Laporan

Tugas Kecil Strategi Algoritma

Penggunaan Algoritma Brute Force pada Permainan Word Search



Disusun Oleh: Fransiskus Davin Anwari/13520025

PROGRAM STUDI TEKNIK INFORMATIKA SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA **INSTITUT TEKNOLOGI BANDUNG** 2021

ALGORITMA BRUTE FORCE

Algoritma Brute Force adalah pendekatan yang lempang untuk memecahkan suatu masalah. Algoritma ini biasanya dilakukan dengan melakukan semua tindakan yang dapat dilakukan untuk memecahkan suatu masalah yang diberikan.

Dalam kasus ini kita menggunakan Algoritma Brute Force untuk menyelesaikan suatu puzzle word search. Kita dapat menggunakan perbandingan huruf yang ada pada kata yang dicari dengan karakter pada kumpulan huruf-huruf yang terdapat pada puzzle word search. Hal ini dilakukan berkali-kali sehingga mendapat huruf yang sama pada kata yang dicari dan di puzzle word search, kemudian dilanjutkan dengan melihat huruf selanjutnya pada kata dan pada puzzle. Jika huruf selanjutnya tidak sama maka kita akan membandingkan ulang kembali mulai dari huruf pertama pada kata yang dicari, sehingga kita perlu mendapat perbandingan yang mendapat huruf sama berturut-turut hingga kata yang dicaripun terbentuk.

Program dimulai dengan meminta nama file wordsearch yang ingin diselesaikan, kemudian menampilkan puzzlenya terlebih dahulu secara penuh sebelum menyelesaikannya. Karena pada puzzle word search dapat menemukan kata dari segala arah, maka dibuat kode sendiri untuk masing-masing arah. Dari horizontal kiri ke kanan dan kanan ke kiri, vertical ke atas, vertical ke bawah, diagonal kiri atas ke kanan bawah, diagonal kanan atas ke kiri bawah, diagonal kanan bawah ke kiri atas, dan yang terakhir diagonal kiri bawah ke kanan atas. Program akan melakukan masing-masing tersebut dan hanya berhenti juga semua fungsi sudah dicoba sekali atau kata yang dicari sudah ditemukan.

Program jika telah menemukan kata pada puzzle akan menulis ulang puzzle tersebut menjadi semua character '-' kecuali kata yang telah ditemukan untuk menunjukan letak kata yang dicari pada puzzle tersebut. Selama pencarian kata juga dilakukan perhitungan banyak perbandingan yang dilakukan sehingga mendapatkan kata yang diinginkan. Program kemudian juga menuliskan banyak waktu yang terlewat selama algoritma word search berjalan.

