

LAPORAN PENYELESAIAN PROJECT BIG CHALLENGE

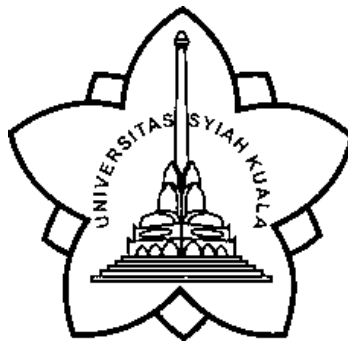
disusun untuk memenuhi
UAS Mata Kuliah Pemrograman

Oleh:

Kelompok 4 :

Mulia Andiki (2308107010013)

Dian Nazira (2308107010011)



**JURUSAN INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SYIAH KUALA
DARUSSALAM, BANDA ACEH
2024**

Permasalahan

Sebuah program dalam bentuk modular perlu dibuat untuk membaca data dari sebuah file teks, mengubah karakter kata menjadi huruf besar diawal kata (title-case), melakukan skoring kata berdasarkan metode skoring yang ditentukan dan disepakati dengan dosen pengasuh, menghimpun kata-kata dengan skor yang sama urut berdasarkan abjad, dan mengurutkan skor kata dari yang paling besar ke yang paling kecil (descending order). Hasil pengurutan itu kemudian disimpan dalam sebuah file binari dengan susunan sebagai berikut:

```
[skor_terbesar][jumlah_kata_dengan_skor_tersebut][panjang_kata][kata]      [panjang_kata]
[kata][panjang_kata][kata] ... [skor_kedua_terbesar] [jumlah_kata_dengan_skor_tersebut]
[panjang_kata][kata][panjang_kata][kata] [panjang_kata][kata] ... [skor_ketiga_terbesar]
[jumlah_kata_dengan_skor_tersebut]          [panjang_kata][kata][panjang_kata][kata]
[panjang_kata][kata] ... dstnya sampai dengan skor terkecil
```

Dimana:

[skor]	bertipe double
[jumlah_kata]	bertipe integer
[panjang_kata]	bertipe integer
[kata]	bertipe string atau array of char

Program ini harus dapat menampilkan maksimum n kata dengan nilai skor terbesar ke layar monitor dan juga untuk menampilkan kata dalam skor yang sama menurut abjad sesuai dengan format dari metode skoring.

Metode skoring :

Metode skoring yang kami gunakan adalah metode skoring Braille. Metode skoring Braille dikembangkan oleh Louis Braille, seorang pria prancis yang buta saat umur tiga tahun akibat kecelakaan. Metode Braille adalah sistem penulisan dan pembacaan yang dirancang untuk digunakan oleh orang buta atau memiliki gangguan penglihatan. Sistem ini memungkinkan individu untuk membaca teks dan menulis menggunakan serangkaian titik-titik timbul yang diatur dalam pola khusus. Sistem Braille sangat penting dalam mendukung literasi dan kemandirian bagi individu yang mengalami kebutaan atau gangguan penglihatan.

Metode Braille

a ●○ ○○ ○○	b ●○ ●○ ○○	c ●● ○○ ○○	d ●● ●○ ○○	e ●○ ●○ ○○	f ●● ●○ ○○	g ●● ●● ○○	h ●○ ●● ○○	i ○● ●○ ○○	j ○● ●● ○○
k ●○ ○○ ●○	l ●○ ●○ ●○	m ●● ○○ ●○	n ●● ○○ ●○	o ●○ ○○ ●○	p ●● ●○ ●○	q ●● ●● ●○	r ●○ ●● ●○	s ○● ●○ ●○	t ○○ ●● ●○
u ●○ ○○ ●●	v ●○ ●○ ●●	w ○● ●● ●○	x ●● ●○ ●○	y ●● ○○ ●○	z ●○ ○○ ●●				

Skema skoringnya mengacu pada letak posisi yang dihitamkan

1	2
3	4
5	6

Jika posisi 1 dihitamkan, maka huruf tersebut bernilai 1

Jika posisi 2 dihitamkan, maka huruf tersebut bernilai 2

Jika posisi 5 dihitamkan, maka huruf tersebut bernilai 5

Jika posisi 1 dan 2 dihitamkan, maka huruf tersebut bernilai $1 + 2 = 3$

Hasil

A. Tampilan Output

1. Untuk menjalankan program ini, sebelum harus dcompile terlebih dahulu menggunakan syntax yang sudah tersedia dalam file makeFile yang syntaxnya adalah `gcc -o kelompok4 main.c bubbleSort.c bacaBin.c garis.c hurufAngka.c kata.c panjang.c pisahkata.c skor.c tulisBin.c hitungHuruf.c` dan juga `.\kelompok4 WIN`, seperti pada gambar dibawah ini .

```
PS D:\SEMESTER 2\MK WAJIB\PEMROGRAMAN\BIG CHALLENGE> gcc -o kelompok4 main.c bubbleSort.c bacaBin.c garis.c hurufAngka.c kata.c panjang.c pisahkata.c skor.c tulisBin.c hitungHuruf.c linuxorwind.c
PS D:\SEMESTER 2\MK WAJIB\PEMROGRAMAN\BIG CHALLENGE> .\kelompok4 WIN
```

