

## LAPORAN

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_LENGTH 2024
#define MIN_LENGTH 1945

void kurangDariDibutuhkan(int *panjangTeks){
    printf("Panjang teks kurang dari yang ditentukan, harap perbarui teks Anda\n");
    printf("Panjang Sebelumnya: %d\n", *panjangTeks);
    *panjangTeks = MIN_LENGTH;
}

void samaDenganDibutuhkan(int *panjangTeks){
    printf("Terima kasih, panjang teks Anda sudah sesuai\n");
}

void lebihDariDibutuhkan(int *panjangTeks){
    printf("Teks Anda terlalu panjang, harap kurangi teksnya\n");
    printf("Panjang Sebelumnya: %d\n", *panjangTeks);
    *panjangTeks = MIN_LENGTH;
}

int periksaKebutuhanPanjang(char* teks){
    int panjang = strlen(teks);
    if (panjang < MIN_LENGTH)
        return 0;
    else if (panjang == MIN_LENGTH)
        return 1;
    else
        return 2;
}

int main() {
    int panjangTeks, pilihan;
    FILE *fptr = NULL;
    char teks[MAX_LENGTH];

    fptr = fopen("file.txt", "r");

    if(fptr == NULL){
        printf("Error: Tidak dapat membuka file\n");
        exit(1);
    }
}
```

```

fgets(teks, MAX_LENGTH, fptr);

fclose(fptr);

pilihan = periksaKebutuhanPanjang(teks);

void (*fungsi[3])(int *) = {kurangDariDibutuhkan, samaDenganDibutuhkan,
lebihDariDibutuhkan};
fungsi[pilihan](&panjangTeks);

printf("\nPanjang Telah Diperbarui Menjadi %d", panjangTeks);