SOURCE CODE WORDSEARCH PADA JAVA

```
k++;
                     banyak++;
                 } else if (k==lenw){
                     banyak++;
                     break;
                 } else {
                     banyak++;
                     break;
            if (k==lenw){
                 System.out.println(W+" Ditemukan");
                 ketemu = true;
                 break;
        if(ketemu){
            break;
    return ketemu;
public static int countwordH(int banyak, String W, char[][] S){
    boolean ketemu = false;
    int lenw = W.length();
    int lens = S[0].length;
    for (int h=0; h<S.length;h++){</pre>
        for (int i=0; i<= lens-lenw;i++){</pre>
            int k = 0;
            for (int j=i; j<lens ; j++){</pre>
                 if (k < lenw && W.charAt(k)==S[h][j]) {</pre>
                     k++;
                     banyak++;
                 } else if (k==lenw){
                     banyak++;
                     break;
                 } else {
                     banyak++;
                     break;
            if (k==lenw){
                 ketemu = true;
                 break;
             }
        if(ketemu){
```

```
break;
    return banyak;
public static char[][] RewritefindwordH(String W, char[][] S){
    boolean ketemu = false;
    char[][] finalm = new char[S.length][S[0].length];
    int lenw = W.length();
    int lens = S[0].length;
    for (int h=0; h<S.length;h++){</pre>
        for (int i=0; i<= lens-lenw;i++){</pre>
            int k = 0;
            for (int j=i; j<lens; j++){
                 if (k < lenw && W.charAt(k)==S[h][j] && !ketemu) {</pre>
                     finalm[h][j] = S[h][j];
                 } else if (k==lenw){
                     finalm[h][j] = '-';
                 } else {
                     finalm[h][j] = '-';
                     break;
            if (k==lenw){
                 ketemu = true;
                 break;
            } else {
                 for(int l=0;l<lens;l++){</pre>
                     finalm[h][l] = '-';
                 }
        if (!ketemu){
            for(int l=0;l<lens;l++){</pre>
                 finalm[h][l] = '-';
    return finalm;
public static boolean findwordDTopLeft(int banyak, String W, char[][] S){
    boolean ketemu = false;
    int lenw = W.length();
    int lens = S[0].length;
    for (int h=0; h<S.length;h++){</pre>
```

```
for (int i=0; i<= lens-lenw;i++){</pre>
                 int k = 0;
                 int tambah = 0;
                 for (int j=i; j<lens ; j++){</pre>
                     if (h+tambah < S.length && k < lenw &&
W.charAt(k)==S[h+tambah][j]) {
                         banyak++;
                         tambah++;
                     } else if (k==lenw){
                         banyak++;
                         break;
                     } else {
                         banyak++;
                         break;
                 if (k==lenw){
                     System.out.println(W+" Ditemukan");
                     ketemu = true;
                     break;
            if(ketemu){
                 break;
        return ketemu;
    public static int countwordDTopLeft(int banyak, String W, char[][] S){
        boolean ketemu = false;
        int lenw = W.length();
        int lens = S[0].length;
        for (int h=0; h<S.length;h++){</pre>
             for (int i=0; i<= lens-lenw;i++){</pre>
                 int k = 0;
                 int tambah = 0;
                 for (int j=i; j<lens ; j++){</pre>
                     if (h+tambah < S.length && k < lenw &&
W.charAt(k)==S[h+tambah][j]) {
                         banyak++;
                         tambah++;
                     } else if (k==lenw){
                         banyak++;
                         break;
                     } else {
```

```
banyak++;
                          break;
                 if (k==lenw){
                     ketemu = true;
                     break;
             if(ketemu){
                 break;
        return banyak;
    public static char[][] RewritefindwordDTopLeft(String W, char[][] S){
        boolean ketemu = false;
        char[][] finalm = new char[S.length][S[0].length];
        int lenw = W.length();
        int lens = S[0].length;
        for (int i1=0;i1<S.length;i1++){</pre>
             for(int j1=0;j1<S[0].length;j1++){</pre>
                 finalm[i1][j1] = '-';
        for (int h=0; h<S.length;h++){</pre>
             for (int i=0; i<= lens-lenw;i++){</pre>
                 int k = 0;
                 int tambah = 0;
                 for (int j=i; j<lens ; j++){</pre>
                      if (h+tambah < S.length && k < lenw &&
W.charAt(k)==S[h+tambah][j] && !ketemu) {
                          finalm[h+tambah][j] = S[h+tambah][j];
                          tambah++;
                     } else{
                          break;
                 if (k==lenw){
                     ketemu = true;
                     break;
                 } else {
                     for (int i1=0;i1<S.length;i1++){</pre>
                          for(int j1=0;j1<S[0].length;j1++){</pre>
                              finalm[i1][j1] = '-';
```

```
if (ketemu){
                break;
        return finalm;
    public static boolean findwordDTopRight(int banyak, String W, char[][] S){
        boolean ketemu = false;
        int lenw = W.length();
        int lens = S[0].length;
        for (int h=0; h<S.length;h++){</pre>
            for (int i=lens-1; i>=lenw-1;i--){
                int k = 0;
                int tambah = 0;
                for (int j=i; j>=0; j--){
                    if (h+tambah < S.length && k < lenw &&
W.charAt(k)==S[h+tambah][j]) {
                        k++;
                        tambah++;
                        banyak++;
                    } else if (k==lenw){
                        banyak++;
                        break;
                    } else {
                        banyak++;
                        break;
                if (k==lenw){
