

Kasus :

Anda sebagai seorang programmer mendapatkan project untuk mengerjakan sebuah game sederhana. Game ini dimainkan oleh satu pemain melawan komputer. Adapun alur permainan sebagai berikut :

1. Pada tampilan awal game, maka akan muncul menu sebagai berikut :
  - a. Generate Kata
  - b. Simpan hasil Generate Kata
  - c. Mulai Permainan
  - d. Tampilkan nama pemenang
2. Menu Generate Kata berfungsi untuk menghasilkan 16 kata secara random, masing masing kata terdiri dari 5 huruf. String yang di generate hanya terdiri dari huruf a-z dan A-Z. Dimana huruf besar dan huruf kecil akan dianggap berbeda
3. Menu Simpan Hasil Generate Kata : 16 kata yang telah di generate akan disimpan dalam file dengan nama data.txt. Apabila ternyata pemain tidak suka dengan 16 kata tersebut, maka pemain dapat melakukan generate kata ulang.
4. Menu Mulai Permainan : pada menu ini, maka permainan akan dimulai dengan alur berikut :
  - a. Game akan membaca isi data.txt yang telah dibuat pada menu sebelumnya. Apabila file tersebut ternyata tidak ada isinya, maka game akan memberikan informasi bahwa permainan tidak dapat diteruskan karena tidak ada kata yang bisa di load
  - b. Setelah game membaca string yang terdapat dalam file data.txt, maka game akan menampilkan kata kata tersebut ke dalam 16 kotak yang telah diberikan nomor. Contoh tampilan sebagai berikut

|       |       |       |       |
|-------|-------|-------|-------|
| Arxio | sjksV | Srshx | sthek |
| dserx | siwKV | srxjg | poseT |
| szler | deioP | tjdKG | sjvTS |
| ptTks | Pestk | sbvkK | JskRT |

Apakah Anda Sudah Siap Bermain [Y/T] ?

- c. Apabila pemain sudah siap untuk memulai maka layar akan dibersihkan, dan anda diminta untuk memasukkan nama pemain  
Masukkan Nama Pemain : Reichan
- d. Setelah nama pemain dimasukkan, maka game akan menampilkan 16 kotak, yang terdiri dari angka 1 – 16

|    |    |    |    |
|----|----|----|----|
| 1  | 2  | 3  | 4  |
| 5  | 6  | 7  | 8  |
| 9  | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |

- e. Game akan menampilkan sebuah string dari 16 string yang ada secara acak. Contoh tampilan sebagai berikut :  
Game dimulai.....  
Dikotak manakah tulisan deioP berada ?  
Giliran Reichan menebak kotak :

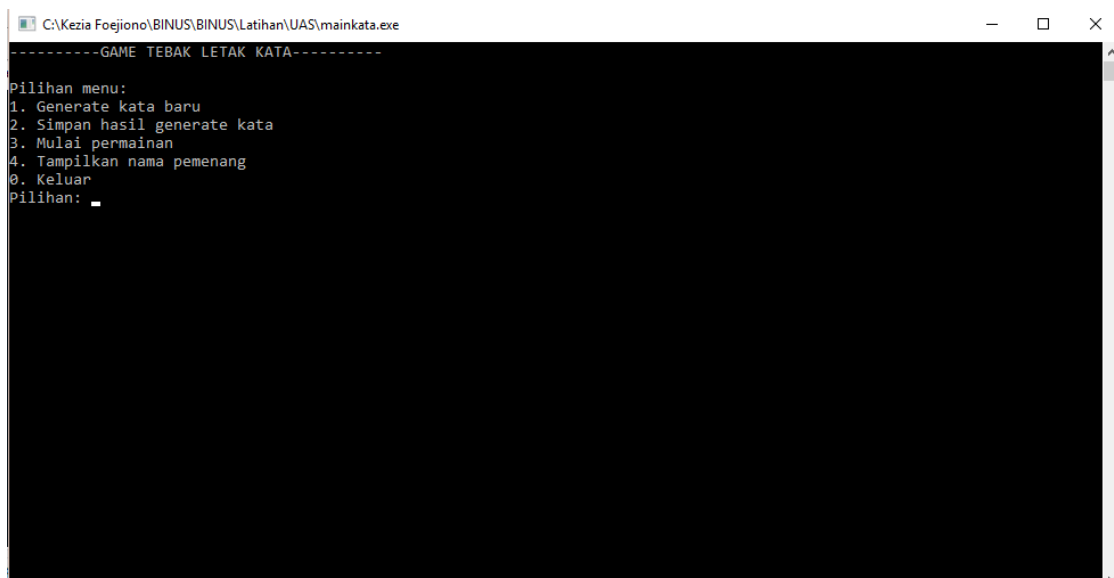
- f. Apabila ternyata tebakan pemain benar , maka nilai 10 akan diberikan kepada pemain. Setelah itu game akan mengulang langkah e dengan string lainnya. String yang sudah berhasil ditebak, tidak akan dimunculkan Kembali
  - g. Apabila pemain menjawab salah, maka giliran berikutnya adalah komputer yang akan menebak. Komputer menebak dengan cara melakukan random angka 1 – 16. Angka yang sudah ditebak pemain sebelumnya, tidak akan dikeluarkan oleh komputer.
  - h. Pemenang adalah pemain dengan nilai tertinggi.
  - i. Setelah permainan selesai, maka akan ditanyakan apakah score akan disimpan. Jika akan disimpan, maka game akan menyimpannya ke dalam file score.txt, yang berisi nama pemain dan score yang didapatkan
5. Menu Tampilkan Pemenang : Menu ini akan menampilkan nama nama pemenang beserta dengan score yang didapatkan. Apabila komputer yang menang maka nama **Kompie** yang akan disimpan. Tampilan akan diurutkan dari score tertinggi ke score terendah.