Gambar 1. Syntax untuk mengcompile program

2. Setelah selesai, program akan menampilkan kata Selamat Datang Di Big Challenge, Sebelum Memulai, Silahkan input nama file. Dibagian ini akan menampilkan 2 pilihan yaitu input file teks dan exit. Berikut ini tampilan dari hasil compile.

```
===== Selamat Datang Di Big Challenge =====
Sebelum Memulai, Silahkan input nama file ( File.txt )

|                               [1] Input file text                               |
|                               [2] exit                                         |

===== BIG CHALLENGE KELOMPOK 4 =====
Pilihan Anda (1/2) : 1
```

Gambar 2. Gambar tampilan hasil compile

3. Jika memilih nomor 1, maka akan masuk ke menu input file teks.

```
Masukkan nama file input ( File.txt ) : inputkelompok4.txt
```

Gambar 3. Gambar untuk input file teks

4. Setelah memasukkan input file teks, kemudian program akan lanjut menampilkan manu Selamat Di Big Challenge dengan beberapa plihan yang sudah tersedia. Dimana pilihan tersebut terdiri dari :

1. Masukkan data ke dalam fileBin
2. Tampilan data dari Bin
3. Keluar dari Bin Challenge

Berikut adalah gambar tampilannya :

```

Masukkan nama file input ( File.txt ) : inputkelompok4.txt
=====--BIG CHALLENGE KELOMPOK 4 -----
|               Selamat Di Big challenge!               |
|               1. Masukan Data Ke File Bin              |
|               2. Tampilkan Data Dari Bin               |
|               3. Keluar Dari Big challenge              |
|=====--CLOSE THE PROGRAM -----
Masukan Pilihan opsi anda :1

```

Gambar 4. Tampilan pilihan untuk program

5. Dari beberapa pilihan tersebut, kita bisa menjalankan satu persatu sesuai dengan yang diperlukan. Jika memilih nomor 1, maka program akan meminta user untuk memasukkan nama file untuk menyimpan program tersebut. Berikut tampilannya :

```

Ingin disimpan di file yang mana (fileName.bin) ? inputkelompok4.txt
Berhasil Memasukkan data ke inputkelompok4.txt

=====--BIG CHALLENGE KELOMPOK 4 -----
|               [1] Masukan Data ke File Bin              |
|               [2] Tampilkan Data Dari Bin               |
|               [3] Keluar Dari Program                   |
|=====--CLOSE THE PROGRAM -----
Masukan Pilihan opsi anda :1

```

Gambar 5. Tampilan setelah memasukkan nama file untuk menyimpan

6. Jika saat pertama masuk ke program langsung memilih menu 2, maka program akan memberitahu untuk memilih menu 1 terlebih dahulu. Berikut tampilannya :

```

Pilihlah nomor 1 terlebih dahulu

=====--BIG CHALLENGE KELOMPOK 4 -----
|               [1] Masukan Data ke File Bin              |
|               [2] Tampilkan Data Dari Bin               |
|               [3] Keluar Dari Program                   |
|=====--CLOSE THE PROGRAM -----
Masukan Pilihan opsi anda :1