                    System.out.println(W+" Ditemukan");
                    ketemu = true;
                    break;
            if(ketemu){
                break;
        return ketemu;
    public static int countwordDTopRight(int banyak, String W, char[][] S){
        boolean ketemu = false;
        int lenw = W.length();
```

```
int lens = S[0].length;
        for (int h=0; h<S.length;h++){</pre>
             for (int i=lens-1; i>=lenw-1;i--){
                 int k = 0;
                 int tambah = 0;
                 for (int j=i; j>=0; j--){
                     if (h+tambah < S.length && k < lenw &&
W.charAt(k)==S[h+tambah][j]) {
                         tambah++;
                         banyak++;
                     } else if (k==lenw){
                         banyak++;
                         break;
                     } else {
                         banyak++;
                         break;
                 }
                 if (k==lenw){
                     ketemu = true;
                     break;
                 }
            if(ketemu){
                 break;
             }
        return banyak;
    public static char[][] RewritefindwordDTopRight(String W, char[][] S){
        boolean ketemu = false;
        char[][] finalm = new char[S.length][S[0].length];
        int lenw = W.length();
        int lens = S[0].length;
        for (int i1=0;i1<S.length;i1++){</pre>
             for(int j1=0;j1<S[0].length;j1++){</pre>
                 finalm[i1][j1] = '-';
        for (int h=0; h<S.length;h++){</pre>
             for (int i=lens-1; i>=lenw-1;i--){
                 int k = 0;
                 int tambah = 0;
                 for (int j=i; j>=0; j--){
                     if (h+tambah < S.length && k < lenw &&
W.charAt(k)==S[h+tambah][j] && !ketemu) {
```

```
k++;
                     finalm[h+tambah][j] = S[h+tambah][j];
                     tambah++;
                 } else {
                     break;
            if (k==lenw){
                 ketemu = true;
                 break;
                 for (int i1=0;i1<S.length;i1++){</pre>
                     for(int j1=0;j1<S[0].length;j1++){</pre>
                         finalm[i1][j1] = '-';
             }
        if (ketemu){
            break;
        }
    return finalm;
public static boolean findwordDBotLeft(int banyak, String W, char[][] S){
    boolean ketemu = false;
    int lenw = W.length();
    int lens = S[0].length;
    for (int h=S.length-1; h>=0;h--){
        for (int i=0; i<= lens-lenw;i++){</pre>
            int k = 0;
            int tambah = 0;
            for (int j=i; j<lens ; j++){</pre>
                 if (h-tambah >= 0 && k < lenw && W.charAt(k)==S[h-tambah][j])</pre>
                     k++;
                     tambah++;
                     banyak++;
                 } else if (k==lenw){
                     banyak++;
                     break;
                 } else {
                     banyak++;
                     break;
            if (k==lenw){
                 System.out.println(W+" Ditemukan");
```

```
ketemu = true;
                break;
        if(ketemu){
            break;
    return ketemu;
public static int countwordDBotLeft(int banyak, String W, char[][] S){
    boolean ketemu = false;
    int lenw = W.length();
    int lens = S[0].length;
    for (int h=S.length-1; h>=0;h--){
        for (int i=0; i<= lens-lenw;i++){</pre>
            int k = 0;
            int tambah = 0;
            for (int j=i; j<lens ; j++){</pre>
                if (h-tambah >= 0 && k < lenw && W.charAt(k)==S[h-tambah][j])</pre>
                    k++;
                    tambah++;
                     banyak++;
                } else if (k==lenw){
                    banyak++;
                     break;
                } else {
                     banyak++;
                    break;
            if (k==lenw){
                ketemu = true;
                break;
        if(ketemu){
            break;
    return banyak;
public static char[][] RewritefindwordDBotLeft(String W, char[][] S){
    boolean ketemu = false;
    char[][] finalm = new char[S.length][S[0].length];
```

```
int lenw = W.length();
        int lens = S[0].length;
        for (int i1=0;i1<S.length;i1++){</pre>
             for(int j1=0;j1<S[0].length;j1++){
                 finalm[i1][j1] = '-';
        for (int h=S.length-1; h>=0;h--){
            for (int i=0; i<= lens-lenw;i++){</pre>
                 int k = 0;
                 int tambah = 0;
                 for (int j=i; j<lens ; j++){</pre>
                     if (h-tambah >= 0 && k < lenw && W.charAt(k)==S[h-tambah][j]</pre>
&& !ketemu) {
                         k++;
                         finalm[h-tambah][j] = S[h-tambah][j];
                         tambah++;
                     } else{
                         break;
                 if (k==lenw){
                     ketemu = true;
                     break;
                 } else {
                     for (int i1=0;i1<S.length;i1++){</pre>
                         for(int j1=0;j1<S[0].length;j1++){</pre>
                              finalm[i1][j1] = '-';
            if (ketemu){
                 break;
        return finalm;
    public static boolean findwordDBotRight(int banyak, String W, char[][] S){
        boolean ketemu = false;
        int lenw = W.length();
        int lens = S[0].length;
        for (int h=S.length-1; h>=0;h--){
             for (int i=lens-1; i>=lenw-1;i--){
                 int k = 0;
                 int tambah = 0;
                 for (int j=i; j>=0; j--){
```

```
if (h-tambah) = 0 \& k < lenw \& W.charAt(k) == S[h-tambah][j])
                    k++;
                    banyak++;
                    tambah++;
                } else if (k==lenw){
                    banyak++;
                    break;
                } else {
                    banyak++;
                    break;
            }
            if (k==lenw){
                System.out.println(W+" Ditemukan");