Program yang anda buat harus terdiri dari :

1. Modul / Fungsi :
  - a. Modul utama
  - b. Modul Generate Kata
  - c. Modul Simpan hasil Generate Kata
  - d. Modul Mulai Permainan
  - e. Modul Tampilkan nama pemenang, didalamnya terdapat sorting
2. Modul lainnya dapat anda tambahkan sesuai dengan kreatifitas anda.
3. Tampilan papan harus terdiri dari 16 kotak.
4. File :
  - a. Data.txt
  - b. Score.txt
5. Anda dapat memodifikasi games agar terlihat lebih menarik.

## PENJELASAN

### Tampilan dan Menu



```
C:\Kezia Foejiono\BINUS\BINUS\Latihan\UAS\mainkata.exe
-----GAME TEBAK LETAK KATA-----
Pilihan menu:
1. Generate kata baru
2. Simpan hasil generate kata
3. Mulai permainan
4. Tampilkan nama pemenang
0. Keluar
Pilihan: _
```

Keterangan: Gambar di atas adalah tampilan awal (menu utama) dari game tebak letak kata. Terdapat 5 pilihan menu yaitu:

1. Generate kata baru: untuk mengacak kata baru
2. Simpan hasil generate kata: untuk menyimpan hasil generate kata (menu no 1) ke dalam file data.txt
3. Mulai permainan: untuk memulai permainan
4. Tampilkan nama pemenang: untuk menampilkan hasil sorting pemenang dari skor terbesar ke terkecil dari file score.txt
0. Keluar: program akan berhenti berjalan dan ditutup bila pemain



```
C:\Kezia Foejiono\BINUS\BINUS\Latihan\UAS\mainkata.exe
-----GAME TEBAK LETAK KATA-----
Pilihan menu:
1. Generate kata baru
2. Simpan hasil generate kata
3. Mulai permainan
4. Tampilkan nama pemenang
0. Keluar
Pilihan: 5
Pilihan yang dimasukkan salah! Silahkan pilih nomor 0-4!
Press any key to continue . . .
```

Keterangan: Gambar diatas adalah tampilan program ketika pemain menginput angka selain 0-4. Program akan memberi peringatan untuk memilih angka 0-4 dan membawa pemain kembali ke menu utama.

```
C:\Kezia Foejiono\BINUS\BINUS\Latihan\UAS\mainkata.exe
-----GAME TEBAK LETAK KATA-----
Pilihan menu:
1. Generate kata baru
2. Simpan hasil generate kata
3. Mulai permainan
4. Tampilkan nama pemenang
0. Keluar
Pilihan: 1
-----
|hRCB1|DVgy1|wGbuS|BMb0r|
|XTlhc|sMpxo|hGmGN|kEuFd|
|XOtOg|bgXpE|yaNFe|tCuKe|
|PZSHK|lJuGg|Gekjd|qzJeN|
-----
Press any key to continue . . .
```

Keterangan: Gambar diatas adalah tampilan program ketika pemain menginput angka 1. Program akan menampilkan 16 string berbeda dalam sebuah tabel. Setelah mengeluarkan ke-16 string tersebut, program akan membawa pemain kembali ke menu utama.

```
C:\Kezia Foejiono\BINUS\BINUS\Latihan\UAS\mainkata.exe
-----GAME TEBAK LETAK KATA-----
Pilihan menu:
1. Generate kata baru
2. Simpan hasil generate kata
3. Mulai permainan
4. Tampilkan nama pemenang
0. Keluar
Pilihan: 2
-----
|hRCB1|DVgy1|wGbuS|BMb0r|
|XTlhc|sMpxo|hGmGN|kEuFd|
|XOtOg|bgXpE|yaNFe|tCuKe|
|PZSHK|lJuGg|Gekjd|qzJeN|
-----
Ingin menggunakan kata-kata diatas?
1. Ya
2. Tidak
Pilihan: _
```

Keterangan: Gambar diatas adalah tampilan program ketika pemain memasukkan angka 2. Program akan menampilkan 16 string yang telah dibentuk di menu 1 dan menanyakan pemain apakah akan menggunakan ke-16 string tersebut atau tidak.

```
C:\Kezia Foejiono\BINUS\BINUS\Latihan\UAS\mainkata.exe
-----GAME TEBAK LETAK KATA-----
Pilihan menu:
1. Generate kata baru
2. Simpan hasil generate kata
3. Mulai permainan
4. Tampilkan nama pemenang
0. Keluar
Pilihan: 2
-----
|hRCB1|DVgyl|wGbuS|BMbOr|
|XTlhc|sMpxo|hGmGN|kEuFd|
|XOtOg|bgXpE|yaNFe|tCuke|
|PZSHK|lJuGg|Gekjd|qzJeN|
-----
Ingin menggunakan kata-kata diatas?
1. Ya
2. Tidak
Pilihan: 1

Data berhasil disimpan. Untuk memulai permainan silahkan tekan enter lalu pilih opsi 3
Press any key to continue . . .
```

