

Nama : Muhammad Naufal Ramadhani

Nim : 1203230036

Kelas : IF 03-03

## 1. SS Source Code

```
C prakes2.c > main()
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(){
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 checkLengthRequirement(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     void (*functionPointer[3])(int *) = {lessThanRequired, equalThanRequired, moreThanRequired};
35     FILE *fptr = NULL;
36     char text[MAX_LENGTH];
37
38     fptr = fopen("file.txt", "r");
39
40     if(fptr == NULL){
41         printf("Error");
42         exit(1);
43     }
44
45     fgets(text, MAX_LENGTH, fptr);
46     fclose(fptr);
47
48     selectOption = checkLengthRequirement(text);
49     functionPointer[selectOption](&lengthOfText);
50
51     printf("\nThe length is updated to %d\n", lengthOfText);
52
53     return 0;
54 }
55
56
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\YUS\OneDrive\Documents\kopal\TP3.c> cd "C:\Users\YUS\OneDrive\Documents\kopal\TP3.c\" ; if ($?) { gcc prakes2.c -o prakes2 } ; if ($?) { .\prakes2 }
The length of your text is less than specified, please update your text

The length is updated to 1945
PS C:\Users\YUS\OneDrive\Documents\kopal\TP3.c>
```

Ini adalah program C yang membaca teks dari sebuah file bernama "file.txt" dan kemudian memeriksa panjang teksnya. Program ini memiliki beberapa fungsi dan menggunakan konsep pointer ke fungsi untuk memilih tindakan yang tepat berdasarkan panjang teks yang terbaca.

1. **#include <stdio.h>**: Ini adalah preprocessor directive yang menyertakan header file standar **stdio.h** yang berisi deklarasi fungsi standar untuk input dan output seperti **printf**, **fgets**, dan **fopen**.

2. **#include <stdlib.h>**: Ini adalah preprocessor directive yang menyertakan header file standar **stdlib.h** yang berisi deklarasi fungsi standar untuk alokasi memori dinamis, kontrol proses, dan fungsi umum lainnya seperti **exit**.
3. **#include <string.h>**: Ini adalah preprocessor directive yang menyertakan header file standar **string.h** yang berisi deklarasi fungsi standar untuk manipulasi string seperti **strlen**.
4. **#define MAX\_LENGTH 2024**: Ini mendefinisikan sebuah konstanta bernama **MAX\_LENGTH** dengan nilai 2024.
5. **#define MIN\_LENGTH 1945**: Ini mendefinisikan sebuah konstanta bernama **MIN\_LENGTH** dengan nilai 1945.
6. **void lessThanRequired(int \*lengthOfText) {...}**: Ini adalah definisi fungsi **lessThanRequired** yang mencetak pesan jika panjang teks lebih pendek dari **MIN\_LENGTH** dan mengatur **lengthOfText** ke **MIN\_LENGTH**.
7. **void equalThanRequired() {...}**: Ini adalah definisi fungsi **equalThanRequired** yang mencetak pesan jika panjang teks sama dengan **MIN\_LENGTH**.
8. **void moreThanRequired(int \*lengthOfText) {...}**: Ini adalah definisi fungsi **moreThanRequired** yang mencetak pesan jika panjang teks lebih panjang dari **MIN\_LENGTH** dan mengatur **lengthOfText** ke **MIN\_LENGTH**.
9. **int checkLengthRequirement(char\* text) {...}**: Ini adalah definisi fungsi **checkLengthRequirement** yang mengembalikan 0 jika panjang teks kurang dari **MIN\_LENGTH**, 1 jika sama, dan 2 jika lebih dari **MIN\_LENGTH**.
10. **int main() {...}**: Ini adalah definisi fungsi **main** yang merupakan titik awal eksekusi program.
11. **int lengthOfText, selectOption**;: Mendeklarasikan dua variabel **lengthOfText** dan **selectOption** bertipe **int**.
12. **void (\*functionPointer[3])(int \*) = {lessThanRequired, equalThanRequired, moreThanRequired}**;: Mendeklarasikan sebuah array bernama **functionPointer** yang berisi tiga pointer ke fungsi yang masing-masing mengambil sebuah pointer ke integer sebagai argumen.
13. **FILE \*fptr = NULL**;: Mendeklarasikan pointer **fptr** bertipe **FILE** dan menginisialisasinya dengan **NULL**.
14. **fptr = fopen("file.txt", "r")**;: Membuka file bernama "file.txt" dalam mode baca ("r") dan menetapkan alamat file tersebut ke **fptr**.
15. **if(fptr == NULL){...}**: Memeriksa apakah pembukaan file berhasil. Jika tidak, mencetak pesan kesalahan dan keluar dari program.
16. **fgets(text, MAX\_LENGTH, fptr)**;: Membaca teks dari file yang terbuka **fptr** maksimal sebanyak **MAX\_LENGTH** karakter dan menyimpannya dalam array **text**.
17. **fclose(fptr)**;: Menutup file yang terbuka.
18. **selectOption = checkLengthRequirement(text)**;: Memanggil fungsi **checkLengthRequirement** dengan teks yang terbaca dari file dan menyimpan hasilnya dalam **selectOption**.
19. **functionPointer[selectOption](&lengthOfText)**;: Memanggil fungsi yang ditunjuk oleh **functionPointer[selectOption]** dengan melewati alamat variabel **lengthOfText**.
20. **printf("\nThe Length is updated to %d\n", lengthOfText)**;: Mencetak pesan yang menyatakan bahwa panjang telah diperbarui ke nilai **lengthOfText**.
21. **return 0**;: Mengembalikan nilai 0 yang menandakan bahwa program telah berakhir dengan sukses.