                ketemu = true;
                break;
        if(ketemu){
            break;
    return ketemu;
public static int countwordDBotRight(int banyak, String W, char[][] S){
    boolean ketemu = false;
    int lenw = W.length();
    int lens = S[0].length;
    for (int h=S.length-1; h>=0;h--){
        for (int i=lens-1; i>=lenw-1;i--){
            int k = 0;
            int tambah = 0;
            for (int j=i; j>=0; j--){
                if (h-tambah) = 0 \& k < lenw \& W.charAt(k) == S[h-tambah][j])
                    k++;
                    banyak++;
                    tambah++;
                } else if (k==lenw){
                    banyak++;
                    break;
                } else {
                    banyak++;
                    break;
```

```
if (k==lenw){
                     ketemu = true;
                     break;
                 }
            if(ketemu){
                break;
        return banyak;
    public static char[][] RewritefindwordDBotRight(String W, char[][] S){
        boolean ketemu = false;
        char[][] finalm = new char[S.length][S[0].length];
        int lenw = W.length();
        int lens = S[0].length;
        for (int i1=0;i1<S.length;i1++){</pre>
             for(int j1=0;j1<S[0].length;j1++){</pre>
                 finalm[i1][j1] = '-';
        for (int h=S.length-1; h>=0;h--){
            for (int i=lens-1; i>=lenw-1;i--){
                 int k = 0;
                 int tambah = 0;
                 for (int j=i; j>=0; j--){
                     if (h-tambah >= 0 && k < lenw && W.charAt(k)==S[h-tambah][j]</pre>
&& !ketemu) {
                         k++;
                         finalm[h-tambah][j] = S[h-tambah][j];
                         tambah++;
                     } else{
                         break;
                 if (k==lenw){
                     ketemu = true;
                     break;
                 } else {
                     for (int i1=0;i1<S.length;i1++){</pre>
                         for(int j1=0;j1<S[0].length;j1++){</pre>
                              finalm[i1][j1] = '-';
                 }
            if (ketemu){
```

```
break;
    return finalm;
public static boolean findwordV(int banyak, String W, char[][] S){
    boolean ketemu = false;
    int lenw = W.length();
    int lens = S.length;
    for (int h=0; h<S[0].length;h++){</pre>
        for (int i=0; i<= lens-lenw;i++){</pre>
             int k = 0;
             for (int j=i; j<lens ; j++){</pre>
                 if (k < lenw \&\& W.charAt(k)==S[j][h]) {
                     k++;
                     banyak++;
                 } else if (k==lenw){
                     banyak++;
                     break;
                 } else {
                     banyak++;
                     break;
             }
             if (k==lenw){
                 System.out.println(W+" Ditemukan");
                 ketemu = true;
                 break;
        if(ketemu){
             break;
    return ketemu;
public static int countwordV(int banyak, String W, char[][] S){
    boolean ketemu = false;
    int lenw = W.length();
    int lens = S.length;
    for (int h=0; h<S[0].length;h++){</pre>
        for (int i=0; i<= lens-lenw;i++){</pre>
             int k = 0;
             for (int j=i; j<lens ; j++){</pre>
                 if (k < lenw && W.charAt(k)==S[j][h]) {</pre>
                     k++;
                     banyak++;
```

```
} else if (k==lenw){
                     banyak++;
                     break;
                 } else {
                     banyak++;
                     break;
            if (k==lenw){
                 ketemu = true;
                 break;
        if(ketemu){
            break;
    return banyak;
public static char[][] RewritefindwordV(String W, char[][] S){
    boolean ketemu = false;
    char[][] finalm = new char[S.length][S[0].length];
    int lenw = W.length();
    int lens = S.length;
    for (int h=0; h<S[0].length;h++){</pre>
        for (int i=0; i<= lens-lenw;i++){</pre>
             int k = 0;
             for (int j=i; j<lens ; j++){</pre>
                 if (k < lenw && W.charAt(k)==S[j][h] && !ketemu) {</pre>
                     finalm[j][h] = S[j][h];
                 } else if (k==lenw){
                     finalm[j][h] = '-';
                 } else {
                     finalm[j][h] = '-';
                     break;
            if (k==lenw){
                 ketemu = true;
                 break;
             } else {
                 for(int l=0;l<lens;l++){</pre>
                     finalm[1][h] = '-';
                 }
```

```
if (!ketemu){
            for(int l=0;1<lens;1++){
                finalm[1][h] = '-';
    return finalm;
public static boolean findwordVReverse(int banyak, String W, char[][] S){
    boolean ketemu = false;
    int lenw = W.length();
    int lens = S.length;
    for (int h=0; h<S[0].length;h++){</pre>
        for (int i=lens-1; i>=lenw-1;i--){
            int k = 0;
            for (int j=i; j>=0 ; j--){
                if (k < lenw && W.charAt(k)==S[j][h]) {</pre>
                    k++;
                    banyak++;
                } else if (k==lenw){
                    banyak++;
                    break;
                } else {
                    banyak++;
                    break;
            if (k==lenw){
                System.out.println(W+" Ditemukan");
                ketemu = true;
                break;
        if(ketemu){
            break;
    return ketemu;
public static int countwordVReverse(int banyak, String W, char[][] S){
    boolean ketemu = false;
    int lenw = W.length();
    int lens = S.length;
    for (int h=0; h<S[0].length;h++){</pre>
        for (int i=lens-1; i>=lenw-1;i--){
            int k = 0;
```

```
for (int j=i; j>=0; j--){
                 if (k < lenw && W.charAt(k)==S[j][h]) {</pre>
                     banyak++;
                 } else if (k==lenw){
                     banyak++;
                     break;
                 } else {
