

Nama : M Alief Alfaridzi

NIM : 1203230045

Kelas : IF 03-03

Tugas Laporan OTS Week 4

Soal

Source Code

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  #define MAX_LENGTH 2024
6  #define MIN_LENGTH 1945
7
8  void lessThanRequired (int* lengthOfText){
9      printf("The length of your text is less than specified, please update your text\n");
10     *lengthOfText = MIN_LENGTH;
11 }
12
13 void equalThanRequired (int* lengthOfText){
14     printf("Thank you, Your text length is correct\n");
15 }
16
17 void moreThanRequired (int* lengthOfText){
18     printf("Your text is too long, please reduce the text\n");
19     *lengthOfText = MIN_LENGTH;
20 }
21
22 int checkLenghtRequirement(char* text){
23     int length = strlen(text);
24     if (length < MIN_LENGTH)
25         return 0;
26     else if (length == MIN_LENGTH)
27         return 1;
28     else
29         return 2;
30 }
31
32 int main() {
33     int lengthOfText, selectOption;
34     FILE *fptr = NULL;
35     char text[MAX_LENGTH];
36
37     fptr = fopen("file.txt", "r");
38
39     if(fptr == NULL){
40         printf("Error");
41         exit(1);
42     }
43
44     fgets(text, MAX_LENGTH, fptr);
45
46     fclose(fptr);
47
48     selectOption = checkLenghtRequirement(text);
49
50     // TODO
51     // Pada fungsi checkLenghtRequirement akan mengembalikan sebuah angka
52     // angka tersebut digunakan untuk memilih secara otomatis salah satu fungsi yang harus diisi
53     // jika fungsi checkLenghtRequirement() mengembalikan nilai 0, maka
54     //     tampilkan - > The length of your text is less than specified, please update your text
55     //     update nilai lengthOfText ke minimum requirement melalui pointer menggunakan operasi aritmatika
56
57     // jika fungsi checkLenghtRequirement() mengembalikan nilai 1, maka
58     //     tampilkan - > Thank you, Your text length is correct
59
60     // jika fungsi checkLenghtRequirement() mengembalikan nilai 2, maka
61     //     tampilkan - > Your text is too long, please reduce the text
62     //     update nilai lengthOfText ke minimum requirement melalui pointer menggunakan operasi aritmatika
63
64     // Catatan :
65     //     - tidak diperkenankan menggunakan if atau switch dalam perpindahan fungsi
66     //     - sesuai dengan requirement diatas.
67     //     - baris kode tidak lebih dari 100 (include comment ini)
68     //     - tidak diperkenankan mengganti yang tertera pada starter code dalam alasan apapun
69
70     void (*functions[3])(int*) = {lessThanRequired, equalThanRequired, moreThanRequired};
71     lengthOfText = selectOption != 0 ? MIN_LENGTH : 0;
72
73     functions[selectOption](&lengthOfText);
74
75     printf("\nThe Length is updated to %d", lengthOfText);
76
77     return 0;
78 }
```

Penjelasan Code

Berikut ini penjelasan code program persatu baris atau perbaris:

- Baris (1) `#include <stdio.h>` Mengimpor header file `stdio.h` yang berisi deklarasi fungsi standar untuk input dan output seperti `printf` dan `scanf`.
- Baris (2) `#include <stdlib.h>` Mengimpor header file `stdlib.h` yang berisi deklarasi fungsi standar untuk alokasi memori, pengelolaan proses, dan fungsi-fungsi utilitas lainnya seperti `exit`.
- Baris (3) `#include <string.h>` Mengimpor header file `string.h` yang berisi deklarasi fungsi-fungsi untuk manipulasi string seperti `strlen`.
- Baris (5) `#define MAX_LENGTH 2024` Mendefinisikan konstanta `MAX_LENGTH` dengan nilai 2024, yang merupakan panjang maksimum teks yang dapat ditampung dalam array `text`.
- Baris (6) `#define MIN_LENGTH 1945` Mendefinisikan konstanta `MIN_LENGTH` dengan nilai 1945, yang merupakan panjang minimum yang diperlukan untuk teks.
- Baris (8) `void lessThanRequired (int* lengthOfText){` Deklarasi fungsi `lessThanRequired` yang menerima pointer ke integer (`int*`) sebagai parameter. Fungsi ini akan menampilkan pesan jika panjang teks kurang dari yang diharapkan dan memperbarui panjang teks menggunakan pointer.
- Baris (9) `printf("The length of your text is less than specified, please update your text\n");` berfungsi mencetak output seperti yang telah dimasukkan sebelumnya dan akan ditampilkan bila jumlah text kurang dari jumlah minimum text.
- Baris (10) `*lengthOfText = MIN_LENGTH;` Mengupdate nilai `lengthOfText` ke nilai `MIN_LENGTH`.
- Baris (11) `}` mengakhiri fungsi `lessThanRequired`.
- Baris (13) `void equalThanRequired (int* lengthOfText){` Deklarasi fungsi `equalThanRequired` yang mirip dengan fungsi `lessThanRequired`, namun menampilkan pesan jika panjang teks sudah sesuai.
- Baris (14) `printf("Thank you, Your text length is correct\n");` berfungsi mencetak output seperti yang telah dimasukkan sebelumnya dan akan ditampilkan bila jumlah text sesuai atau sama dengan panjang teks minimum.
- Baris (15) `}` mengakhiri fungsi `equalThanRequired`.
- Baris (17) `void moreThanRequired (int* lengthOfText){` Deklarasi fungsi `moreThanRequired` yang mirip dengan kedua fungsi sebelumnya, namun menampilkan

pesan jika panjang teks terlalu panjang dan memperbarui panjang teks menggunakan pointer.

- Baris (18) `printf("Your text is too long, please reduce the text\n");` berfungsi mencetak output seperti yang telah dimasukkan sebelumnya dan akan ditampilkan bila jumlah text melebihi jumlah maximal kata.
- Baris (19) `*lengthOfText = MIN_LENGTH;` Mengupdate nilai `lengthOfText` ke nilai `MIN_LENGTH`.