Keterangan: Berikut adalah tampilan program ketika pemain memilih nomor 1 (ya). Program akan menyimpan 16 kata yang telah ditampilkan ke dalam file data.txt. Setelah tampilan ini, program akan membawa pemain kembali ke menu utama.

```
C:\Kezia Foejiono\BINUS\BINUS\Latihan\UAS\mainkata.exe
-----GAME TEBAK LETAK KATA-----
Pilihan menu:
1. Generate kata baru
2. Simpan hasil generate kata
3. Mulai permainan
4. Tampilkan nama pemenang
0. Keluar
Pilihan: 2
-----
|hRCB1|DVgyl|wGbuS|BMbOr|
|XTlhc|sMpxo|hGmGN|kEuFd|
|XOtOg|bgXpE|yaNFe|tCuke|
|PZSHK|lJuGg|Gekjd|qzJeN|
-----
Ingin menggunakan kata-kata diatas?
1. Ya
2. Tidak
Pilihan: 2
Silahkan tekan enter dan pilih opsi 1 untuk mengacak ulang kata
Press any key to continue . . .
```

Keterangan: Gambar diatas adalah tampilan program ketika memilih nomor 2 (tidak). Program akan memberi instruksi untuk menekan enter dan memilih menu 1 untuk melakukan generate kata baru. Setelah ini, program akan membawa pemain kembali ke menu utama.

```
C:\Kezia Foejiono\BINUS\BINUS\Latihan\UAS\mainkata.exe
-----GAME TEBAK LETAK KATA-----
Pilihan menu:
1. Generate kata baru
2. Simpan hasil generate kata
3. Mulai permainan
4. Tampilkan nama pemenang
0. Keluar
Pilihan: 3
-----
|hRCB1|DVgy1|wGbuS|BMbOr|
|xTlhc|sMpxo|hGmGN|kEuFd|
|XOt0g|bgXpE|yaNFe|tCuke|
|PZSHK|lJuGg|Gekjd|qzJeN|
-----
Mulai main?
1. Ya
2. Tidak
Pilihan: _
```

Keterangan: Gambar diatas adalah tampilan program ketika pemain memasukkan menu 3. Program akan menampilkan 16 string yang telah disimpan di data.txt sebelumnya. Program juga akan menanyakan pemain untuk memulai permainan atau tidak.

```
C:\Kezia Foejiono\BINUS\BINUS\Latihan\UAS\mainkata.exe
-----GAME TEBAK LETAK KATA-----
Pilihan menu:
1. Generate kata baru
2. Simpan hasil generate kata
3. Mulai permainan
4. Tampilkan nama pemenang
0. Keluar
Pilihan: 3
-----
|hRCB1|DVgy1|wGbuS|BMbOr|
|xTlhc|sMpxo|hGmGN|kEuFd|
|XOt0g|bgXpE|yaNFe|tCuke|
|PZSHK|lJuGg|Gekjd|qzJeN|
-----
Mulai main?
1. Ya
2. Tidak
Pilihan: 1
Press any key to continue . . . _
```

```
C:\Kezia Foejiono\BINUS\BINUS\Latihan\UAS\mainkata.exe
Masukkan nama pemain: Kezia
-----
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |
-----
Game mulai ...
Di kotak manakah tulisan lJuGg berada?
Giliran Kezia menebak kotak: _
```

Gambar diatas adalah tampilan program ketika pemain memilih memulai permainan. Program akan meminta pemain menekan enter. Selanjutnya program meminta input berupa nama pemain. Selanjutnya program akan menampilkan tabel yang terdiri dari 16 angka. Program juga akan menampilkan suatu string acak yang merupakan salah satu dari ke-16 string yang telah ditampilkan sebelumnya. Program akan meminta input user berupa angka 1-16.

```
C:\Kezia Foejiono\BINUS\BINUS\Latihan\UAS\mainkata.exe
-----
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |
-----
Game mulai ...
Di kotak manakah tulisan BMbOr berada?
Giliran Kezia menebak kotak: 4
Benar!
Press any key to continue . . .
```

Keterangan: Gambar diatas adalah tampilan program ketika pemain memasukkan jawaban yang benar. Selanjutnya program akan menampilkan string lain untuk ditebak letaknya.

```
C:\Kezia Foejiono\BINUS\BINUS\Latihan\UAS\mainkata.exe
Masukkan nama pemain: Kezia
+---+
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |
+---+
Game mulai ...
Di kotak manakah tulisan lJuGg berada?
Giliran Kezia menebak kotak: 1
Salah!
Giliran Kompie menebak kotak: 15
Salah!
Press any key to continue . . .
```

Keterangan: Gambar diatas adalah tampilan program setelah pemain menginput jawabannya. Jika jawaban yang diberikan pemain salah maka program akan gantian menebak letak string tersebut.

```
C:\Kezia Foejiono\BINUS\BINUS\Latihan\UAS\mainkata.exe
Score Kezia: 160
Score Kompie: 0
Data pemenang berhasil disimpan
Press any key to continue . . .
```

Keterangan: Gambar diatas adalah tampilan program ketika ke-16 string berhasil ditebak. Program akan menampilkan skor pemain dan skor komputer. Skor tertinggi akan disimpan ke dalam file score.txt.

