

Laporan Tugas Kecil
Word Search Puzzle dengan Algoritma Brute Force



Pembuat:
Vincent Christian Siregar 13520136

Algoritma Brute Force

Tahapan algoritma yang digunakan:

Proses pembacaan input:

1. Puzzle yang dibaca dari file txt disimpan dalam bentuk array multidimensi (2 dimensi) bertipe character.
2. Kata yang akan dicari pada puzzle disimpan dalam bentuk array bertipe string.

Proses algoritma brute force:

1. Program melakukan nested loop sebanyak 3 buah loop. Loop pertama merupakan representasi kata yang akan dicari (misal: kata EARTH merupakan kata yang pertama ingin dicari, maka pada putaran pertama program akan mencari keberadaan EARTH pada puzzle). Loop didalamnya merupakan loop untuk mengakses tiap character pada puzzle yang merupakan array 2 dimensi (terdiri dari 2 loop yaitu loop baris dan loop kolom).
2. Program dimulai dengan mengecek character puzzle pada absis 0 dan ordinat 0. Jika character tersebut sama dengan huruf pertama maka lanjut ke poin 3. Jika character tidak sama maka lanjut ke poin 5.
3. Program akan mengecek character yang berada di sekitar huruf tersebut (maksimal 8 karakter yang ada di sekitar character tersebut). Jika character tersebut sama dengan huruf kedua dari kata yang dicari maka lanjut ke poin 4, jika tidak maka lanjut ke poin 5.
4. Program akan mengecek karakter selanjutnya (sesuai arah dari karakter kedua terhadap karakter pertama). Program akan melakukan loop hingga semua huruf pada kata secara terurut sama dengan yang terdapat pada puzzle. Jika rangkaian kata ditemukan maka lanjut ke poin 6. Jika tidak, lanjut ke poin 5
5. Jika kata atau character tidak valid maka proses berpindah ke character selanjutnya, kembali ke poin 2. Proses berulang sampai menemukan rangkaian karakter yang sesuai dengan kata.
6. Jika rangkaian kata ditemukan, maka program akan menampilkan kata yang ditemukan pada layar sesuai letaknya pada puzzle. Loop dilanjutkan ke kata berikutnya. Proses diulang sampai menemukan seluruh kata pada puzzle.

Source Code

```
import java.io.*;
import java.util.Scanner;

public class WordPuzzle {
    private static final Scanner keyboard_scanner = new Scanner(System.in);

    static Scanner s = null;

    public static int[] RowNColCount(String FileName)
    {
        try {
            s = new Scanner(new File(FileName));
            String RRows = s.nextLine();
            RRows = RRows.replaceAll("\\s", "");

            int Rows = 0;

            String line = "dummy";
            while(s.hasNextLine() && line != "") {
                Rows++;
                line = s.nextLine();
            }
            s.nextLine();

            int total_words=1;

            while(s.hasNextLine()) {
                total_words++;
                s.nextLine();
            }

            s.close();

            int Cols = RRows.length();

            int[] Result = new int[3];

            Result[0] = Rows;
            Result[1] = Cols;
            Result[2] = total_words;

            return Result;
        } catch(Exception e) {
```

```

        int[] Result = new int[1];
        Result[0] = -1;
        s.close();
        return Result;
    }
}

public static void ReadPuzzle(String FileName,char[][]puzzle, String[] words)
{
    try{
        s = new Scanner(new File(FileName));

        int Rows = puzzle.length;
        int Cols = puzzle[0].length;

        int count = 0;
        int count2 = 0;

        while(s.hasNextLine())
        {
            int countL = 0;
            String str = s.nextLine();
            str = str.replaceAll("\\s", "");
            if(count < Rows){
                while (countL < Cols) {
                    puzzle[count][countL] = str.charAt(countL);
                    countL++;
                }
                count++;
            } else
            {
                if (str != "")
                {
                    words[count2] = str;
                    count2++;
                }
            }
        }

    } catch (Exception e){

    }
}

public static void print2D(char mat[][])
```

```

{
    for (int i = 0; i < mat.length; i++){

        for (int j = 0; j < mat[i].length; j++){
            System.out.print(mat[i][j] + " ");
        }
        System.out.println("");
    }
}

public static void PuzzleOutput(int[][]arr, char[][]puzzle)
{
    for (int x=0; x < puzzle.length; x++) {
        for (int y=0; y < puzzle[0].length; y++) {
            if(arr[x][y] == 1)
            {
                System.out.print(puzzle[x][y] + " ");
            } else {
                System.out.print("- ");
            }
        }
        System.out.println("");
    }
    System.out.println("");
}

public static void TurnToZero(int[][]arr)
{
    for (int row = 0; row < arr.length; row++)
    {
        for (int col = 0; col < arr[row].length; col++)
        {
            arr[row][col] = 0;
        }
    }
}

public static int BruteForceWordPuzzle(char[][]puzzle,String[]words,
int[][]Result)
{
    int Rows = puzzle.length;
    int Cols = puzzle[0].length;
    int Pwords = words.length;
    int TotalCompare = 0;