- Baris (20) `}` mengakhiri fungsi `moreThanRequired`.
- Baris (22) `int checkLenghtRequirement(char* text){` Deklarasi fungsi `checkLenghtRequirement` yang menerima pointer ke karakter (`char*`) sebagai parameter dan mengembalikan nilai integer yang menunjukkan apakah panjang teks kurang dari, sama dengan, atau lebih dari panjang minimum yang diharapkan.
- Baris (23) `int length = strlen(text);` berfungsi menghitung Panjang teks.
- Baris (24) `if (length < MIN_LENGTH)` berfungsi jika `length` kurang dari `min_length` maka.
- Baris (25) `return 0;` jika panjang kurang dari `MIN_LENGTH`, kembalikan 0.
- Baris (26) `else if (length == MIN_LENGTH)` berfungsi jika `length` sama dengan `min_length` maka.
- Baris (27) `return 1;` jika panjang sama dengan `MIN_LENGTH`, kembalikan 1.
- Baris (28) `else` berfungsi jika kondisi diatas belum terpenuhi maka lanjut lagi.
- Baris (29) `return 2;` jika panjang lebih dari `MIN_LENGTH`, kembalikan 2.
- Baris (30) `}` mengakhiri fungsi `checkLenghtRequirement`.
- Baris (32) `int main(){` mendeklarasikan fungsi utama, yang merupakan titik masuk program.
- Baris (33) `int lengthOfText, selectOption;` Deklarasi variabel `lengthOfText` dan `selectOption` sebagai integer.
- Baris (34) `FILE *fptr = NULL;` Deklarasi pointer ke FILE `fptr` dan inisialisasi dengan `NULL`.
- Baris (35) `char text[MAX_LENGTH];` Deklarasi array `text` dengan panjang maksimum yang telah ditentukan sebelumnya.
- Baris (37) `fptr = fopen("file.txt", "r");` Membuka file `file.txt` dalam mode baca ("`r`") dan menyimpan file pointer dalam variabel `fptr`.

- Baris (39 – 41) `if(fptr == NULL){ printf("Error"); exit(1); }` Memeriksa apakah file berhasil dibuka. Jika tidak, cetak pesan error dan keluar dari program dengan kode kesalahan 1.
- Baris (44) `fgets(text, MAX_LENGTH, fptr);` Membaca teks dari file `file.txt` dan menyimpannya dalam array `text`. `fgets` digunakan untuk menghindari overflow array.
- Baris (46) `fclose(fptr);` Menutup file setelah selesai membacanya.
- Baris (48) `selectOption = checkLenghtRequirement(text);` Memeriksa panjang teks menggunakan fungsi `checkLenghtRequirement` dan menyimpan hasilnya dalam variabel `selectOption`.
- Baris (70) `void (*functions[3])(int*) = {lessThanRequired, equalThanRequired, moreThanRequired};` Mendefinisikan array pointer ke fungsi yang akan dipanggil berdasarkan hasil dari `checkLenghtRequirement`.
- Baris (71) `lengthOfText = selectOption != 0 ? MIN_LENGTH : 0;` Memperbarui nilai `lengthOfText` dengan panjang minimum jika panjang teks kurang dari yang diharapkan.
- Baris (73) `functions[selectOption](&lengthOfText);` Memanggil fungsi yang sesuai dari array pointer ke fungsi berdasarkan `selectOption` dan memperbarui `lengthOfText` jika diperlukan.
- Baris (75) `printf("\nThe Length is updated to %d", lengthOfText);` Mencetak panjang teks yang telah diperbarui.
- Baris (77) `return 0;` ini mengembalikan nilai 0, menandakan bahwa program telah dijalankan dengan sukses dan akan keluar dari fungsi `main`.
- Baris (78) `}` Menutup fungsi `main` utama program.

Output Program

Program ini dirancang untuk membaca file teks bernama `"file.txt"` dan memeriksa panjang teks yang dibaca dari file tersebut. Berdasarkan panjang teks, ini akan menampilkan pesan tertentu dan memperbarui panjang teks jika perlu.

Mari kita lihat keluaran program langkah demi langkah:

1. Program membuka `"file.txt"` dalam mode baca menggunakan `fopen()` fungsi tersebut. Jika pembukaan file gagal, maka akan dicetak `"Error"` dan keluar dari program.

2. Program membaca sebaris teks dari file menggunakan `fgets()` fungsi dan menyimpannya dalam textvariabel. Panjang maksimum teks yang dapat dibaca ditentukan oleh `MAX_LENGTH` konstanta.
3. Setelah membaca teks, program menutup file menggunakan `fclose()` fungsi tersebut.
4. Program memanggil `checkLenghtRequirement()` fungsi dan meneruskan textvariabel sebagai argumen. Fungsi ini menghitung panjang teks menggunakan `strlen()` fungsi tersebut dan mengembalikan nilai berdasarkan panjangnya:
 - Jika panjang teks kurang dari panjang minimum yang disyaratkan (`MIN_LENGTH`), fungsi akan mengembalikan 0.
 - Jika panjang teks sama dengan panjang minimum yang diperlukan, fungsi akan mengembalikan 1.
 - Jika panjang teks lebih besar dari panjang minimum yang disyaratkan, fungsi akan mengembalikan 2.
5. Program ini membuat larik penunjuk fungsi yang diberi nama `functions` dan menginisialisasinya dengan tiga fungsi: `lessThanRequired()`, `equalThanRequired()`, dan `moreThanRequired()`. Fungsi-fungsi ini dipanggil berdasarkan nilai yang dikembalikan oleh `checkLenghtRequirement()`.
6. Program memperbarui `lengthOfText` variabel berdasarkan nilai yang dikembalikan oleh `checkLenghtRequirement()`. Jika nilainya bukan 0, maka akan disetel `lengthOfText` ke panjang minimum yang diperlukan (`MIN_LENGTH`).
7. Program memanggil fungsi yang ditentukan oleh nilai yang dikembalikan `checkLenghtRequirement()` dan meneruskan `lengthOfText` variabel dengan referensi menggunakan `&` operator. Fungsi ini mencetak pesan tertentu berdasarkan nilai yang diteruskan ke pesan tersebut.
8. Terakhir, program mencetak nilai terbaru dari `lengthOfText` penggunaan `printf()` fungsi tersebut.

Output program akan bergantung pada konten file "file.txt" dan panjangnya.

Contoh Output Program

- Jika Panjang teks kurang dari Panjang minimum yang telah ditentukan.