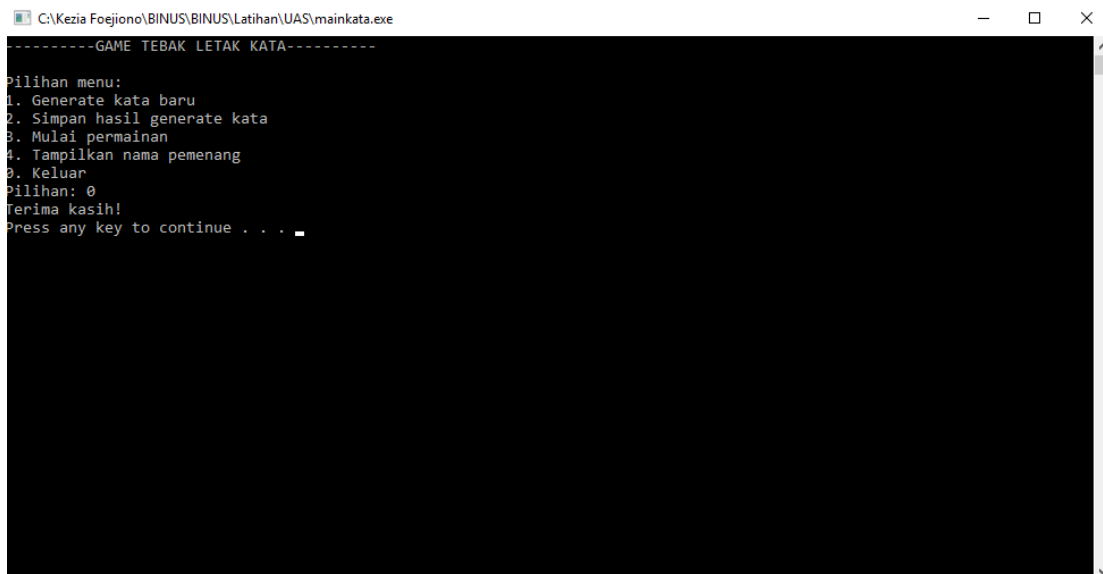


```
C:\Kezia Foejiono\BINUS\BINUS\Latihan\UAS\mainkata.exe
-----GAME TEBAK LETAK KATA-----
Pilihan menu:
1. Generate kata baru
2. Simpan hasil generate kata
3. Mulai permainan
4. Tampilkan nama pemenang
0. Keluar
Pilihan: 3
-----
|hRCB1|DVgy1|wGbuS|BMbOr|
|xTlhc|sMpxo|hGmGN|kEuFd|
|XOtOg|bgXpE|yaNFe|tCuKe|
|PZSHK|lJuGg|Gekjd|qzJeN|
-----
Mulai main?
1. Ya
2. Tidak
Pilihan: 2
Kembali ke menu utama
Press any key to continue . . .
```

Keterangan: Gambar diatas adalah tampilan program ketika pemain memutuskan untuk tidak memulai permainan. Selanjutnya pemain akan diarahkan kembali ke menu utama.

```
C:\Kezia Foejiono\BINUS\BINUS\Latihan\UAS\mainkata.exe
-----GAME TEBAK LETAK KATA-----
Pilihan menu:
1. Generate kata baru
2. Simpan hasil generate kata
3. Mulai permainan
4. Tampilkan nama pemenang
0. Keluar
Pilihan: 4
Rank:
(R). Nama-Score
1. Kezia-160
2. nadya-140
3. mei-120
4. yuni-110
5. Pinie-100
6. ian-100
7. Kompie-90
8. Nana-80
9. Puput-80
Press any key to continue . . .
```

Keterangan: Gambar diatas adalah tampilan program ketika pemain memilih menu 4 yaitu menu untuk melihat skor pemenang. Program akan menampilkan hasil sorting dari file score.txt. Data yang ditampilkan telah di sorting berdasarkan nilai terbesar ke nilai terkecil.



```
C:\Kezia Foejiono\BINUS\BINUS\Latihan\UAS\mainkata.exe
-----GAME  TEBAK  LETAK  KATA-----
Pilihan menu:
1. Generate kata baru
2. Simpan hasil generate kata
3. Mulai permainan
4. Tampilkan nama pemenang
0. Keluar
Pilihan: 0
Terima kasih!
Press any key to continue . . .
```

Keterangan: Gambar diatas adalah tampilan program ketika pemain memilih menu 0 yaitu keluar dari program. Program akan berhenti berjalan dan tertutup.

## CODE YANG MENUNJUKKAN MEMBACA FILE, MENYIMPAN FILE DAN MENUTUP FILE

### 1. CODE MENULIS FILE

```
void simpan() {

    int i,pilih;

    printf("-----\n");

    for(i=1;i<=80;i++) {

        if(i%5==1) {

            printf("|");

        }

        if (i==21||i==41||i==61||i==81) {

            printf("\n|");

        }

        printf("%c",kata[i]);

    }

    printf("|\n");

    printf("-----\n");

    printf("Ingin menggunakan kata-kata diatas?\n");

    printf("1. Ya\n");

    printf("2. Tidak\n");

    printf("Pilihan: ");
```

```

scanf("%d",&pilih);
switch(pilih){
    case 1:
        FILE *fp;
        fp=fopen("data.txt","w");
        for(i=1;i<=80;i++){
            if(i%5==1){
                fprintf(fp,"|");
            }
            if(i==21||i==41||i==61||i==81){
                fprintf(fp,"\n|");
            }
            fprintf(fp,"%c",kata[i]);
        }
        fprintf(fp,"|\n");
        fclose(fp);

        printf("\nData berhasil disimpan. Untuk memulai
permainan silahkan tekan enter lalu pilih opsi 3\n");

        break;

        case 2: printf("Silahkan tekan enter dan pilih opsi 1
untuk mengacak ulang kata\n"); break;

        default: printf("Pilihan salah! Silahkan pilih ulang di
opsi 2\n"); break;
    }
}

```

Code berwarna ungu diatas menunjukkan kode untuk membuka file

**fp=fopen("data.txt","w");** kode w digunakan untuk menginstruksikan program untuk menulis/menyimpan sebuah file baru. Untuk menyimpan ke dalam file digunakan **fprintf** dan di akhir code diberi **fclose(fp);** yang berguna untuk menutup file yang telah dibuka sebelumnya.