```

```

for (int i=0; i < Pwords ; i++){
    int Pword = words[i].length();
    for (int j=0; j < Rows; j++){
        for (int k=0; k < Cols; k++){
            if(words[i].charAt(0) == puzzle[j][k]){
                boolean Found = false;
                int m;
                int n;
                int counter;

                // cek ke atas
                if(!Found){
                    counter = 1;
                    m = j;
                    n = k;
                    while(m != 0 && counter < Pword) {
                        m--;
                        if (words[i].charAt(counter) != puzzle[m][n]) {
                            TotalCompare++;
                            break;
                        }
                        TotalCompare++;
                        counter++;
                    }
                    if(counter == Pword) {
                        counter = 0;
                        System.out.println(words[i]);

                        int [][]arr = new int[Rows][Cols];

                        TurnToZero(arr);

                        m = j;
                        n = k;
                        while(counter < Pword) {
                            arr[m][n] = 1;
                            m--;
                            counter++;
                        }

                        PuzzleOutput(arr,puzzle);
                    }
                }
            }
        }
    }

    // cek ke kiri

```

```

        if(!Found){
            counter = 1;
            m = j;
            n = k;
            while(n != 0 && counter < Pword) {
                n--;
                if (words[i].charAt(counter) != puzzle[m][n]) {
                    TotalCompare++;
                    break;
                }
                TotalCompare++;
                counter++;
            }
            if(counter == Pword) {
                counter = 0;
                System.out.println(words[i]);

                int [][]arr = new int[Rows][Cols];

                TurnToZero(arr);

                m = j;
                n = k;
                while(counter < Pword) {
                    arr[m][n] = 1;
                    n--;
                    counter++;
                }

                PuzzleOutput(arr,puzzle);
            }
        }

        // cek ke bawah
        if(!Found){
            counter = 1;
            m = j;
            n = k;
            while(m != Rows-1 && counter < Pword) {
                m++;
                if (words[i].charAt(counter) != puzzle[m][n]) {
                    TotalCompare++;
                    break;
                }
            }
            TotalCompare++;

```

```

        counter++;
    }
    if(counter == Pword) {
        counter = 0;
        System.out.println(words[i]);

        int [][]arr = new int[Rows][Cols];

        TurnToZero(arr);

        m = j;
        n = k;
        while(counter < Pword) {
            arr[m][n] = 1;
            m++;
            counter++;
        }

        PuzzleOutput(arr,puzzle);
    }
}

// cek ke kanan
if(!Found){
    counter = 1;
    m = j;
    n = k;
    while(n != Cols-1 && counter < Pword) {
        n++;
        if (words[i].charAt(counter) != puzzle[m][n]) {
            TotalCompare++;
            break;
        }
        TotalCompare++;
        counter++;
    }
    if(counter == Pword) {
        counter = 0;
        System.out.println(words[i]);

        int [][]arr = new int[Rows][Cols];

        TurnToZero(arr);

        m = j;