                     banyak++;
                     break;
            if (k==lenw){
                ketemu = true;
                break;
        if(ketemu){
            break;
    return banyak;
public static char[][] RewritefindwordVReverse(String W, char[][] S){
    boolean ketemu = false;
    char[][] finalm = new char[S.length][S[0].length];
    int lenw = W.length();
    int lens = S.length;
    for (int h=0; h<S[0].length;h++){</pre>
        for (int i=lens-1; i>=lenw-1;i--){
            int k = 0;
            for (int j=i; j>=0; j--){
                 if (k < lenw && W.charAt(k)==S[j][h] && !ketemu) {</pre>
                     k++;
                     finalm[j][h] = S[j][h];
                 } else if (k==lenw){
                     finalm[j][h] = '-';
                 } else {
                     finalm[j][h] = '-';
                     break;
            }
            if (k==lenw){
                ketemu = true;
                break;
            } else {
                for(int l=0;l<lens;l++){</pre>
```

```
finalm[1][h] = '-';
        if (!ketemu){
            for(int l=0;l<lens;l++){</pre>
                finalm[1][h] = '-';
    return finalm;
public static boolean findwordHReverse(int banyak, String W, char[][] S){
    boolean ketemu = false;
    int lenw = W.length();
    int lens = S[0].length;
    for (int h=0; h<S.length;h++){</pre>
        for (int i=lens-1; i>=lenw-1;i--){
            int k = 0;
            for (int j=i; j>=0; j--){
                if (k < lenw && W.charAt(k)==S[h][j]) {</pre>
                    banyak++;
                    k++;
                } else if (k==lenw){
                    banyak++;
                    break;
                } else {
                    banyak++;
                    break;
            if (k==lenw){
                System.out.println(W+" Ditemukan");
                ketemu = true;
                break;
        if(ketemu){
            break;
    return ketemu;
public static int countwordHReverse(int banyak, String W, char[][] S){
    boolean ketemu = false;
    int lenw = W.length();
```

```
int lens = S[0].length;
    for (int h=0; h<S.length;h++){</pre>
        for (int i=lens-1; i>=lenw-1;i--){
            int k = 0;
            for (int j=i; j>=0; j--){
                if (k < lenw && W.charAt(k)==S[h][j]) {</pre>
                     banyak++;
                     k++;
                } else if (k==lenw){
                     banyak++;
                     break;
                } else {
                     banyak++;
                     break;
            if (k==lenw){
                ketemu = true;
                break;
            }
        if(ketemu){
            break;
    return banyak;
public static char[][] RewritefindwordHReverse(String W, char[][] S){
    boolean ketemu = false;
    char[][] finalm = new char[S.length][S[0].length];
    int lenw = W.length();
    int lens = S[0].length;
    for (int h=0; h<S.length;h++){</pre>
        for (int i=lens-1; i>=lenw-1;i--){
            int k = 0;
            for (int j=i; j>=0; j--){
                if (k < lenw && W.charAt(k)==S[h][j] && !ketemu) {</pre>
                     finalm[h][j] = S[h][j];
                } else if (k==lenw){
                     finalm[h][j] = '-';
                } else {
                     finalm[h][j] = '-';
                     break;
            if (k==lenw){
```

```
ketemu = true;
                     break;
                 } else {
                     for(int l=0;l<lens;l++){</pre>
                         finalm[h][l] = '-';
            if (!ketemu){
                 for(int l=0;l<lens;l++){</pre>
                     finalm[h][l] = '-';
        return finalm;
    public static void tulisMatriks(char[][] m) {
        int i,j;
        int brs = m.length;
        int kol = m[0].length;
        for(i=0;i<brs;++i)</pre>
            for(j=0;j<kol;++j)</pre>
                System.out.print(m[i][j]+" ");
            System.out.println();
    public static char[][] ReadFile(String filePath) throws IOException {
        Scanner scnr = new Scanner(new BufferedReader(new
FileReader("../test/"+filePath+".txt")));
        char[][] mat = new char[99][99];
        int j=0;
        int k=0;
        while(scnr.hasNextLine()){
            String line = scnr.nextLine();
            if(line==""){
                break;
            String noSpaceStr = line.replaceAll("\\s", "");
            char[] line1 = noSpaceStr.toCharArray();
            mat[j] = line1;
            j++;
            k = line1.length;
```

```
char[][] finmat = new char[j][k];
        for (int i=0;i<j;i++){
            for (int i1=0;i1<k;i1++){
                finmat[i][i1]=mat[i][i1];
        scnr.close();
        return finmat;
    public static String[] ReadFileString(String filePath) throws IOException {
        Scanner scnr = new Scanner(new BufferedReader(new
FileReader("../test/"+filePath+".txt")));
        while(scnr.hasNextLine()){
            String line = scnr.nextLine();
            if(line==""){
                break;
        String[] Words = new String[99];
        int i = 0;
        while(scnr.hasNextLine()){
            String line = scnr.nextLine();
            String noSpaceStr = line.replaceAll("\\s", "");
            Words[i] = noSpaceStr;
            i++;
        scnr.close();
        String[] finals = new String[i];
        for(int j=0;j<i;j++){</pre>
            finals[j]=Words[j];
        return finals;
    public static void main(String[] args) throws Exception {
        Scanner input = new Scanner(System.in);
        System.out.print("Nama file:");
        String filename = input.next();
        String[] Words = ReadFileString(filename);
        char[][] puzzle = ReadFile(filename);
        tulisMatriks(puzzle);
        long Start = System.nanoTime();
        for(int i=0;i<Words.length;i++){</pre>
            int banyak = 0;
            boolean found = findwordH(banyak, Words[i],puzzle);
            banyak += countwordH(banyak, Words[i],puzzle);
            if (!found){
```