#### CODE MEMBACA FILE

```

void permainan(){
    char tampilhuruf[100];

```

```

int i;

FILE *fp;

fp=fopen("data.txt","r");

if(fp==NULL){

    printf("Data tidak ditemukan. Silahkan generate kata
ulang\n"); exit(1);

}

printf("-----\n");

while(fgets(tampilhuruf,100,fp)){

    printf("%s",tampilhuruf);

}

printf("-----\n");

fclose(fp);

FILE *fb;

fb=fopen("data.txt","r");

while(fscanf(fp,"%[^|]|%[^|]|%[^|]|%[^|]\n",&mainkata[i],&ma
inkata[i+1],&mainkata[i+2],&mainkata[i+3])!=EOF){

    i=i+4;

}

fclose(fb);

....

}

```

Code diatas adalah contoh code yang membaca sebuah file dan menampilkan datanya ke display program. Pertama dilakukan pembukaan file dengan code `fp=fopen("data.txt","r");` dan `fb=fopen("data.txt","r");` kode r digunakan untuk menginstruksikan program membaca sebuah file. Selanjutnya untuk membaca isi file dapat menggunakan `fgets` dan `fscanf` Setelah membaca isi file, digunakan `fclose(fp);` dan `fclose(fb);` untuk menutup file yang digunakan sebelumnya.

#### CODE YANG MENAMPILKAN SORTING DAN PENJELASAN METODE SORTING YANG DIGUNAKAN

```

void swap(struct dataskorpemain *a, struct dataskorpemain *b){

    struct dataskorpemain temp;

    strcpy(temp.nama, a->nama);

    temp.score=a->score;

```

```

        strcpy(a->nama, b->nama);

        a->score=b->score;

        strcpy(b->nama, temp.nama);

        b->score=temp.score;

    }

void sort(struct dataskorpemain skor[], int i){
    int j,k,low;
    for(j=0;j<i-1;j++){
        low=j;
        for(k=j+1;k<i;k++){
            if(skor[low].score<skor[k].score) low=k;
            if(low>j) swap(&skor[j],&skor[low]);
        }
    }
}

```

Code diatas adalah code yang menunjukkan sorting. Awalnya data dalam file score.txt akan dibaca dan dimasukkan ke dalam array struct **skor[i].nama** dan **skor[i].score**. Selanjutnya digunakan function untuk melakukan sorting. Pada program ini saya menggunakan metode selection sort. Saya menggunakan selection sort karena code nya mudah dimengerti dan mudah digunakan. Pertama, function akan melakukan looping **for (j=0;j<i-1;j++)** untuk mencari nilai terkecil terlebih dahulu dalam interval j=0 selama j<i-1. Nilai tersebut akan dimasukkan ke dalam variabel low. Lalu dilakukan looping **for (k=j+1;k<i;k++)** untuk membandingkan skor yang satu dengan skor yang lainnya. Jika didapatkan keadaan **skor[low].score<skor[k].score** maka nilai variabel low akan diganti dengan k. Selain itu jika ditemukan keadaan **low>j** maka akan dilakukan pertukaran data dari struct **skor[j]** ke struct **skor[low]**. Pertukaran data akan dilakukan dengan bantuan function swap. Pertama dibuat sebuah struct penampung sementara. Data dari struct a akan disimpan ke struct penampung. Data dari struct b akan disalin ke struct a, dan terakhir data di struct penampung akan disalin ke struct b. Sehingga hasil akhir akan didapatkan pertukaran data dari struct a ke struct b.

## CODE YANG MENUNJUKKAN SEARCHING DAN PENJELASAN METODE SEARCHING

```

void permainan() {
    ....
    index=rand()%16;

    while (ada[index]==1) {

```

```

        index=rand()%16;

    }

    printf("Di kotak manakah tulisan %s
berada?\n",mainkata[index]);

    printf("Giliran %s menebak kotak: ",pemain.nama);

    scanf("%d",&tebak);

    ....

else{

    printf("Salah!\n");

    printf("Giliran Kompie menebak kotak: ");

    tebakkomp=rand()%16+1;

    while (tebakkomp==tebak) {

        tebakkomp=rand()%16+1;

    }

    printf("%d\n",tebakkomp);

    ....

}

```

Code diatas adalah potongan code yang menunjukkan searching dalam program. Pada while pertama dilakukan searching dengan metode linear (satu-satu) dan pencocokan berdasarkan index acak yang telah diketahui. Bila ditemukan bahwa "status" kata tersebut bervalue 1 maka program akan merandom angka ulang sampai "status" kata yang ditemukan bervalue 0 (jika value 1 maka kata tidak bisa digunakan sedangkan bila valuenya 0 maka kata tersebut masih bisa digunakan). Selain itu, di while kedua juga digunakan sebagai searching dan pencocokan. Bila angka random dalam variabel tebakkomp sama dengan value variabel tebak, maka perintah while akan dijakankan. Program akan merandom angka ulang sampai value tebakkomp berbeda dengan value tebak.