```



```

        n = k;
        while(counter < Pword) {
            arr[m][n] = 1;
            n++;
            counter++;
        }

        PuzzleOutput(arr,puzzle);
    }
}

// cek ke kiri atas
if(!Found){
    counter = 1;
    m = j;
    n = k;
    while(n != 0 && m != 0 && counter < Pword) {
        m--;
        n--;
        if (words[i].charAt(counter) != puzzle[m][n]) {
            TotalCompare++;
            break;
        }
        TotalCompare++;
        counter++;
    }
    if(counter == Pword) {
        counter = 0;
        System.out.println(words[i]);

        int [][]arr = new int[Rows][Cols];

        TurnToZero(arr);

        m = j;
        n = k;
        while(counter < Pword) {
            arr[m][n] = 1;
            m--;
            n--;
            counter++;
        }

        PuzzleOutput(arr,puzzle);
    }
}

```

```

    }

    // cek ke kiri bawah
    if(!Found){
        counter = 1;
        m = j;
        n = k;
        while(n != 0 && m != Rows-1 && counter < Pword) {
            m++;
            n--;
            if (words[i].charAt(counter) != puzzle[m][n]) {
                TotalCompare++;
                break;
            }
            TotalCompare++;
            counter++;
        }
        if(counter == Pword) {
            counter = 0;
            System.out.println(words[i]);

            int [][]arr = new int[Rows][Cols];

            TurnToZero(arr);

            m = j;
            n = k;
            while(counter < Pword) {
                arr[m][n] = 1;
                m++;
                n--;
                counter++;
            }
            PuzzleOutput(arr,puzzle);
        }
    }

    // cek ke kanan bawah
    if(!Found){
        counter = 1;
        m = j;
        n = k;
        while(n != Cols-1 && m != Rows-1 && counter < Pword) {
            m++;
            n++;

```

```

        if (words[i].charAt(counter) != puzzle[m][n]) {
            TotalCompare++;
            break;
        }
        TotalCompare++;
        counter++;
    }
    if(counter == Pword) {
        counter = 0;
        System.out.println(words[i]);

        int [][]arr = new int[Rows][Cols];

        TurnToZero(arr);

        m = j;
        n = k;
        while(counter < Pword) {
            arr[m][n] = 1;
            m++;
            n++;
            counter++;
        }

        PuzzleOutput(arr,puzzle);
    }
}

// cek ke kanan atas
if(!Found){
    counter = 1;
    m = j;
    n = k;
    while(n != Cols-1 && m != 0 && counter < Pword) {
        m--;
        n++;
        if (words[i].charAt(counter) != puzzle[m][n]) {
            TotalCompare++;
            break;
        }
        TotalCompare++;
        counter++;
    }
    if(counter == Pword) {
        counter = 0;

```

```

        System.out.println(words[i]);
        int [][]arr = new int[Rows][Cols];
        TurnToZero(arr);
        m = j;
        n = k;
        while(counter < Pword) {
            arr[m][n] = 1;
            m--;
            n++;
            counter++;
        }

        PuzzleOutput(arr,puzzle);
    }
}
    }
    TotalCompare++;
}
}
    }
    return TotalCompare;
}
}

public static void main(String[] args){
    System.out.print("Masukkan nama file: ");
    String FileName = keyboard_scanner.next();
    System.out.println();

    int[] RC = RowNColCount(FileName);
    char[][] puzzle = new char[RC[0]][RC[1]];
    String[] words = new String[RC[2]];
    int[][] Result = new int[RC[2]][4];

    ReadPuzzle(FileName, puzzle, words);

    long StartTime = System.currentTimeMillis();
    int TotalCompare = BruteForceWordPuzzle(puzzle, words, Result);
    long StopTime = System.currentTimeMillis();

    System.out.println("");

    System.out.println("Execution time: " + (StopTime - StartTime) + " ms");

    System.out.print("Total Comparison: " + TotalCompare);
}
}