return 0;
}

```

## PENJELASAN

Program ini adalah contoh sederhana dari sebuah aplikasi yang membaca teks dari sebuah file dan kemudian memeriksa panjang teks tersebut. Program kemudian memberikan tanggapan berdasarkan panjang teks apakah sesuai dengan kebutuhan yang telah ditentukan.

program ini menggunakan tiga library standar untuk operasi I/O, alokasi memori, dan manipulasi string: `stdio.h`, `stdlib.h`, dan `string.h`.

`MAX_LENGTH` dan `MIN_LENGTH` didefinisikan sebagai konstanta dengan nilai 2024 dan 1945 masing-masing. Konstan ini menentukan panjang maksimum dan minimum yang diizinkan untuk teks.

Empat fungsi dideklarasikan di bagian ini.

`kurangDariDibutuhkan`, `samaDenganDibutuhkan`, dan `lebihDariDibutuhkan` bertanggung jawab untuk memberikan respons tergantung pada panjang teks.

`periksaKebutuhanPanjang` digunakan untuk memeriksa panjang teks yang diberikan.

Fungsi `main` dimulai dengan mendeklarasikan beberapa variabel, termasuk `panjangTeks`, `pilihan`, dan `teks`.

Program membuka file "file.txt" untuk dibaca. Jika tidak dapat membuka file, program menampilkan pesan kesalahan dan keluar.

Fungsi `fgets` digunakan untuk membaca teks dari file ke array teks.

Setelah membaca teks, file ditutup.

Fungsi `periksaKebutuhanPanjang` dipanggil untuk menentukan apakah panjang teks sesuai dengan kebutuhan.

Sebuah array dari pointer ke fungsi `void` yang menerima `int *` (pointer ke integer) dideklarasikan di baris berikutnya. Ini digunakan untuk memilih fungsi respons yang tepat berdasarkan pilihan yang dihasilkan oleh `periksaKebutuhanPanjang`.

Fungsi respons dipanggil sesuai dengan pilihan dan `panjangTeks` dikirim sebagai parameter.

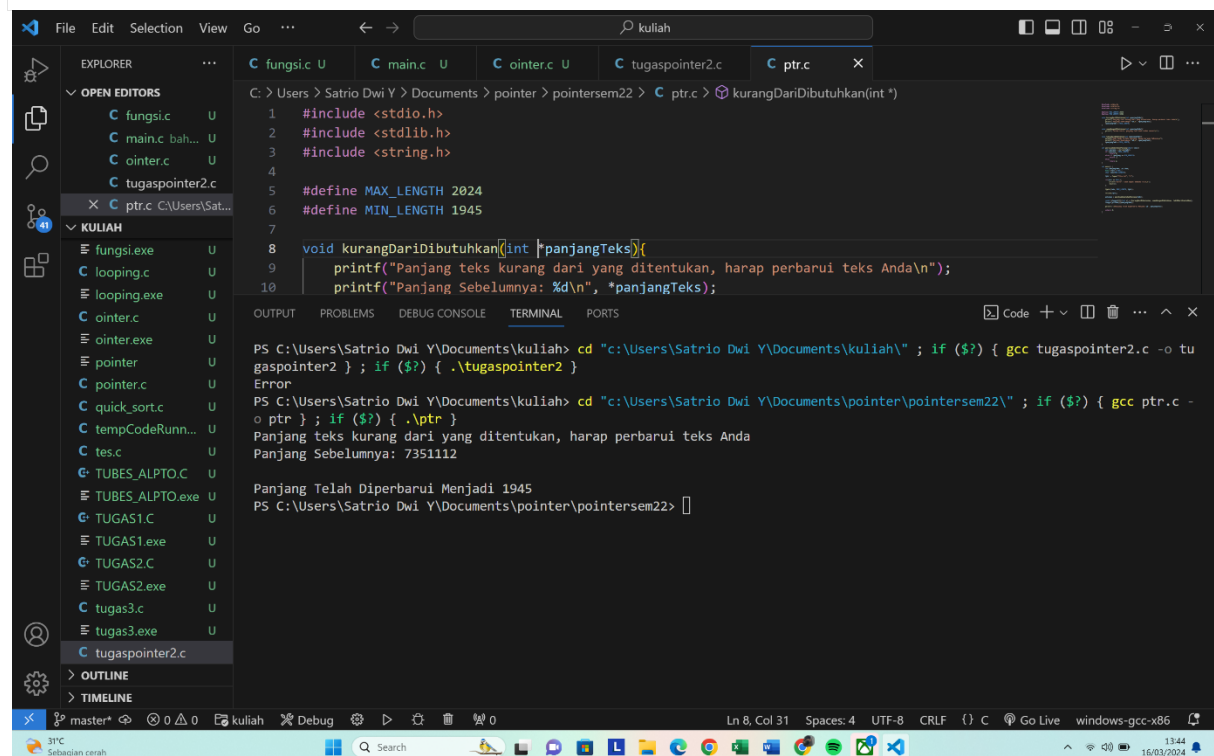
Terakhir, panjang yang diperbarui dari teks dicetak ke layar.

Fungsi `periksaKebutuhanPanjang` mengembalikan nilai 0 jika panjang teks kurang dari `MIN_LENGTH`, nilai 1 jika panjangnya sama dengan `MIN_LENGTH`, dan nilai 2 jika panjangnya lebih dari `MIN_LENGTH`.

Pointer `panjangTeks` digunakan untuk memperbarui panjang teks jika perlu dalam fungsi respons.

Program ini menggunakan pointer ke fungsi untuk memilih fungsi respons berdasarkan nilai yang dihasilkan oleh `periksaKebutuhanPanjang`.

Keseluruhan, program ini membaca teks dari file, memeriksa panjangnya, dan memberikan respons sesuai dengan kebutuhan panjang yang telah ditetapkan.



The screenshot shows the Visual Studio Code interface with a C program open in the editor. The program is located at `C:\Users\Satrio Dwi Y\Documents\pointer\pointersem22\ptr.c`. The code includes `stdio.h`, `stdlib.h`, and `string.h`. It defines `MAX_LENGTH` as 2024 and `MIN_LENGTH` as 1945. The function `kurangDariDibutuhkan` takes an integer pointer `panjangTeks` and prints a message if the text length is less than `MIN_LENGTH`, then updates the length to 1945. The terminal shows the command to compile and run the program, and the output displays the current length (7351112) and the updated length (1945).

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_LENGTH 2024
#define MIN_LENGTH 1945

void kurangDariDibutuhkan(int *panjangTeks){
    printf("Panjang teks kurang dari yang ditentukan, harap perbarui teks Anda\n");
    printf("Panjang Sebelumnya: %d\n", *panjangTeks);
}
```

PS C:\Users\Satrio Dwi Y\Documents\kuliah> cd "C:\Users\Satrio Dwi Y\Documents\kuliah\" ; if (\$?) { gcc tugaspointer2.c -o tugaspointer2 } ; if (\$?) { .\tugaspointer2 }  
Error  
PS C:\Users\Satrio Dwi Y\Documents\kuliah> cd "C:\Users\Satrio Dwi Y\Documents\pointer\pointersem22\" ; if (\$?) { gcc ptr.c -o ptr } ; if (\$?) { .\ptr }  
Panjang teks kurang dari yang ditentukan, harap perbarui teks Anda  
Panjang Sebelumnya: 7351112  
  
Panjang Telah Diperbarui Menjadi 1945  
PS C:\Users\Satrio Dwi Y\Documents\pointer\pointersem22>