```

Gambar 6. Tampilan pemberitahuan untuk memilih menu 1

7. Jika memilih menu 2, maka program akan meminta nama file output yang dibuat sebelumnya untuk menyimpan data, dan program juga akan meminta pengguna untuk memasukkan nama file yang ingin di baca oleh program, setelah itu program juga akan menampilkan berapa banyak kata yang ingin ditampilkan pada layar. Kemudian program akan menampilkan skor terbesar sebanyak yang diminta oleh pengguna beserta kata dengan skor tersebut dan panjang dari masing-masing kata.

```
===== BIG CHALLENGE KELOMPOK 4 =====
[1] Masukan Data ke File Bin
[2] Tampilkan Data Dari Bin
[3] Keluar Dari Program
===== CLOSE THE PROGRAM =====
Masukan Pilihan opsi anda :2
Ingin dibaca dari file yang mana (filename.bin)? input

Berapa point yang ingin di tampilkan (0<n<=50) : 50
96.0 {keuntungan (10), keuntungan (10)}
90.0 {dikunjungi (10)}
89.0 {menunjukkan (10), peningkatan (11), peningkatan (11), peningkatan (11)}
87.0 {perharinya (10)}
83.0 {diungkapkan (11)}
79.0 {sparspart (9)}
77.0 {didapatnya (10), mengganti (9)}
75.0 {konsumen (8), konsumen (8)}
73.0 {masyarakat (10)}
72.0 {penjualan (9), penjualan (9), penjualan (9)}
71.0 {detikoto (8), menjelang (9)}
68.0 {onderdil (8), onderdil (8), onderdil (8), onderdil (8), onderdil (8), onderdil (8), onderdil (8)}
67.0 {disampaikan (11)}
66.0 {ujarnya (7)}
65.0 {fatmawati (9)}
63.0 {kopling (7), seperti (7), seperti (7)}
61.0 {suzuki (6)}
60.0 {lonjakan (8), moving (6), moving (6), moving (6)}
59.0 {mengakui (8), mengalami (9)}
58.0 {kendaraan (9), khusus (6)}
57.0 {minggu (6), pedagang (8)}
56.0 {menjual (7), persen (6), persen (6), untuk (5)}
55.0 {berarti (7)}
54.0 {diganti (7), terjadi (7), terjadi (7), terjadi (7)}
53.0 {diburu (6), sebelum (7), sebelum (7)}
52.0 {biasanya (8), biasanya (8), biasanya (8), biasanya (8), biasanya (8), biasanya (8), biasanya (8)}
50.0 {begitu (6)}
48.0 {bensin (6), sebesar (7), sebesar (7)}
47.0 {mencari (7), onset (5), pembeli (7), tahun (5), tahun (5)}
46.0 {hingga (6), setiap (6)}
45.0 {datang (6), jakarta (7), jakarta (7), lebaran (7), lebaran (7), lebaran (7), namun (5)}
44.0 {geliat (6), harus (5), jumat (5), nanti (5)}
43.0 {djoni (5), djoni (5), djoni (5), selaku (6)}
41.0 {jenis (5), yang (4), yang (4), yang (4), yang (4), yang (4), yang (4)}
40.0 {suku (4), suku (4)}
39.0 {berkala (7), empat (5)}
38.0 {belum (6), belum (5), belum (5), dapat (5)}
37.0 {lampas (6), ketika (6), merek (5)}
36.0 {deden (5), deden (5), juta (4), juta (4), mobil (5), mobil (5), mobil (5), pasar (5), sampai (6), sampai (6), sampai (6), senada (6)}
35.0 {ujar (4)}
34.0 {cadang (6), cadang (6), haru (4)}
33.0 {secara (6)}
32.0 {enis (4), juga (4), masih (5), masih (5)}
31.0 {busi (4), fast (4), fast (4), fast (4), itu (3), itu (3), lalu (4), roda (4)}
30.0 {angka (5), baru (4)}
29.0 {kalau (5)}
28.0 {atau (4), ramai (5)}
27.0 {adalah (6), hari (4), hari (4), hari (4), hari (4)}
```

Gambar 7. Tampilan banyaknya data dan skor yang diminta

8. Jika memilih menu 3 pada program, maka program akan berakhir.

```
===== BIG CHALLENGE KELOMPOK 4 =====
[1] Masukan Data ke File Bin
[2] Tampilkan Data Dari Bin
[3] Keluar Dari Program
===== CLOSE THE PROGRAM =====
Masukan Pilihan opsi anda :3
PS D:\SEMESTER 2\MK WAJIB\PEMROGRAMAN\BIG CHALLENGE>
```

Gambar 8. Tampilan program pada saat sudah keluar dari program

B. Penjelasan Source Fungsi

1. Metode Scoring

```
int calculate_score(char * nilai) {
    int score = 0;

    while (*nilai != '\0') {
        if (*nilai == 'a') {
            score += 1;
        } else if (*nilai == 'b') {
            score += 4;
        } else if (*nilai == 'c') {
            score += 3;
        } else if (*nilai == 'd') {
            score += 7;
        } else if (*nilai == 'e') {
            score += 5;
        } else if (*nilai == 'f') {
            score += 6;
        } else if (*nilai == 'g') {
            score += 10;
        } else if (*nilai == 'h') {
            score += 8;
        } else if (*nilai == 'i') {
            score += 5;
        } else if (*nilai == 'j') {
            score += 9;
        } else if (*nilai == 'k') {
            score += 6;
        } else if (*nilai == 'l') {
            score += 9;
        } else if (*nilai == 'm') {
            score += 8;
        } else if (*nilai == 'n') {
            score += 12;
        } else if (*nilai == 'o') {
            score += 10;
        } else if (*nilai == 'p') {
            score += 11;
        } else if (*nilai == 'q') {
            score += 15;
        } else if (*nilai == 'r') {
```

```

        score += 13;
    } else if (*nilai == 's') {
        score += 10;
    } else if (*nilai == 't') {
        score += 14;
    } else if (*nilai == 'u') {
        score += 12;
    } else if (*nilai == 'v') {
        score += 15;
    } else if (*nilai == 'w') {
        score += 15;
    } else if (*nilai == 'x') {
        score += 14;
    } else if (*nilai == 'y') {
        score += 18;
    } else if (*nilai == 'z') {
        score += 16;
    }
    nilai++; // Maju ke karakter berikutnya
}

return score;
}