```

Test Case

A. Small

a. Small 1

Isi File

```
S COU P L E E C D S A N A T S W
E N O M M O C Q R P W B W V P E L
A Y G N I V I L E K M S M O S I D
F D E U E G F C A N I O T O F F N
U A M S X E I X T I D L L E O L U
G Y K P V A P O U S E U H R H N O
X S B E L Q P E R W A T C M Y E R
O M N C Y E S F E E R E S E L A G
B T H I N G L I T R D L V S K T R
Y M H F V C D G C L E Y O S C H E
V U F I B E L M E O U D P I I G C
P R O C E S S E D C R C E C U U V
M M W A R T N N A A A N I E Q O E
O D Z L A D G T W N Y L E F R H N
V I T L L Y L R E D R O P R F T S
E N N Y J Y T R I D V S B E G I N
G N I R A T S M Y N E E D E D H D

ABSOLUTELY
FIGMENT
PEERED
BEGIN
FORCED
PLACE
GROUND
PROCESS
CLEAN
IDEA|
```

Hasil run

```
Masukkan nama file: test_small_1.txt

ABSOLUTELY
- - - - - A - - - - -
- - - - - B - - - - -
- - - - - S - - - - -
- - - - - O - - - - -
- - - - - L - - - - -
- - - - - U - - - - -
- - - - - T - - - - -
- - - - - E - - - - -
- - - - - L - - - - -
- - - - - Y - - - - -
- - - - - 
- - - - - 
- - - - - 
- - - - - 
- - - - - 
- - - - - 
- - - - - 
- - - - - 

FIGMENT
- - - - - 
- - - - - 
- - - - - 
- - - - - 
- - - - - 
- - - - - 
- - - - - 
- - - - - 
- - - - - F - - - - -
- - - - - I - - - - -
- - - - - G - - - - -
- - - - - M - - - - -
- - - - - E - - - - -
- - - - - N - - - - -
- - - - - T - - - - -
- - - - - 
- - - - - 
- - - - - 

PEERED
- - - - - 
- - - - - 
- - - - - 
- - - - - 
- - - - - 
- - - - - 
- - - - - P - - - - -
- - - - - E - - - - -
- - - - -   E - - - - -
- - - - -     R - - - - -
- - - - -       E - - - - -
- - - - -         D - - - - -
- - - - - 
- - - - - 
- - - - - 
- - - - - 
- - - - - 
- - - - - 
- - - - - 
- - - - -
```

BEGIN

B E G I N

FORCED

F O R C E D

PLACE

P L A C E

[illegible]

```
IDEA  
- - - - -  
- - - - -  
- - - - -  
- - - I - - -  
- - - D - - -  
- - - E - - -  
- - - A - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
- - - - -  
Execution time: 153 ms  
Total Comparison: 3746
```

b. small 2

Isi File

RAWXWATERMAFZENGY
MTBESHAPETUNJENAQ
TXONEZISUNNTPILEY
BALLOONENRIILPUMI
CVWBPHYEFPDORDED
SLAFIBSKOPDOEPTSA
ECRSAOARDUIGSMNSI
KWUYFMPBCONCSEEIL
WCNTTNIKOOGEEKZMTP
VONSOEELLDRHTNOMC
KLINYEIUUVYGUOMMC
GONEWLUREGIRLXPAQ
WRGFNOITAZINAGROX
NXEDRHPOTVTVXBRATP
SUA AIETNAELCYIDIY
CEFPTISAPUPPYYGMM
HOQKHMGBADJUSTES

ADJUST
GIRL
BACK
GREAT
PUPPY
BALLOON
HEAD
PURPOSE
BATH
RUNNING
LONGER

Hasil run

```
Masukkan nama file: test_small_2.txt

ADJUST

- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - A D J U S T - -
- - - - -

GIRL

- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - G I R L - - -
- - - - -
- - - - -
- - - - -
- - - - -

BACK

- - - - -
- - - - -
- - - - -
- - B - - - - -
- - A - - - - -
- C - - - - -
K - - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
```