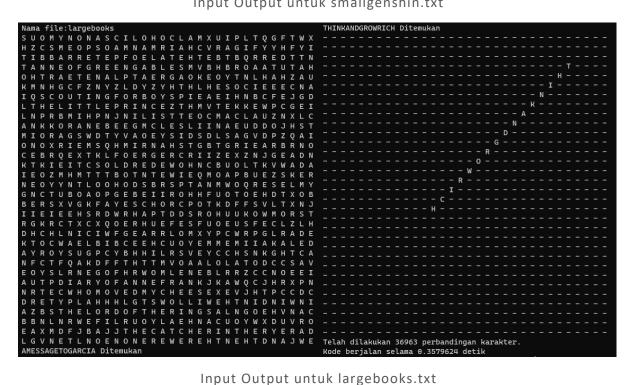
```
boolean foundR = findwordHReverse(banyak, Words[i],puzzle);
                banyak += countwordHReverse(banyak, Words[i], puzzle);
                if (foundR){
                    tulisMatriks(RewritefindwordHReverse(Words[i],puzzle));
                    System.out.println("Telah dilakukan "+banyak+" perbandingan
karakter.");
                } else{
                    boolean foundV = findwordV(banyak, Words[i], puzzle);
                    banyak += countwordV(banyak, Words[i], puzzle);
                    if (foundV){
                        tulisMatriks(RewritefindwordV(Words[i], puzzle));
                        System.out.println("Telah dilakukan "+banyak+"
perbandingan karakter.");
                    } else {
                        boolean foundvR = findwordVReverse(banyak, Words[i],
puzzle);
                        banyak += countwordVReverse(banyak, Words[i], puzzle);
                        if (foundvR){
                            tulisMatriks(RewritefindwordVReverse(Words[i],
puzzle));
                            System.out.println("Telah dilakukan "+banyak+"
perbandingan karakter.");
                        } else {
                            boolean foundDTL = findwordDTopLeft(banyak, Words[i],
puzzle);
                            banyak += countwordDTopLeft(banyak, Words[i], puzzle);
                            if (foundDTL){
                                tulisMatriks(RewritefindwordDTopLeft(Words[i],
puzzle));
                                System.out.println("Telah dilakukan "+banyak+"
perbandingan karakter.");
                            } else {
                                boolean foundDTR = findwordDTopRight(banyak,
Words[i], puzzle);
                                banyak += countwordDTopLeft(banyak, Words[i],
puzzle);
                                if (foundDTR){
                                    tulisMatriks(RewritefindwordDTopRight(Words[i]
, puzzle));
                                    System.out.println("Telah dilakukan "+banyak+"
perbandingan karakter.");
                                } else {
                                    boolean foundDBL = findwordDBotLeft(banyak,
Words[i], puzzle);
                                    banyak += countwordDBotLeft(banyak, Words[i],
puzzle);
                                    if (foundDBL){
```

```
tulisMatriks(RewritefindwordDBotLeft(Words
[i], puzzle));
                                        System.out.println("Telah dilakukan
"+banyak+" perbandingan karakter.");
                                    } else {
                                        boolean foundDBR =
findwordDBotRight(banyak, Words[i], puzzle);
                                        banyak += countwordDBotRight(banyak,
Words[i], puzzle);
                                        if (foundDBR){
                                            tulisMatriks(RewritefindwordDBotRight(
Words[i], puzzle));
                                            System.out.println("Telah dilakukan
"+banyak+" perbandingan karakter.");
            if (found){
                tulisMatriks(RewritefindwordH(Words[i], puzzle));
                System.out.println("Telah dilakukan "+banyak+" perbandingan
karakter.");
        long end = System.nanoTime();
        long Total = end-Start;
        double TotalTime = (double) Total / 1_000_000_000;
        input.close();
        System.out.println("Kode berjalan selama "+TotalTime+" detik");
```