```

Fungsi scoring adalah sebuah fungsi yang digunakan untuk menghitung skor dari sebuah nilai atau kata berdasarkan aturan penilaian tertentu. Fungsi ini menerima satu parameter, yaitu char * nilai, dan menghitung skor dengan cara memeriksa setiap karakter dalam token tersebut. Jika huruf/karakter (A-Z), (a-z) maka nilai skor akan ditambahkan sesuai dengan nilai yang telah ditentukan untuk masing-masing huruf/karakter.

2. Fungsi file bacaBin

Fungsi file bacaBin.c berisi fungsi readBinary yang digunakan untuk membaca data yang telah disimpan ke dalam file biner dan memuatnya kembali ke dalam struktur data program, kemudian untuk meminta nama file dari pengguna dan jumlah data yang ingin ditampilkan dan juga untuk menampilkan data yang telah dibaca dari file biner dalam format yang mudah dibaca oleh pengguna.

- Code untuk membuka file biner serta memeriksa pembukaannya

C bacabin.c

```
void readBinary(result hasil[],int kata){
    FILE * bin;
    char fileName[20];
    printf("Ingin dibaca dari file yang mana (filename.bin)? ");
    scanf("%s",fileName);
    bin = fopen(fileName,"rb");

    if(bin==NULL){
        printf("Tidak Dapat Membuka File...\n");
        exit(1);
    }
}
```

- Code untuk membaca data dari file biner

```
for(int i=0;i<kata;i++){
    fread(&hasil[i].skor,sizeof(hasil[i].skor),1,bin);
    fread(hasil[i].kata,sizeof(hasil[i].kata),1,bin);
    fread(&hasil[i].panjangKata,sizeof(hasil[i].panjangKata),1,bin);
}
```

- `Code meminta input dari pengguna

```
int n;
int i=0;
int j=0,k;
do{
    printf("\n");
    printf("Berapa point yang ingin di tampilkan (0<n<=50) : ");
    scanf(" %d",&n);
    if(n<0||n>50){
        printf("\nMasukkan angka dari 1 - 50 saja!!!\n");
    }
}while(n<0||n>50);
```

- Cara data untuk menampilkan data yang dibaca

```
for(i=0;i<n;i++){
    for(j=0;j<kata;j++){
        printf("%.1f %s (%d)",hasil[j].skor,hasil[j].kata,hasil[j].panjangKata);
        for(k=j+1;k<kata;k++){
            if(hasil[j].skor==hasil[k].skor){
                printf(", %s (%d)",hasil[k].kata,hasil[k].panjangKata);
            }
            else{
                j=k;
                break;
            }
        }
        printf("}\n");
        break;
    }
}
```

loop di mana variabel *i* digunakan sebagai indeks luar, *j* sebagai indeks dalam untuk sebuah array, dan *k* sebagai indeks dalam loop dalam yang digunakan untuk memeriksa elemen-elemen lain dalam array. Pada setiap iterasi loop luar (*i*), program mencetak nilai-nilai dari array hasil, yaitu skor, kata, dan panjang kata, dengan format tertentu. Kemudian, dalam loop dalam pertama (*j*), program mencetak elemen-elemen array yang memiliki skor yang sama dengan elemen saat ini, dan melanjutkan ke elemen-elemen berikutnya jika ada. Setelah selesai mencetak elemen-elemen dengan skor yang sama, program kembali ke loop luar untuk iterasi selanjutnya, dan proses ini diulang sebanyak *n* kali

3. Fungsi file bubbleSort.c

Fungsi dari file bubbleSort ini digunakan untuk memastikan bahwa data yang akan ditampilkan atau disimpan tersusun dengan benar, sehingga memudahkan pengguna untuk melihat data. Dalam file ini juga harus ada struct atau struktur data yang berfungsi untuk mendefinisikan data yang akan diurutkan termasuk anggota skor dan kata.

