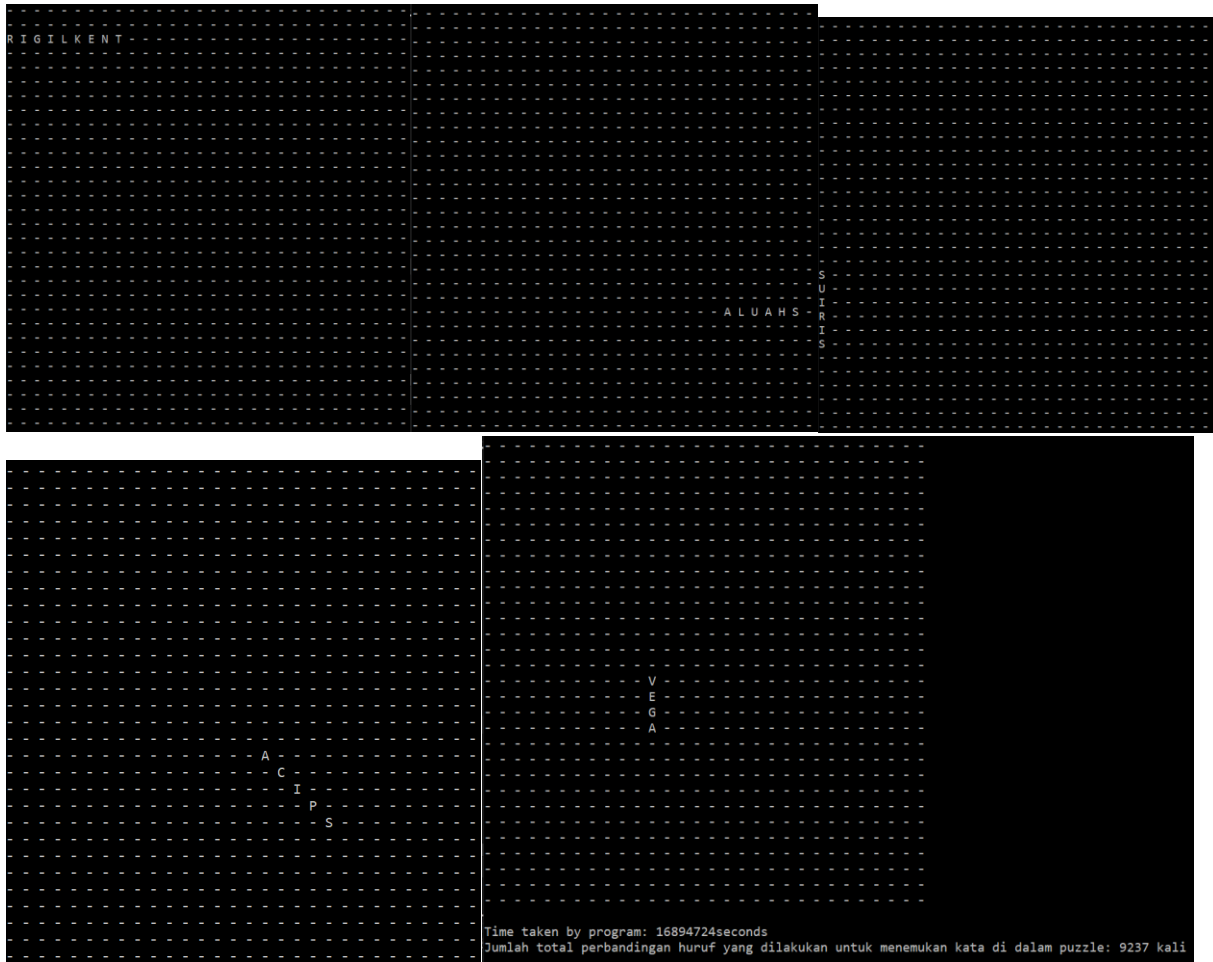


```
Time taken by program: 22176437seconds
Jumlah total perbandingan huruf yang dilakukan untuk menemukan kata di dalam puzzle: 21259 kali
```

Test 8. (Besar 30x30)







Link Drive Kode Program

https://drive.google.com/drive/folders/1Z3Qil6msTq7ytK-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	