**Program Studi Teknik Elektro ITB** Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)

Tahun / Semester : 2023-2024 / Genap

**Modul** : 3 – Strings and External Files

**Nama Asisten / NIM** : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Nama Praktikan / NIM** : Pradigta Hisyam Ramadhan / 18322008

**Tugas Pendahuluan**

|  |
| --- |
| **1.** Bahasa pemrograman C tidak mendukung deklarasi untuk tipe data string. Oleh karena itu, string harus dideklarasikan dengan tipe data *char* dan dipandang sebagai *array of characters* [1].  **2.** Header **string.h** merupakan salah satu header standar dalam bahasa pemrogranan C yang digunakan untuk melakukan berbagai macam operasi manipulasi string (*array of characters*) [2]. Beberapa fungsi yang ada di dalam header string.h adalah:  a.) strlen() : Merupakan fungsi yang digunakan untuk menghitung panjang dari string dalam bentuk integer [3].  b.) strcpy() / strncpy() : Merupakan salah satu fungsi yang digunakan untuk menyalin suatu ke string ke variabel string yang lain [4].  c.) strcat() / strncat() : Merupakan fungsi yang digunakan untuk menggabungkan dua string (konkatenasi) / n-karakter pertama dari variabel ke variabel lainnya [5].  d.) strcmp() / strncmp() : Merupakan fungsi yang digunakan untuk membandingkan dua string / n-karakter partama dari dua string [6].  e.) strtok() : Merupakan fungsi yang digunakan untuk memisahkan sebuah string berdasarkan beberapa karakter sebagai *delimiter*.  **3.** Mode ‘r’ pada fungsi fopen() digunakan untuk **membuka & membaca file yang sudah ada**.  Sedangkan mode ‘w’ pada fungsi fopen() digunakan untuk **membuka & menulis isi file**. Jika nama file belum ada, maka komputer akan membuat sebuah file baru.  **Contoh penggunaan fungsi fopen dengan mode ‘r’:**  #include <stdio.h>  #include <string.h>  #include <stdlib.h>  #define MAX\_LEN 255  int main(void){      FILE \*file1Pointer;      char buffer[MAX\_LEN];    // Variabel sementara untuk menyimpan pembacaan baris string      // Membuka file bernama example.txt      file1Pointer = fopen("tp03-soal3.txt", "r");      // Cek apakah file ada dan berhasil dibuka      if (file1Pointer == NULL){          printf("file tidak ada.\n");          return 1;      }      else{          printf("File berhasil dibuka dengan mode 'r'.\n\n");      }      // Membaca tiap baris pada file menggunakan fungsi fgets      while (fgets(buffer, sizeof(buffer), file1Pointer)){          // Menampilkan tiap baris yang dibaca pada file          printf("Baris yang dibaca pada baris: %s", buffer);      }      fclose(file1Pointer);      return 0;  }  **Output yang dihasilkan apabila file “tp03-soal3.txt” tidak ada:**  >> file tidak ada.  **Output yang dihasilkan apabila file “tp03-soal3.txt” ada:**  >> File berhasil dibuka dengan mode 'r'.  >> Baris yang dibaca pada baris: Pradigta Hisyam Ramadhan  >> Baris yang dibaca pada baris: 18322008  >> Baris yang dibaca pada baris: Teknik Biomedis  >> Baris yang dibaca pada baris: Institut Teknologi Bandung  **Contoh penggunaan fungsi fopen() dengan mode ‘w’:**  #include <stdio.h>  #include <string.h>  #include <stdlib.h>  #define MAX\_LEN 255  int main(void){      FILE \*file1Pointer;      // Deklarasi string yang akan ditulis ke dalam file      char content[] = "Pradigta Hisyam Ramadhan\n18322008\nTeknik Biomedis\nTugas Pendahuluan Modul 3: Strings and External Files";      // Membuka file bernama example.txt      file1Pointer = fopen("tp03-soal3.txt", "w");      // Cek apakah file ada dan berhasil dibuka      if (file1Pointer == NULL){          printf("file tidak ada.\n");          return 1;      }      else{          printf("File berhasil dibuka dengan mode 'w'.