Berikut Fungsi struct untuk bubbleSort:

C bubbleSort.c

```
void bubbleSort(data datas[],int kata)
{
    //meurutkan skor(desc)
    for(int i=0;i < kata; i++){
        for(int j=0;j<kata-i-1;j++){
            if(datas[j].skor<datas[j+1].skor){
                data ganti = datas[j];
                datas[j] = datas[j+1];
                datas[j+1] = ganti;
            }
        }
    }

    //mengurutkan kata sesuai abjad
    for(int i=0;i < kata; i++){
        for(int j=0;j<kata-i-1;j++){
            if((datas[j].skor==datas[j+1].skor)&&(strcmp(datas[j].kata, datas[j+1].kata) > 0)){
                data ganti = datas[j];
                datas[j] = datas[j+1];
                datas[j+1] = ganti;
            }
        }
    }
}
```

- Bubblesort ialah Struktur data berfungsi untuk menyimpan informasi yang dari

skor dan kata yang akan diurutkan berdasarkan skor dalam urutan menurun dan kata dalam urutan abjad jika skornya sama. Sehingga program dapat mengelola dan menyiapkan data dengan cara yang terstruktur dan mudah dipahami oleh pengguna.

4. Fungsi file garis.c

Fungsi dari file garis.c dengan nama countLine digunakan untuk menghitung jumlah baris dalam sebuah file teks bernama input.txt. fungsi ini juga untuk membuka file, membaca isinya karakter demi karakter, dan menghitung jumlah baris dengan menghitung karakter newline(\n). fungsi ini sangat diperlukan dalam program ini karena digunakan untuk menentukan berapa banyak baris data yang ada dalam file teks.

- Fungsi untuk membuat file.txt dan mengecek kesalahan

C garis.c

```
int countLine(int n)
{
    char huruf;
    FILE * input;
    input = fopen("input.txt","r");

    if(input==NULL){
        printf("Tidak bisa membuka file...");
        exit(1);
    }
```

- Code untuk membaca file dan menghitung baris

```
huruf=fgetc(input);
while(!feof(input)){
    if(huruf=='\n'){
        n++;
    }
    huruf=fgetc(input);
}
n+=1;
```

5. Fungsi file hitungHuruf.c

Fungsi file `hitungHuruf.c` dengan nama `countWord` digunakan untuk menghitung jumlah kata dalam sebuah string yang mewakili kalimat. Fungsi ini penting dalam program ini karena memerlukan pengolahan terhadap teks, seperti pada menghitung jumlah kata dalam kalimat yang di ambil dari file teks. Fungsi ini bekerja dengan memeriksa setiap karakter dalam string untuk menemukan spasi yang memisahkan kata-kata, dan menghitung jumlah kata berdasarkan jumlah spasi yang ditemukan.

Berikut Fungsi untuk file `hitungHuruf.c` :

C `hitunghuruf.c`

```
int countWord(int kata, int karakter, char *kalimat){
    for (int i = 0; i < karakter; i++){
        if (kalimat[i] == ' '){
            kata++;
            if(kalimat[i+1]==' '){
                ++i;
                if(kalimat[i+1]==' '){
                    ++i;
                }
            }
        }
    }

    kata+=1;
}
```

Fungsi ini bertujuan untuk menghitung jumlah kata dalam sebuah kalimat yang diwakilkan oleh string yang diberikan (kalimat). Iterasi dilakukan melalui setiap karakter dalam string. Jika karakter yang sedang diperiksa adalah spasi (' '), maka itu menandakan akhir kata, sehingga variabel `kata` yang merepresentasikan jumlah kata bertambah satu. Namun, untuk mengatasi kemungkinan spasi ganda, ada pengecekan tambahan: jika karakter berikutnya juga merupakan spasi, iterasi akan maju satu langkah lagi untuk menghindari penghitungan ganda.

6. Fungsi file `hitungAngka.c`

Fungsi dari file `hitungAngka` dengan nama `characterToVariabel` digunakan untuk membaca file teks `input.txt` dan memasukkan karakter-karakter valid ke dalam sebuah string yang disimpan dalam variabel `kalimat`. Fungsi ini berguna untuk menggabungkan semua teks dalam file menjadi satu string, mengonversi karakter menjadi huruf kecil, dan menggantikan karakter non-alfanumerik dengan spasi. Ini

penting dalam konteks program yang melakukan analisis teks atau pemrosesan kata, seperti penghitungan kata atau pengurutan kata berdasarkan kriteria tertentu.