G R E A T

P U P P Y

B A L L O O N

c. small 3

Isi File

F E V E R Y T H I N G N S X E
A E D Y Q E T A L O C O H C L
C T E J L T G V W H A T E S P
T O M B K L A F I E L D D S R
A N A G I R A F F E S E K E U
H L O Q D S S U H B R O K I P
T O R X R E T T T E T E N N L
G O D H E R I W D C R N I O L
R K W M A F E N E U A Y P P M
O E E R M R A F T P E C X E O
U D R O G E R S N O T T O C R
N I E C M E A S T O N E S U F
D Z X K P P A X R E V I R R W

PINK
PONIES
PURPLE
GIRAFFES
ROAMED
FIELD
COTTON
GREW
FROM
GROUND

Hasil run

Masukkan nama file: test_small_3.txt

PINK

KNIP

PONIES

SEINOP

PURPLE

ELPRUP

GIRAFFES

G I R A F F E S

ROAMED

D
E
M
A
O
R

FIELD

FIELD

COTTON

GREW

FROM

GROUND

```
Execution time: 113 ms
Total Comparison: 2512
```

B. Medium

a. Medium 1

Isi File

I U Z K S E S U A C E J P Y D N Y T I L I B A H
R R E H M O F C B O T F N I O A N X I E T Y M G
E E N E V E R E A U D K I S S U D S S D O O M E
T D O N J S L L O M C U E L G D R B E E N O J T
F N I M L P E H O A E S T U D I E S Y Y R E V U
A O Y E O T T T L T H R U Y K T H R O U G H G M
O W E E Z I I N E K A T A P H O T O I C O H G O
R P P Y W D R E T U P M O C H C U M R A G F J M
C H A L L E N G E S P O N O I T A V I R P E D E
I I I A K V E V A H E H T N S E O D T L A M E N
R D P U S I H T W H N V E U I F L E E H I C I T
O R D W C L D L R O W Y A P O N E P G R E Q T H
L I E T H O U S A N D S C R R N D T M N I H M O
L V L J F K H E G K E Q H L O O H U W I A T Z U
X I L D N T Q N L W Z D A U E A B O C T S H F R
I N O U A E I I H F I O G N S A H L G E J G C S
M G R M E Y T E A C L H R F D S R D E E D S K W
A D C H A N N I R B A Y E Z S N E L A M M G A O
G S S T U C N A Q O E Y L T A E R G Y B S N W R
E E S F L A Z W F N R C I C V H T N O D O I A K
S N Q N M Y C A N E R E M E M B E R G Q U H K E
H O S O R T S S E N O I S S E R P E D F C T E J

SLEEP
DEPRIVATION
CAUSES
SORTS
CHALLENGES
AND
PROBLEMS
WHEN
DOESNT
ENOUGH
WORK
CLEARLY
STUDIES
AFTER
STAYING
AWAKE
HOURS
ABILITY

SLEEP

```

- - - - S
- - - L -
- - E - -
- E - - -
P - - - -

```

DEPRIVATION

-NOITAVIRPED

CAUSES

- SESUAC

CHALLENGES

CHALLENGES

AND

- A N D

AND

A
N
D

PROBLEMS

P
R
O
B
L
E
M
S

WHEN

W
H
E
N

DOESNT

T N S E O D

ENOUGH

E
N
O
U
G
H

WORK

CLEARLY

STUDIES

STUDIES

AFTER

RETFA

STAYING

- - - - - C
 - - - - - N
 - - - - - I
 - - - Y - -
 - - A - - -
 - T - - - -
 S - - - - -

AWAKE

- A
- W
- A
- K
- E

HOURS

H
O
U
R
S

[illegible]