SCREENSHOT INPUT DAN OUTPUT

Nama file:smallgenshin	Telah dilakukan 91 perbandingan karakter.
	lisa Ditemukan
ICHILDEUNYZY	LISA DICEMURAN
REBMAJEONNHR	
BXEBAEKDXNOO	
EQINUALIIUNZ	
CGIAINEEAHGA	
HATBNNEBOZLR	
ONNUEGGABIIA	
NYEEDNLGEAUB	
GUVAUINIUBGR	
YUQIQILENANA	
ULISAANUTGNB	- L I S A
NANOELLECTRG	
FISCHLKAEYAU	
TBXINGQUILAA	
AMBER Ditemukan	Telah dilakukan 101 perbandingan karakter.
	RAZOR Ditemukan
R E B M A	
	R
	0
	z
	A
	R
Tolah dilakukan 266 pembandingan kanaktan	Telah dilakukan 1746 perbandingan karakter.
Telah dilakukan 266 perbandingan karakter.	Kode berjalan selama 0.0701193 detik

Input Output untuk smallgenshin.txt



Nama filarmadiumanaganias	TOMATOES Ditemukan
Nama file:mediumgroceries	TOMATOES DICEMURAN
WCHIPSBASITORTILASQYF	
ECIUJDQREGRUBEIGGEVDQ	
TSWBROCCLLISREMAETSNL	
SLAHGGSUYSAACPEASHWAS	
AVTLJFFHGFOWUHAZZIPCH	
PLECMORGALAUGHINGCOWR	
HZAGTOEEBMNROSCCKGTII	
TYCFGDNQPPPDEOELKONAM	
OEPAUIISAPYONRIPIEVGP	
OKJCRLEPNTEDOMULAENRN	
TRHUQRESAOTPTEEBHRESR	
RUHPFROETIPIDTSSMNGEO	
UTDSTLMTORLMPNAENAVAC	
GMROHYONSAAAAWAIEFHSP	
0 Y W J C S E U P T P W Y T D T W H Z O O	
YLKMARTINEUDSVPWLBCNP	
SBSUNDAURDONTDKBAAUSZ	
BXZGIAJFNBESAODKRTSNF	T O M A T O E S
VTREPLEHREGRUBMAHEEQS	Telah dilakukan 312 perbandingan karakter.
SJNXSMWOAZCVSWNQMSARP	TOOTHPASTE Ditemukan
TOMATOESQOFCUCUMBERDE	
BODYWASH Ditemukan	E
BODYWASH DICEMURAN	T
	S
	A
	P
	H
	T
	0
	0
H	
A	
W	
D	
0	Telah dilakukan 3820 perbandingan karakter.
B	Kode berjalan selama 0.1238584 detik

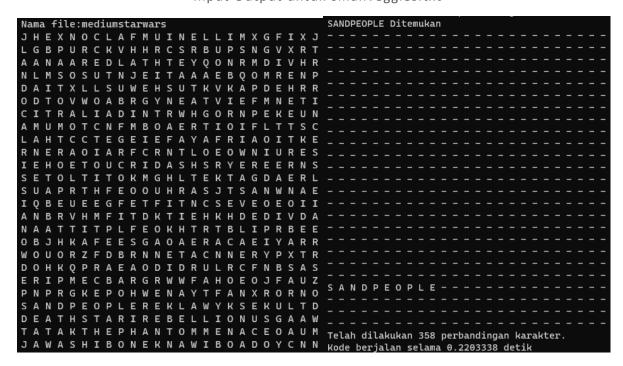
Input Output untuk mediumgroceries.txt

Nama file:smallaction	PLANOF Ditemukan
RCXIEVISAVESEUA	
V R D K R Y P F A L O V W O N	
0 V C E E F O E S R U O C N B	
SGVQYLNQGCILBAR	
VEJMXWALMNINAIQ	
LNTPOLICEVIFBPF	
RIKUOKWOIBFKPJE	
DOWJOWZCAIAMADU	
ZEJGHAPPRGRZVRS	P
POLQAPGMQPUORGB	L
VOTALVAGTGCVFQL	A
CUYAYTXQCMANOFA	N
OQNZIEZKPQTRCVG	- 0
VOWVAKDEIWMOBJE	F
FXEWRYRALLIPACL	Telah dilakukan 9986 perbandingan karakter
AFFIRMATIVE Ditemukan	POLICE Ditemukan
A	
F	POLICE
F	
I	
R	
M	
A	
T	
I	
V	
E	Telah dilakukan 61 perbandingan karakter.
Telah dilakukan 5089 perbandingan karakter.	Kode berjalan selama 0.0614744 detik