- Code untuk membuka file dan membaca isinya

C hurufangka.c

```
void characterToVariable(int garis,int panjang,char * kalimat){
    char string[garis][panjang];
    int i=0;
    int a=0;
    FILE * input;
    input = fopen("input.txt","r");
    if(input==NULL){
        printf("Tidak dapat membuka file...");
        exit(1);
    }
}
```

- Code untuk membaca baris dan memproses karakter

```
while(fgets(string[i],(panjang*sizeof(int)),input)!=NULL){
    i++;
}
```

- Code untuk menggabungkan karakter valid ke string besar

```
for(int j=0;j < garis;j++){
    for(int k=0;k < panjang;k++){
        if(string[j][k]=='\0' || string[j][k]==EOF){
            break;
        }
        if(isalpha(string[j][k]) || isdigit(string[j][k]) || string[j][k]==' '){
            kalimat[a]=string[j][k];
            kalimat[a]=tolower(kalimat[a]);
            a++;
        }
        else{
            kalimat[a]=' ';
            a++;
        }
    }
}
fclose(input);
```

Dua loop bersarang digunakan untuk mengiterasi melalui elemen-elemen dari array string. Loop pertama (j) digunakan untuk mengiterasi melalui setiap "baris" atau "indeks" dalam array, sedangkan loop kedua (k) digunakan untuk mengiterasi melalui setiap "kolom" atau "karakter" dalam setiap baris.

setiap karakter dalam string[j][k] diperiksa. Jika karakter tersebut adalah null terminator (\0) atau end-of-file (EOF), itu menandakan akhir dari string atau

file, dan loop dihentikan untuk baris saat ini. Jika karakter tersebut adalah huruf, digit, atau spasi, maka itu akan dimasukkan ke dalam array kalimat setelah diubah menjadi huruf kecil menggunakan fungsi `tolower()`, dan `a` (mungkin adalah indeks untuk kalimat) ditingkatkan. Jika karakter tersebut bukan huruf atau digit, itu akan diganti dengan spasi dalam kalimat.

7. Fungsi file kata.c

Fungsi dari file `kata.c` dengan nama `countCharacter` digunakan untuk menghitung jumlah karakter dalam sebuah file teks yang bernama `input.txt`. fungsi ini membaca setiap karakter dalam file dan menghitung total jumlahnya. Fungsi ini penting dalam program ini dikarenakan untuk mengetahui informasi tentang ukuran teks, misalnya untuk mengalokasikan memori yang tepat saat memproses teks atau saat menghitung jumlah kata dan baris dalam file.

- Fungsi untuk membuka file dan membaca karakter

```
C kata.c

int countCharacter(int karakter){
    char huruf;
    FILE * input;
    input = fopen("input.txt","r");
    if(input==NULL){
        printf("Tidak dapat membuka file...");
        exit(1);
    }
}
```

- cara untuk menghitung karakter dalam file

```
huruf=fgetc(input);
while(!feof(input)){
    // if(isalpha(huruf)||isdigit(huruf)||huruf==' '||huruf=='\n'){
    |   karakter++;
    // }
    huruf=fgetc(input);
}
fclose(input);
return karakter;
```

8. Fungsi file panjang.c

Fungsi dari file `panjang.c` dengan nama `countLenght` digunakan untuk menghitung

panjang bars terpanjang dalam file teks bernama input.txt. . Fungsi ini membaca setiap karakter dalam file dan melacak panjang setiap baris untuk menemukan baris terpanjang. Ini penting dalam konteks program yang membutuhkan informasi tentang struktur teks, khususnya untuk menentukan alokasi memori yang tepat saat memproses baris dari file.

- Code untuk membuka file dan menganalisis variable

C panjang.c

```
int countLenght(int panjang){
    int n=0;
    char huruf;
    char * sementara;
    FILE * input;
    input = fopen("input.txt","r");

    if(input==NULL){
        printf("Tidak bisa membuka file...");
        exit(1);
    }
}
```

- Code untuk menghitung panjang baris terpanjang

```
huruf=fgetc(input);
while(!feof(input)){
    if(huruf=='\n'){
        ++n;
        if(n>panjang){
            panjang=n;
        }
        n=0;
        huruf=fgetc(input);
    }
    n++;
    huruf=fgetc(input);
    if(huruf==EOF){
        if(n>panjang){
            panjang=n;
        }
    }
}

fclose(input);
return (panjang);
```

9. Fungsi file pisahKata.c

Fungsi dari file pisahKata.c dengan nama strtokCharacter sangat berguna dalam memproses string teks menjadi token-token (kata-kata) yang lebih mudah diolah dalam program. Dengan memisahkan teks menjadi token, program dapat dengan mudah mengakses dan menganalisis setiap kata secara terpisah, misalnya untuk menghitung skor atau melakukan operasi lainnya. Ini membantu dalam pengolahan teks dan analisis data yang memerlukan pemrosesan kata per kata.