## 5 FUNGSI DALAM PROGRAM DAN PENJELASANNYA

2. Modul utama:

CODE

```

295 int main(){
296     int pilih;
297     do{
298         printf("-----GAME TEBAK LETAK KATA-----\n\n");
299         printf("Pilihan menu:\n");
300         printf("1. Generate kata baru\n");
301         printf("2. Simpan hasil generate kata\n");
302         printf("3. Mulai permainan\n");
303         printf("4. Tampilkan nama pemenang\n");
304         printf("0. Keluar\n");
305         printf("Pilihan: ");
306         scanf("%d",&pilih);
307         switch(pilih){
308             case 1: generate(); break;
309             case 2: simpan(); break;
310             case 3: permainan(); break;
311             case 4: pemenang(); break;
312             case 0: keluar(); break;
313             default: printf("Pilihan yang dimasukkan salah! Silahkan pilih nomor 0-4!\n"); break;
314         }
315         system("pause");
316         system("cls");
317     }while(pilih!=0);
318     return 0;
319 }
320
321

```

Code diatas adalah code yang menampilkan modul menu utama. Digunakan do-while untuk menjalankan menampilkan menu utama karena menu utama setidaknya akan ditampilkan 1 kali. Perulangan do-while akan selesai ketika pemain menginputkan angka 0 yang artinya program selesai melakukan looping. Ada beberapa function lain yang digunakan untuk membantu jalannya program. Ada switch case selection yang digunakan untuk menu. Jika pemain menginputkan 1 maka case 1 yaitu function generate() akan dijalankan. Jika pemain menginputkan 2 maka case 2 yaitu function simpan() akan dijalankan. Jika pemain menginputkan 3 maka case 3 yaitu function permainan() akan dijalankan. Jika pemain menginputkan 4 maka case 4 yaitu function pemenang() akan dijalankan. Dan jika pemain menginputkan 0 maka case 0 yaitu function keluar() akan dijalankan dan diakhiri dengan menutup program. Bila pemain menginput angka selain 0-4 maka program akan memberi peringatan untuk menginput angka 0-4 dan meminta input kembali.

Modul generate kata:

CODE

```

13
14 char huruf(int angka){
15     char a;
16     switch(angka){
17         case 0: a='A'; break;
18         case 1: a='B'; break;
19         case 2: a='C'; break;
20         case 3: a='D'; break;
21         case 4: a='E'; break;
22         case 5: a='F'; break;
23         case 6: a='G'; break;
24         case 7: a='H'; break;
25         case 8: a='I'; break;
26         case 9: a='J'; break;
27         case 10: a='K'; break;
28         case 11: a='L'; break;
29         case 12: a='M'; break;
30         case 13: a='N'; break;
31         case 14: a='O'; break;
32         case 15: a='P'; break;
33         case 16: a='Q'; break;
34         case 17: a='R'; break;

```

```

35     case 18: a='S'; break;
36     case 19: a='T'; break;
37     case 20: a='U'; break;
38     case 21: a='V'; break;
39     case 22: a='W'; break;
40     case 23: a='X'; break;
41     case 24: a='Y'; break;
42     case 25: a='Z'; break;
43     case 26: a='a'; break;
44     case 27: a='b'; break;
45     case 28: a='c'; break;
46     case 29: a='d'; break;
47     case 30: a='e'; break;
48     case 31: a='f'; break;
49     case 32: a='g'; break;
50     case 33: a='h'; break;
51     case 34: a='i'; break;
52     case 35: a='j'; break;
53     case 36: a='k'; break;
54     case 37: a='l'; break;
55     case 38: a='m'; break;

```

```

56     case 39: a='n'; break;
57     case 40: a='o'; break;
58     case 41: a='p'; break;
59     case 42: a='q'; break;
60     case 43: a='r'; break;
61     case 44: a='s'; break;
62     case 45: a='t'; break;
63     case 46: a='u'; break;
64     case 47: a='v'; break;
65     case 48: a='w'; break;
66     case 49: a='x'; break;
67     case 50: a='y'; break;
68     case 51: a='z'; break;
69 }
70 return a;
71 }
72
73 void acakhuruf(char kata[]){
74     int i;
75     for(i=1;i<=80;i++){
76         kata[i]=huruf(rand()%52);

```

```

77     }
78 }
79
80 void generate(){
81     int i;
82     acakhuruf(kata);
83     printf("-----\n");
84     for(i=1;i<=80;i++){
85         if(i%5==1){
86             printf("|");
87         }
88         if(i==21||i==41||i==61||i==81){
89             printf("\n");
90         }
91         printf("%c",kata[i]);
92     }
93     printf("\n");
94     printf("-----\n");
95 }

```

Code diatas adalah code yang menampilkan modul generate kata dan function-function pendukungnya. Yang dilakukan pertama kali dalam modul generate adalah menjalankan modul acakhuruf. Modul acakhuruf digunakan untuk mencari angka random dari 0-51 (karena ada 52 karakter yang diperbolehkan) dan menggunakan bantuan function huruf untuk mencocokkan angka dan mereturn huruf. Hasil return value dari function huruf akan disimpan ke dalam variabel global array kata[i]. Function huruf akan digunakan sebanyak 80 kali, sesuai dengan jumlah huruf yang diperlukan untuk mendapatkan 16 string dengan 5 huruf di setiap stringnya. Setelah didapatkan 80 huruf acak dan disimpan ke dalam array kata[i], modul generate akan mencetak 80 huruf tersebut dan membaginya menjadi 16 string dalam suatu tabel. Sehingga hasil tampilannya akan berupa tabel dengan 16 tempat dimana tiap tempat akan diisi oleh 5 huruf.