\n");      }      // Menuliskan konten string ke dalam file menggunakan fungsi fputs      if (fputs(content, file1Pointer) == EOF){          printf("Terjadi error saat menuliskan konten ke file.\n");          fclose(file1Pointer);          return 1;      }      printf("Konten berhasil ditulis ke dalam file.\n");      fclose(file1Pointer);      return 0;  }  **Output yang ditampilkan pada terminal apabila konten berhasil ditulis ke dalam file:**  >> File berhasil dibuka dengan mode 'w'.  >> Konten berhasil ditulis ke dalam file.  **4.** Empat fungsi operasi pengolahan file eksternal pada bahasa pemrograman C adalah:  a.) **Membuka file**: fopen()  b.) **Menulis file**: fputc() untuk menulis tipe data *int* dan fputs() untuk menulis tipe data *array of characters*  c.) **Membaca file**: fgetc() untuk pembacaan file berdasarkan karakter dan fgets() untuk pembacaan file berdasarkan string.  d.) **Menutup file:** fclose() yang akan memberikan nilai EOF jika terjadi error saat menutup sebuah file. |
| **Soal Pemrograman**  /\*EL2208 Praktikum Pemecahan Masalah dengan C 2023/2024  \*TP Modul         : 3  \*Hari dan Tanggal : Jumat, 15 Maret 2024  \*Nama (NIM)       : Pradigta Hisyam Ramadhan (18322008)  \*Nama File        : tp-03-soalPemrograman.c  \*Deskripsi        : Program digunakan untuk menampilkan nama dari sebuah file web.txt  \*/  #include <stdio.h>  #include <stdlib.h>  #include <string.h>  #define MAX\_LINE 255  int main(void) {      // Deklarasi variabel      FILE \*filePointer;      char buffer[MAX\_LINE];      // membuka file web.txt      filePointer = fopen("web.txt", "r");      // Cek apakah file ada dan dapat dibuka      if (filePointer == NULL) {          printf("Folder tidak ada atau tidak dapat dibuka!\n");          return 1;      }      // File ada dan berhasil dibuka      printf("Folder berhasil dibuka!\n\n");      // Menampilkan string pada tiap baris file dengan urutan indeks yang terbalik      printf("Web praktikum:\n");      // Deklarasi variabel untuk menyimpan array of strings dan panjang baris.      char lines[MAX\_LINE][100];      int lineCount = 0;  // Pada akhir iterasi, nilai lineCount akan sama dengan panjang baris      // Membaca semua baris dan simpan nilainya ke dalam array      while (fgets(buffer, sizeof(buffer), filePointer) != NULL && lineCount < MAX\_LINE) {          // Hapus karakter newline pada file          int len = strlen(buffer);          if (len > 0 && buffer[len - 1] == '\n') {              buffer[len - 1] = '\0';          }          // Salin string pada baris ke dalam array lines          strcpy(lines[lineCount], buffer);          lineCount++;      }      // Hasil pembacaan file dengan indeks terbalik      for (int i = lineCount - 1; i >= 0; i--) {          // Tampilkan string pada file dengan titik, kecuali pada indeks terakhir yang diproses          printf("%s%s", lines[i], (i == 0) ? "" : ".");      }      // Tutup file setelah selesai      fclose(filePointer);      return 0;  } |

**Referensi**

[1] A. Beri, “C Strings | Declaring Strings in C - Scaler Topics,” Scaler Topics, Feb. 14, 2022. https://www.scaler.com/topics/c/c-string-declaration/ (accessed Mar. 11, 2024).

[2]GfG, “C Library    string.h,” GeeksforGeeks, Mar. 15, 2023. https://www.geeksforgeeks.org/c-library-string-h/ (accessed Mar. 11, 2024).

[3]GfG, “strlen() function in c,” GeeksforGeeks, May 29, 2023. https://www.geeksforgeeks.org/strlen-function-in-c/ (accessed Mar. 11, 2024).

[4]GfG, “strcpy in C,” GeeksforGeeks, Jan. 20, 2023. https://www.geeksforgeeks.org/cpp-strcpy/ (accessed Mar. 11, 2024).

[5]GfG, “strcat() in C,” GeeksforGeeks, Mar. 11, 2023. https://www.geeksforgeeks.org/strcat-in-c/ (accessed Mar. 11, 2024).

[6]GfG, “C strcmp(),” GeeksforGeeks, Jun. 11, 2023. https://www.geeksforgeeks.org/strcmp-in-c/ (accessed Mar. 11, 2024).