- Fungsi untuk memotong string kata atau pisah data

C pisahkata.c

```
void strtokCharacter(char *nilai, char *kalimat, data datas[])
{
    int i = 0;
    nilai = strtok(kalimat, " ");
    while (nilai != NULL){
        // printf("%s\n", nilai);
        strcpy(datas[i].kata, nilai);
        datas[i].skor = calculate_score(nilai);
        datas[i].panjangKata = strlen(nilai);
        nilai = strtok(NULL, " ");
        i++;
    }
}
```

10. Fungsi file tulisBin.c

Fungsi dari file tulisBin.c dengan nama writerToBinary sangat penting digunakan dalam program ini karena memungkinkan data yang telah diproses (seperti hasil pengurutan atau analisis kata) untuk disimpan dengan efisien dalam format biner. Ini membuat data lebih mudah diakses di kemudian hari, menghemat ruang penyimpanan, dan meningkatkan kecepatan baca/tulis dibandingkan dengan format teks. Fungsi ini membantu dalam mengelola data yang mungkin digunakan kembali tanpa harus melakukan pemrosesan ulang dari awal

- Code proses input nama file

C tulisbin.c

```
void writeToBinary(data datas[],int kata){
    char fileName[20]    ;

    printf("\nIngin disimpan di file yang mana (fileName.bin) ? ");
    scanf("%s",fileName);
```

- Code untuk membuka file biner untuk penulisan

```
FILE * bin;
bin = fopen(fileName,"wb");
```

- Code untuk menulis data ke file biner

```
for(int i=0;i<kata;i++){
    fwrite(&datas[i].skor,sizeof(datas[i].skor),1,bin);
    fwrite(datas[i].kata,sizeof(datas[i].kata),1,bin);
    fwrite(&datas[i].panjangKata,sizeof(datas[i].panjangKata),1,bin);
}

printf("\nBerhasil Memasukkan data ke %s\n",fileName);

fclose(bin);
```

11. Fungsi file main.c

Fungsi dari file main.c digunakan untuk mengatur alur utama dari program dengan fungsi-fungsi lain yang melakukan tugas seperti membaca file teks, menghitung karakter dan kata, memproses kalimat menjadi kata-kata, dan menyimpan serta membaca data dari file biner.

- Fungsi utama (main)

```
int main() {
    int garis = 0;
    int panjang = 0;
    int karakter = 0;
    int kata=0;
    FILE *Big;
    char fileName[20];
    char *nilai;

    //Fungsi System
    BersihkanSistem();
    Jeda();
```

Fungsi Main Berisikan integer garis = 0, Panjang = 0, karakter = 0, kata = 0. Lalu ada char array yang berisikan filename[20], lalu ada pointer char *nilai

- Mendeklarasikan Fungsi menghitung jumlah karakter, baris, dan panjang

```
//Menghitung ada berapa banyak karakter
karakter = countCharacter(karakter);

//Menghitung berapa banyak baris
garis = countLine(garis);

//Menghitung kalimat tergaris ada pada karakter
panjang = countLenght(panjang);
```

- Mendeklarasikan Fungsi untuk mengelola data karakter

```
//Mendeklarasikan array kalimat terpanjang
char kalimat[karakter];

//Melakukan Scan //dan memasukkan hasil scan ke array kalimat
characterToVariable(garis, panjang, kalimat);

//menghitung jumlah kata yang sudah di masukkan ke array kalimat
kata = countWord(kata,karakter,kalimat);

//mendeklarasikan struct data dengan array sebanyak jumlah kata
data datas[kata];

//tokenisasi dan memamsukkan hasil token ke data struct
strtokCharacter(nilai, kalimat,datas);
```

Saya mendeklasrikan semua Fungsi ini di dalam Main.c

- Code untuk menampilkan menu utama

```

    int Kelompok4;
    // Tampilkan menu masuk
    int Answer;
do {
    BersihkanSistem();
    printf("----- Selamat Datang Di Big Challenge -----\\n");
    printf(" Sebelum Memulai, Silahkan input nama file ( File.txt )\\n\\n");
    printf(" |                      [1] Input file text                      |\\n");
    printf(" |                      [2] exit                                |\\n\\n");
    printf("----- BIG CHALLENGE KELOMPOK 4 -----\\n");
    printf(" Pilihan Anda (1/2) : ");
    scanf("%d", &Kelompok4);

    if (Kelompok4 == 1) {
        int Berhasil = 0;
        BersihkanSistem();
        printf(" Masukkan nama file input ( File.txt ) : ");
        scanf("%s", fileName);
        Big = fopen(fileName, "r");

        if (Big == NULL) {
            printf(" File tidak ditemukan, Mohon tunggu ...\\n");
            Jeda();
        } else {