Modul simpan hasil generate kata:

CODE



```

97 void simpan(){
98     int i pilih;
99     printf("-----\n");
100     for(i=1;i<=80;i++){
101         if(i%5==1){
102             printf("|");
103         }
104         if(i==21||i==41||i==61||i==81){
105             printf("\n|");
106         }
107         printf("%c",kata[i]);
108     }
109     printf("\n");
110     printf("-----\n");
111     printf("Ingin menggunakan kata-kata diatas?\n");
112     printf("1. Ya\n");
113     printf("2. Tidak\n");
114     printf("Pilihan: ");
115     scanf("%d",&pilih);
116     switch(pilih){
117         case 1:
118             FILE *fp;
119             fp=fopen("data.txt","w");
120             for(i=1;i<=80;i++){
121                 if(i%5==1){
122                     fprintf(fp,"|");
123                 }
124                 if(i==21||i==41||i==61||i==81){
125                     fprintf(fp,"\n|");
126                 }
127                 fprintf(fp,"%c",kata[i]);
128             }
129             fprintf(fp,"\n");
130             fclose(fp);
131             printf("\nData berhasil disimpan. Untuk memulai permainan silahkan tekan enter lalu pilih opsi 3\n");
132             break;
133             case 2: printf("Silahkan tekan enter dan pilih opsi 1 untuk mengacak ulang kata\n"); break;
134             default: printf("Pilihan salah! Silahkan pilih ulang di opsi 2\n"); break;
135         }
136     }
137 }

```

Code diatas adalah code yang menampilkan modul simpan hasil generate kata. Pertama modul akan menampilkan kata yang akan disimpan dalam bentuk tabel (tampilannya sama dengan yang ditampilkan dalam modul generate kata sebelumnya). Selanjutnya program akan meminta persetujuan pemain untuk menyimpan kata-kata tersebut ke dalam file data.txt. Bila pemain memilih opsi 1 maka kata-kata yang ditampilkan di dalam tabel akan disimpan ke dalam file data.txt, sedangkan bila pemain memilih opsi 2 maka program tidak akan menyimpan kata-kata ke dalam file. Namun kata-kata tersebut tidak langsung dihapus dalam program. Pemain harus menginput opsi 1 pada menu utama agar huruf yang disimpan dalam program berubah menjadi huruf-huruf baru.

Modul untuk memulai permainan:

CODE

```

138 int cek(){
139     int i;
140     for(i=0;i<16;i++){
141         if(ada[i]==0) return 1;
142     }
143     return -1;
144 }
145
146 void permainan(){
147     char tampilhuruf[100];
148     int i;
149     FILE *fp;
150     fp=fopen("data.txt","r");
151     if(fp==NULL){
152         printf("Data tidak ditemukan. Silahkan generate kata ulang\n"); exit(1);
153     }
154     printf("-----\n");
155     while(fgets(tampilhuruf,100,fp)){
156         printf("%s",tampilhuruf);
157     }
158     printf("-----\n");

```

```

159     fclose(fp);
160     FILE *fb;
161     fb=fopen("data.txt","r");
162     while(fscanf(fp,"%[^|]|%[^|]|%[^|]|%[^|]\n",&mainkata[i],&mainkata[i+1],&mainkata[i+2],&mainkata[i+3])!=EOF){
163         i=i+4;
164     }
165     fclose(fb);
166     int mulai;
167     printf("Mulai main?\n");
168     printf("1. Ya\n");
169     printf("2. Tidak\n");
170     printf("Pilihan: ");
171     scanf("%d",&mulai);
172     while(mulai!=1&&mulai!=2){
173         printf("Input salah! Silahkan masukkan angka 1 atau 2!\n");
174         printf("Pilihan: ");
175         scanf("%d",&mulai);
176     }
177     switch(mulai){
178     case 1:{
179         system("pause");

```

```

180         system("cls");
181         struct dataskorpemain pemain;
182         printf("Masukkan nama pemain: "); fflush(stdin); gets(pemain.nama);
183         int i,index,tebak,tebakkomp,scorekomputer=0;
184         for(i=0;i<16;i++){
185             ada[i]=0;
186         }
187         do{
188             printf("-----\n");
189             printf("| 1 | 2 | 3 | 4 | \n");
190             printf("| 5 | 6 | 7 | 8 | \n");
191             printf("| 9 | 10 | 11 | 12 | \n");
192             printf("| 13 | 14 | 15 | 16 | \n");
193             printf("-----\n");
194             printf("Game mulai ... \n");
195             index=rand()%16;
196             while(ada[index]==1){
197                 index=rand()%16;
198             }
199             printf("Di kotak manakah tulisan %s berada?\n",mainkata[index]);
200             printf("Giliran %s menebak kotak: ",pemain.nama);