Input Output untuk smallaction.txt

Nama file:smallveggies	PEAS Ditemukan
CZRJHAYGMBOJCF	
AZRBOSAEPHSTYO	SAEP
UHVHTEMXJPYNNL	
LBORACSFFYDAMX	
IIQUTFUIUSNLFR	
FONIONLMBELPWO	
LVGBPSQLZGYGBT	
OREGNIFYDALGDA	
WJXEHESLZYYEOM	
EDVNIKPMUPWGDO	
RUSSRIKAOOLVCT	
PPTVJIJYKJPHXJ	
CORNECZNISUICL	
LTSKGAGXWOVKWD	Toloh dilabukan 205 nombandingan kamaktan
CAULIFLOWER Ditemukan	Telah dilakukan 345 perbandingan karakter. TOMATO Ditemukan
C	TOMATO DICEMBRATI
A	
n II	
1	
T	
5	0
1	T
0	A
W	M
F	0
E	T
R	
Talah dilalulua 260 sashandiana	Telah dilakukan 1948 perbandingan karakter.
Telah dilakukan 362 perbandingan karakter.	Kode berjalan selama 0.0382485 detik

Input Output untuk smallveggies.txt



Input Output untuk mediumstarwars.txt

Nama file:mediumjunkfood	SWEETS Ditemukan
KRVPIHSIBCLRAGCSYBTSND	
OOVSEIRFGCNERFOEDXYBCT	
SBPRCWYDDBGEKAHSKLIMPH	
GEUEEDLUXRTUFOWRVGJBGV	
XLFCVSXDXHIOFZUMEDFYTA	
XWYAXUIETUXCJYDZKGLJKE	
M D N R S L Q S Y C Y R O E N H J N R E A T	
DZTIECEDTIHJUNTNPEZUVA	
BOICWJEJSFWJVTNYOQJNBL	
J U A S Q G W N F S V Q M P D P U Z L M B O	
IFBFDEYODTUNODGHTPAQOC	
PSNAIDDGDOUCHIPSIVVRLO	
YUORRQNACWGSPIHCNROCFH	
YJDHJSAVOJJODPJTEQWLNC	
ZOBECZCFOUFBDLYDONLYES	
BKZTUAXTKXQPQTEKLXQRYW	E
AWBTUKNCIKVEILOZRWGKJE	
J R N B T X V F E U T Q B B S H M I R C A E	s
PRUZPXPGSBNYAOCOPEZCTT	Telah dilakukan 2489 perbandingan karakter.
NDGNIRCICECREAMEJBVMYS	Kode berjalan selama 0.1061134 detik

Input Output untuk mediumjunkfood.txt

Nama file:largemovie	THEABYSS Ditemukan
SLQYUFNAEBBIRACEHTFOSETARIPEH	
G R E A S E I L L I T T L E M I S S S U N S H I N E Y A U	
TTSEZXADAKNAMREDIPSGNIZAMAEHT	
WHRQTQCIDIAUTHEINCREDIBLEHULK	
IUUJBSBLNLRVYERGFOSEDAHSYTFIF	
LCIMRRFHGRETEDAORYRUFXAMDAMPS	
IRAIEEIOIUARSNDNIWEHTHTIWENOG	
GDERTWCTNYANOEGDISLJOHNCARTER	
H Y X C N R Q H D A X R F N R E Q N O I T P E C N I X B C	
TCOIUALOETMSDOTRROTANIMRETEHT	
HTDFHLKRPBHSTISHESTOKZEXZUHTT	
EHUIRLFTELUETVAEETANTJANBDRIH	
LESCEEPHNSINDHPNLRAGEHMFWPQTE	
O L G A E T G E D S B K S A E L S C O R E C E G Q S Z A T	
N O O P D S N D E U I R R H R H E O I O T O I W R R O N E	
ERD X ERO ANN N A E EU KUS FN FX FFOL JIN	
RDSOHEKRCYPDFCTDKNMTOFEUEOJCC	
A O A M T T G K E P O O S J U T I N G I H R V E L L D J O	
N F N L W N N W D U H T E R F T O R I E S E H X H T A S M	
GTDMMIIOALENIGFBIPTGREGCJTRMM	
EHKSVVKRYPRILOORABYYHGRAEWTOA	
R E I P A F O L Y F C K E D S K A N B R D T A A L H T E N	
J R N I G G H D C I U E U Z Y Z N T A O R A R M B A T G D	
	S S Y B A E H T
FIGXZEZEJCLRRIAGZKAMHANIELXXM	
YNSSYBAEHTETTLDBABOVNEHCSSEYE	
YGQOBZOXYISRRLNTSICHAOHRIEUSN	
ASRAWRATSOTAWAESRBVUFDRTYNSWT	
J J U A K N D K R N X T L S M O U T O F A F R I C A G V S	Telah dilakukan 1984 perbandingan karakter.
INDIANAJONESIIXAWONESPYLACOPA	Kode berjalan selama 0.3230375 detik

Input Output untuk largemovie.txt

Kode Program

Google Drive: https://drive.google.com/drive/folders/1PcsefwNA0VhCllplHy-mwx3c0Nm7zuiU?usp=sharing

Cara menjalankan Program dapat dibaca pada README.md

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalaha (nosyntax error)	✓	
2. Program berhasil running	✓	
3. Program dapat membaca file masukan dan menuliskan luaran	✓	
4. Program berhasil menemukan semua kata di dalam puzzle	✓	