```

Disini saya mendeklarasikan int Kelompok4 agar memeriksa bahwa nilai kelompok 4 = 1, lalu printf “masukan nama file input (file.txt)” lalu scanf (“&s”, filename) Filename Berfungsi mencari data dengan keyboard input user

- untuk menampilkan menu dan proses pilihan pengguna

```
do {
    printf("----- BIG CHALLENGE KELOMPOK 4 -----\\n");
    printf("|                Selamat Di Big challenge!                |\\n");
    printf("|                1. Masukan Data Ke File Bin                |\\n");
    printf("|                2. Tampilkan Data Dari Bin                  |\\n");
    printf("|                3. Keluar Dari Big challenge                 |\\n");
    printf("-----CLOSE THE PROGRAM -----\\n");
    printf(" Masukan Pilihan opsi anda :");
    scanf("%d", &Kelompok4);

    // Proses pilihan
    if (Kelompok4 == 1) {
        system("cls");
        writeToBinary(datas, kata);
        break;
    } else if (Kelompok4 == 2) {
        system("cls");
        printf("\\nPilihlah nomor 1 terlebih dahulu\\n\\n");
        break;
    } else if (Kelompok4 == 3) {
        exit(1);
    } else {
        system("cls");
        printf("Selamat Datang di BIG CHALLENGE\\n");
        printf("Pilihan Kamu tidak valid!!\\nSilahkan pilih Ulang");
    }
} while (1);

result hasil[kata];
```

Disini seperti diatas saya mendeklarasikan int Kelompok4 agar memeriksa bahwa nilai kelompok4 = 1, lalu if saya mendeklarasikan “cls” yang berarti system operasi nya Window lalu saya juga mendeklarasikan “writeToBinary” atau tulis ke filebin. Lalu break, setelah itu ada else if yang kelompok4 = 2 yang artinya memeriksa bahwa nilai kelompok4 = 2, disini yang berartikan jika kita memilih opsi 2 tanpa, memilih opsi 1 maka akan terprintf “Pilihlah Nomor 1 Terlebih dahulu”, dan lalu kelompok4 = 3 itu merupakan fungsi exit atau keluar program

12. Fungsi file header.h

Fungsi dari file header.h digunakan untuk menggabungkan dan deklarasi fungsi – fungsi dan struktur dari file.c lainnya.

- Deklarasi Struct dan fungsi – fungsi

```
#ifndef HEADER_H
#define HEADER_H

typedef struct dataMentah
{
    double skor;
    char kata[50];
    int panjangKata;
} data;

typedef struct dataHasil{
    double skor;
    char kata[50];
    int panjangKata;
} result;

int countCharacter(int);
int countLine(int);
int countLenght(int);
void characterToVariable(int, int, char *);
void strtokCharacter(char* token, char* kalimat, data* datar);
int countWord(int ,int ,char *);
int calculate_score(char *);
void bubbleSort(data datas[],int );
void selectionSort(data datas[],int );
void writeToBinary(data datas[],int );
void readBinary(result hasil[],int );
void BersihkanSistem();
void Jeda();
```

13. Fungsi file linuxOrWind.c

Fungsi dari file linuxOrWind.c digunakan untuk operasi sistem

```
void BersihkanSistem() {
#ifdef _WIN32
    system("cls");
#else
    system("clear");
#endif
}
```

- Code untuk menjeda sistem

```
void Jeda() {  
#ifdef _WIN32  
    Sleep(2000); // Windows  
#else  
    sleep(2); // Linux  
#endif  
}
```

14. Fungsi file input.txt

Fungsi dari file input.txt berisi kalimat-kalimat yang akan dibaca oleh file c lainnya dan akan di keluarkan pada file output sesuai dengan keinginan dari pengguna.

Berikut contoh dari file input.txt :

```
inputkelompok4.txt  
1  Pasar Onderdil Baru Ramai di Jakarta.  
2  Dua minggu sebelum Lebaran penjualan onderdil kendaraan roda empat belum menunjukan geliat yang berarti, dan saat ini penjualan onderdil masih biasa saja.
```

15. Fungsi file makeFile

File makeFile berisi tentang cara mengkompile program tersebut supaya programnya bisa jalan.

Berikut contoh file makeFile :

```
M MakeFile  
1  Run=  
2  gcc -o kelompok4 main.c bubbleSort.c bacabin.c garis.c hurufangka.c kata.c panjang.c pisahkata.c skor.c tulisbin.c hitunghuruf.c linuxorwind.c  
3  ./kelompok4 WIN
```

LINK YT DAN GITHUB

-. <https://youtu.be/cxfJq7xXG3o>

-. <https://github.com/Zipiyy/BIG-CHALNGE--->

KELOMPOK-4.git