b. Medium 2

Isi File

HEBEHTBDTPQSAMETHTEMOH
HFTNQZDEETRIAEAPHLGYIG
UISARQCNEINSKKKZNELQNX
LLAMPINKRNOEMJLZNIYGKM
YSFESENSESHLCASWADGKOY
IAKZDGDWHUWVCSSODVBOHAT
OWAELIHVDIOTJTEDNALWTR
NFEALGTERYHEDTMRGDOLID
SURPRISEOMWVNHGDOKDVAU
ZZBPCKJVRLEETIAEUEIKA
CDELLACEEFRREISYIRLZMJ
ETAUCOIVLEEYRWLRHYUFTO
DNUOFHELFFYDOGHATAAETQIB
TUGDENAFYPANRWDLIEVPMH
EABOUTINSLMEOPRAHSYAEA
JNAYADUHGDDRBTDIBROFDL
QCI EWZQMUEKTHATCEINRNW
KDRTAHSAMERPXSBI VNQTHA
XG IOUIKEDOFVPOUFEHETXY
UBAPHOEGTMCBTVTFRUBTTS
IBWTMSRBATUFMJEOYMJTFQ

DAVE
FOUND
DAILY
ROUTINE
LIFE
AWOKE
SAME
TIME
BREAKFAST
DROVE
COMMUTE
WORKED
THAT
NEVER
SEEMED
CHANGE

Masukkan nama file: test_medium_2.txt

DAVE

E
V
A
D

FOUND

D N U O F

DAILY

Y
L
I
A
D

ROUTINE

E
N
I
T
U
O
R

LIFE

E
F
I
L

AWOKE

A
W
O
K
E

-	T
-	S
-	A
-	F
-	K
-	A
-	E
-	R
-	B

.....

[illegible]

WORKED

W
O
R
K
E
D

THAT

T H A T

NEVER

R
E
V
E
N

SEEMED

D
E
M
E
S

CHANGE

C
H
A
N
G
E

Execution time: 400 ms
Total Comparison: 10075

c. Medium 3

Isi File

Q E N O Y N A F X Z P T L T H E R E M K F T Z
K L H X S Z U O Y U T H O U M I R H D C F S W
E F I T P S O W N H S E I B X C U X D S Y U I
Z Z Y O N Q G T I F R Y J E R P Q F B I X J Z
W N D E T S I S N I I P L V R T E L P O E P Y
A W C D F F F X D Y F S H I U K W Y A S P G F
S F P M N I S O O N H P B F Y M V N T B R R X
J R H F F I N U J J S J H Z Z S T A L K I V I
X S O S T L F E M U Z R N L R T F S J E Y I I
U R R Y C W E E S S U K E S J L C B N A W Z I
H A G A I N Y G N I H T V R S D I D N T W R Q
U E Y T D R E S C F D K E E S Y Z I N K O Z M
P Y E F W E L Y P W R L R G Y K V G L D U P H
E A H M E B C H I L D X K N M B E O T O L E Q
R H T O V M J J F W Y E N A Z Y E E I M D V H
O Y S R R Y E C A R T T E R E H W Y R E V E A
M P Q P I Z D R X I J M W T O Z Q F X O H R C
N D U A T C E M L Y L K Q S N E V E B P H Z H
E G B E E N K O M E O J I T Z V M A X V T F W
V H X S U R P R I S I N G P F C J K Y L M N C