```
C: > Users > HP > OneDrive > Documents > ALGOSDAT > file.txt
1  telkom university
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\HP\OneDrive\Documents\ALGOSDAT> cd "c:\Users\HP\OneDrive\Documents\ALGOSDAT\" ; if ($?) { gcc
The length of your text is less than specified, please update your text

The Length is updated to 1945
PS C:\Users\HP\OneDrive\Documents\ALGOSDAT> |
```

- Jika Panjang teks sama dengan Panjang minimum yang diperlukan 1945 karakter huruf.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\HP\OneDrive\Documents\ALGOSDAT> cd "c:\Users\HP\OneDrive\Documents\ALGOSDAT\" ; if ($?) { gcc
Thank you, Your text length is correct

The Length is updated to 1945
PS C:\Users\HP\OneDrive\Documents\ALGOSDAT> █
```

- Jika Panjang teks lebih besar dari Panjang minimum yang di isyartkan lebih dari 1945 karakter hingga 2024 panjang karakter.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\HP\OneDrive\Documents\ALGOSDAT> cd "c:\Users\HP\OneDrive\Documents\ALGOSDAT\" ; if ($?) { gcc
Your text is too long, please reduce the text

The Length is updated to 1945
PS C:\Users\HP\OneDrive\Documents\ALGOSDAT> █
```