```

```

201             scanf("%d",&tebak);
202             while(tebak<1&&tebak>16){
203                 printf("Input salah! Masukkan angka 1-16.\n");
204                 printf("Giliran %s menebak kotak: ",pemain.nama);
205                 scanf("%d",&tebak);
206             }
207             if(tebak-1==index){
208                 printf("Benar!\n");
209                 ada[index]=1;
210                 pemain.score=pemain.score+10;
211             }
212             else{
213                 printf("Salah!\n");
214                 printf("Giliran Kompie menebak kotak: ");
215                 tebakkomp=rand()%16+1;
216                 while(tebakkomp==tebak){
217                     tebakkomp=rand()%16+1;
218                 }
219                 printf("%d\n",tebakkomp);
220                 if(tebakkomp-1==index){
221                     printf("Benar!\n");

```

```

222                     ada[index]=1;
223                     scorekomputer=scorekomputer+10;
224                 }
225                 else{
226                     printf("Salah!\n");
227                 }
228             }
229             system("pause");
230             system("cls");
231         } while(cek()!=-1);
232         printf("Score %s: %d\n",pemain.nama,pemain.score);
233         printf("Score Kompie: %d\n",scorekomputer);
234         FILE *fp;
235         fp=fopen("score.txt","a");
236         if(pemain.score>scorekomputer){
237             fprintf(fp,"%s %d\n",pemain.nama,pemain.score);
238         }
239         else{
240             fprintf(fp,"Kompie %d\n",scorekomputer);
241         }
242         fclose(fp);

```

```

243         printf("Data pemenang berhasil disimpan\n");
244     } break;
245     case 2: printf("Kembali ke menu utama\n"); break;
246 }
247 }
248

```

Code diatas adalah code yang menunjukkan modul mulai permainan dan function pendukungnya. Pertama modul akan membuka file data.txt dan membaca isi serta menampilkannya ke dalam layar (tampilannya berupa tabel yang sama dengan tabel dalam modul generate dan modul simpan). selanjutnya program akan menutup file. Program akan kembali membuka data.txt dan menyimpan semua kata-kata yang ada di dalam file ke array mainkata[i]. Program akan meminta persetujuan pemain untuk memulai permainan. Jika pemain tidak setuju, program akan kembali ke menu uti,a. Jika pemain setuju maka program akan meminta pemain menginput namanya dan memulai permainan. Program akan memberi "status" kepada setiap kata. Angka 0 berarti kata masih bisa digunakan, sedangkan angka 1 berarti kata tidak bisa digunakan. Program akan menampilkan tabel berisi angka 1-16 dan menampilkan sebuah kata acak yang didapat dari merandom angka. Program akan meminta input jawaban pemain. Bila jawaban pemain benar maka program akan menambah 10 poin ke poin pemain dan memberu kata baru. Jika pemain salah menebak maka kompie akan mulai merandom angka. Bila hasil random angka kompie benar maka score kompie akan bertambah 10, jika tidak maka giliran menebak akan diberikan kepada pemain. Hal ini akan terus dilakukan sampai semua "status" kata bernilai 1 setelah itu program akan menampilkan skor akhir pemain dan kompie. Skor terbesar akan disimpan ke dalam file score.txt.

Modul tampilkan nama pemenang:

#### CODE

```

249 void swap(struct dataskorpemain *a, struct dataskorpemain *b){
250     struct dataskorpemain temp;
251     strcpy(temp.nama, a->nama);
252     temp.score=a->score;
253     strcpy(a->nama, b->nama);
254     a->score=b->score;
255     strcpy(b->nama, temp.nama);
256     b->score=temp.score;
257 }
258
259 void sort(struct dataskorpemain skor[], int i){
260     int j,k,low;
261     for(j=0;j<i-1;j++){
262         low=j;
263         for(k=j+1;k<i;k++){
264             if(skor[low].score>skor[k].score) low=k;
265             if(low<j) swap(&skor[j],&skor[low]);
266         }
267     }
268 }
269
270 void print(struct dataskorpemain skor[], int i){
271     int j;
272     printf("Rank:\n");
273     printf("(R). Nama-Score\n");
274     for(j=0;j<i;j++){
275         printf("%d. %-15s %d\n",j+1,skor[j].nama,skor[j].score);
276     }
277 }
278 void pemenang(){
279     struct dataskorpemain skor[101];
280     int i;
281     FILE *fp;
282     fp=fopen("score.txt","r");
283     while(fscanf(fp,"%s %d\n",&skor[i].nama,&skor[i].score)!=EOF){
284         i++;
285     }
286     fclose(fp);
287     sort(skor,i);
288     print(skor,i);
289 }
290

```

Code diatas adalah code yang menunjukkan modul tempilkan nama pemenang beserta function pendukungnya. Pertama modul pemenang akan membaca data nama dan skor dari file score.txt. Data tersebut akan disimpan ke dalam struct array. Selanjutn file ditutup. Selanjutnya dilakukan sorting data dengan bantuan modul sort. Dalam modul sort digunakan metode selection sort. Akan

dicari data terkecil dan dimasukkan ke dalam variabel low. lalu dilakukan nested looping untuk membandingkan skor. Bila didapatkan keadaan  $\text{skor}[\text{low}].\text{score} < \text{skor}[\text{k}].\text{score}$  maka nilai variabel low akan diganti k. Bila didapatkan keadaan  $\text{low} > \text{j}$  maka akan dilakukan pertukaran data yang dibantu oleh modul swap. Modul swap akan membuat tempat penampungan. Data dalam struct a akan disalin ke dalam struct tempung. Data dalam struct b akan disalin ke struct a. Dan data dalam struct tempung akan disalin ke dalam struct b. Metode sorting ini dilakukan sampai semua data terurut dari terbesar ke terkecil.