THERE
TRACEY
FIRST
ANYONE
EVER
KNEW
WOULD
CHILD
EVERYWHERE
BEEN
FRIEND
FIVE
YEARS
THING
SURPRISING

B. Large

a. Large 1

Isi File

VELPRUPYUWHJXPMVTHUEVILBPD EATH
YPQAVUNTAWBAASRTTSGNMEANDEREDD
LZEWOIETQWYSDOUZLNSATYYAISPSRB
LYBRASAEASEHTBOQEVTPINECCIUCS
AFGRPKPGEADVENTUREFEHOLREZDGVG
EQYUEDXRRTNAHTUSYDUOLCENAFGLANR
RODKEEAGAIJRRWMRDITCPSNIEASOYO
YTNM BINKLDLOBOUTATYNGXERENIABU
LQAEENIANOJTSNNVRVEDBEQYLHSGLIN
NOCIHPNICCSAWYLLAUTCAVFQI IINED
RHNTNGDPOWXSWEXPLORETODRRHMTJD
TGOIOPLDHRDWREGRET XAELTAOGWMQC
UNHPTRVOC POTENTIALHCIRFHNME LFT
BILQTEKEUDAILYFZSWUKMFOIEHRPIE
QTOSOHTLUCIFFIDQADEUEXNMTRGAEL
DIONCTETLIGHTIDEENPSBEGINZSSL L
ICKHCAALLBSENITUORPFPASHTIWSDI
AXEEDEMGNINRAWSHTVBOKSCVQATEEN
GEDRUWMTPEREHTAESNSALLERBMUDLG
NOITATIVNIZATBPDQEOHREVIRRITPR
OSNSSELJLVURJGOCCVUHIOFFBMQTOC
SLIPPEDAPPROACHEDILEJTFEKEZIEE
ISTOPPEDKLDUNOTRFGLSOMETHINGPR
SARAHEIXATDNFGOTUKWTJKND DAERUT
FEFILUDTWOHDPFNANHMONIL IARTJLA
BPEPSPEIRGWITWRIGHTHATNOCBWCLI
PCHEERRTSALFSDRNKGBLEDIHRBININ
JDDANGERSTESKPLXIPHMS ENOTSMNL
REHTARINILUVSINYSXHIEVERYONEGZ
SIDEXXTWPZDOCTOGEFWXS WOULD RDTH

CLOUDY
OUTSIDE
CERTAINLY
DIFFICULT
EXPLAIN
DIAGNOSIS
SOMETHING
DRUDGERY
ADVENTUROUS
EXCITING
OTHERS
BEGIN
ADVENTURE
REGRET
GIRAFFES
ROAMED
GROUND
CHOCOLATE
RIVER
MEANDERED

Hasil run

```
Masukkan nama file: test_big_1.txt
```

```
CLOUDY
```

```
- - - - - Y D U O L C - - - - -
```

```
OUTSIDE
```

```
E  
D  
I  
S  
T  
U  
O
```

CERTAINLY

C
E
R
T
A
I
N
L
Y

DIFFICULT

T L U C I F F I D

SOMETHING

S O M E T H I N G

DRUDGERY

D
R
U
D
G
E
R
Y

ADVENTUROUS

S
U
O
R
U
T
N
E
V
D
A

EXCITING

G
N
I
T
I
C
X
E

OTHERS

O
T
H
E
R
S

BEGIN

B E G I N

ADVENTURE

Handwriting practice area for the word "ADVENTURE". The area contains 20 rows of dashed lines for tracing and writing. The word "ADVENTURE" is printed in the center of the first row as a guide.

REGRET

Handwriting practice area for the word "REGRET". The area contains 20 rows of dashed lines for tracing and writing. The word "REGRET" is printed in the center of the first row as a guide.

GIRAFFES

G
I
R
A
F
F
E
S

ROAMED

D
E
M
A
O
R

GROUND



G
R
O
U
N
D

CHOCOLATE



E
T
A
L
O
C
O
H
C

ETALOCCHIO

This image shows a full page of graph paper. It features a uniform grid of small squares created by thin, light gray horizontal and vertical lines. The grid covers the entire area of the page, leaving no margins or other markings.

A blank sheet of graph paper with a grid of small squares. The grid consists of 20 columns and 15 rows of squares. The lines are thin and light gray, forming a uniform pattern across the page. There are no margins or other markings on the paper.

b. Large 2

Isi File

UGUMWLWFYWRPPCND EGANAMTMYTGNSROJSEO
FNXJOWQSVGAIHNLIKEDVVHOMDNBYDREDVFE
NIQEOUESGRUJLABIXHMQRUENIUOYQEPBU
KKECIVDAAUTABEVEIHCAPMHJHYATLLRERON
FOZCJGVGLDFWHYACDECROFEIYHJASNEOKUO
MOKRFDRYCDXOJFCUMIAOMHVSBBOSDKVHLES
MLWVUASEEMEDXRXTNAWFTHVPBRMYDUBIEHS
YFQCPZJRHRRTZAYVOWBOUAHCOUPLEYLWVLIZ
JJNHMSGVCBVNTRAUKRVMQEJFIZNWGOAAKC
ZQQSAAWUKOTGYUEDNOYEBQJZRWIVIJKLTDWZ
JSNEVETIEEPYVYUONNYSXS AUVHLSWAMMBZH
HTBOYADNTPUMRILETDTELJOHRLBXMCLYNU
BIUCEXQPIINNEDRRYAQGGKOMYLCFATVJLTHC
TLTNLSVPITHNETVLRQFTNIEYDLRMBDFINXA
YLLAUTCAHKSHCWTTSSOMCOMLOLFBXALVKRNX
NKYXOWGGZLSKYNGAYUMRWMYUYLIDCUYPYXH
NYYILIUDDTBOERXEGEDEVEIECERACKOVOTSC
TSAIDOJIPFWTHXNQAVXAGOFGYHOROWNBFS A
KRPBHMNDNATSREDNUNTYTSSZGKHSIZEENEOME
MRATONEEXIHHEEEQORRAUMEUGHDRWZTATVWR
DAICTROFSXGDISIAEELRNMENIUBAOAIWEES
QSVJKHGNIKAMESWVTIFETXTEKAAITHBLHUR
KWSDNPOCCLEANYREVERFDAYSNWGKBSLCGAZ
KNUGQCZMFOXEFZNIVACXOLF EZUVJSM PGNET
LDISOZZLRJHZEZNETKKIBRBQGLBEENEMLU
EICSNVSNFELOUGJIDRMPNATXOWVAXSIBIRP
JRFKMAWXRUUMODNARMOYAGAKNRNUTMPMIDD
STMAKGSPBGFESGGIIJEDEBJTXIGTOKJDELV
TYTCIGMGWRESCATKNMUWDUATNTSMGEIWRFB
ETYWIOSGOBRSHBWITAFXQHRGSIEOFCHGEMK
HVHICDCGMMAATGOXHPEZTYUFUNMBUEOXHNC
EBOXSQXGXCCHOSBOYWALQVMQTGALNKDUTKP
RRYLREDRONITUMPPGRTHCDJV TYOSGNIHTHP
AWHWPZAYNSMYDYPN IAGAPEYQWUIPQOZKYZX
OKXBAILEDYHGNIHTEMOSWBQXSWUIXUWFUF

HAVENT
GENERATING
EFFORT
WILL
RIDICULOUS
RECEIVED
SURE
BEEN
WELLMEANING
SAID
UNDERSTAND
LITERALLY
KILL
ACTUALLY
MANAGED
ACHIEVE

Hasil Run

HAVENT

H
A
V
E
N
T

GENERATING

G
E
N
E
R
A
T
I
N
G

EFFORT

E
F
F
O
R
T

WILL

W
I
L
L

R
 I
 D
 I
 C
 U
 L
 O
 U
 S



[illegible][illegible]

A large grid of small squares, resembling graph paper or a data table. The grid is composed of many small squares arranged in a regular pattern, with a slightly darker border on the left side.

[illegible]

KILL

L
L
I
K

ACTUALLY

Y L L A U T C A

MANAGED

- D E G A N A M -

ACHIEVE

- E V E I H C A -

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan (no syntax error)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2. Program berhasil running	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3. Program dapat membaca file masukan dan menuliskan luaran	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4. Program berhasil menemukan semua kata di dalam puzzle.	<input checked="" type="checkbox"/>	<input type="checkbox"/>