



Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)
Tahun / Semester : 2023-2024 / Genap
Modul : 5 – Structure & Dynamic Array
Nama Asisten / NIM : _____
Nama Praktikan / NIM : Pradigta Hisyam Ramadhan / 18322008

Tugas Pendahuluan

1. Definisi *structure (struct)* pada C

Struktur adalah turunan dari tipe data yang memungkinkan kita untuk mendefinisikan berbagai macam variabel yang saling terkait satu sama lain, dalam satu nama.

Hal baru yang bisa diimplementasikan menggunakan struct dan tidak bisa diimplementasikan hanya dengan tipe data pada modul lain:

- Mengimplementasi struktur data khusus: *Linked list*, *queue*, *stack*, dan *tree*
- Membuat semacam array yang memiliki berbagai macam tipe data (tidak homogen) sehingga data dapat diakses dan dimanipulasi dengan lebih mudah.
- Sebagai dasar dari paradigma pemrograman berorientasi objek (OOP)

Hal yang diimplementasikan dengan lebih mudah menggunakan *structure* daripada menggunakan implementasi dari modul-modul sebelumnya

- Deklarasi sekumpulan array dari suatu objek dengan berbagai macam tipe data yang saling terkait

2. Perbedaan array statis dan dinamis:

Array statis: Pada jenis array statis, memori yang dialokasikan memiliki ukuran yang tetap, sehingga kita tidak dapat mengubah atau memperbaharui ukuran dari array.

Array dinamis: Pada jenis array dinamis, memori untuk array dialokasikan saat program berjalan, namun ukurannya belum pasti.

Kapan kita menggunakan array statis?

Ketika kita sudah mengetahui ukuran array yang ingin kita deklarasikan sebelumnya.

Kapan kita menggunakan array dinamis?

Ketika kita tidak mengetahui ukuran array, sehingga ukurannya dapat ditentukan dari input pengguna alih-alih dideklarasikan dalam program.

3. Implementasi array dinamis tanpa menggunakan fungsi *malloc()* atau *calloc()* tidak dimungkinkan karena kita perlu mengalokasikan memori array sebelum program dijalankan. Jika memori tidak dialokasikan, maka array tidak dapat melakukan "pemesanan" memori pada program.

Berbeda dengan fungsi *free()*, jika program tidak menggunakan fungsi tersebut, maka memori tidak akan dibebaskan saat program berakhir. Pada program yang berjalan dalam jangka waktu yang lama, penumpukan memori yang tidak dibebaskan dapat menyebabkan terjadinya error pada komputer.

Soal Pemrograman

```
/*EL2208 Praktikum Pemecahan Masalah dengan C 2023/2024
*TP Modul      : 5 - Structure & Dynamic Array
*Hari dan Tanggal : Senin, 1 April 2024
*Nama (NIM)     : Pradigta Hisyam Ramadhan (18322008)
>Nama File      : tp-m05.c
*Deskripsi     : Program untuk melakukan implementasi autograder
*/

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

// Membuat node
struct Praktikan{
    char *nama;
    int NIM;
    int pemilihanSoal;
    int test1;
    int test2;
    int test3;
    int test4;
    int test5;
};

void addPraktikan(struct Praktikan **praktikan, int *total_praktikan){
    // realokasi memori struktur dari total sebelumnya
    *praktikan = realloc(*praktikan, (*total_praktikan + 1) * sizeof(struct
Praktikan));
    if (*praktikan == NULL){
        printf("Alokasi memori gagal.\n");
        exit(1);
    }

    struct Praktikan praktikan_baru;
    // Alokasi memori untuk nama praktikan, asumsi nama tidak pernah lebih dari
50 karakter
    praktikan_baru.nama = (char*)malloc(50 * sizeof(char));
    if(praktikan_baru.nama == NULL){
        printf("Alokasi memori gagal.\n");
        exit(1);
    }

    // Mendapatkan nama praktikan
    printf("\nMasukkan nama praktikan: ");
    getchar(); // Menghilangkan karakter newline dari input sebelumnya
    fgets(praktikan_baru.nama, 50, stdin);
    praktikan_baru.nama[strcspn(praktikan_baru.nama, "\n")] = '\0'; // menghapus
karakter newline dari input nama

    // Mendapatkan NIM praktikan
```

```

printf("Masukkan NIM praktikan: ");
scanf("%d", &praktikan_baru.NIM);

// Mendapatkan pemilihan soal praktikan
printf("Masukkan pilihan soal praktikan: ");
scanf("%d", &praktikan_baru.pemilihanSoal);

// Mendapatkan hasil testcase praktikan
printf("Masukkan pilihan hasil testcase praktikan secara berturut-turut
(skala 0-10): ");
scanf("%d %d %d %d %d", &praktikan_baru.test1, &praktikan_baru.test2,
&praktikan_baru.test3, &praktikan_baru.test4, &praktikan_baru.test5);

// Masukkan data ke dalam list
(*praktikan)[*total_praktikan] = praktikan_baru;
(*total_praktikan)++;
}

void calculateGrade(struct Praktikan *praktikan, int total_praktikan, int
search_NIM){
    int found = 0;
    int total_value = 0;
    for (int i = 0; i < total_praktikan; i++){
        if (praktikan[i].NIM == search_NIM){
            printf("Praktikan ditemukan!\n");
            total_value += praktikan[i].test1 + praktikan[i].test2 +
praktikan[i].test3 + praktikan[i].test4 + praktikan[i].test5;
            printf("Total nilai hasil testcase praktikan: %d/50.\n",
total_value);
            found = 1;
            break;
        }
    }
    if (!found){
        printf("Praktikum dengan NIM %d ngga ada nih :(\n", search_NIM);
    }
}

int main(void){
    struct Praktikan *praktikan = NULL;
    int total_praktikan = 0;
    int opsi, search_NIM;

    do
    {
        printf("\nAutograder baru ni bos, kuy dipilih opsinya\n");
        printf("1. Tambah praktikan baru\n");
        printf("2. Hitung nilai praktikan\n");
        printf("3. Exit\n");
        printf("Masukkan pilihan: ");

```

```

scanf("%d", &opsi);

switch (opsi){
    case 1:
        addPraktikan(&praktikan, &total_praktikan);
        break;
    case 2:
        printf("\nMasukkan NIM praktikan yang ingin dihitung nilainya:");
        scanf("%d", &search_NIM);
        calculateGrade(praktikan, total_praktikan, search_NIM);
        break;
    case 3:
        printf("\nProgram selesai, yay.\n");
        break;
    default:
        printf("\nNgga ada opsi itu, bro.\n");
}
} while (opsi != 3);

// Bebaskan memori pada array dinamis string dan struktur
for(int i = 0; i < total_praktikan; i++){
    free(praktikan[i].nama);
}
free(praktikan);

return 0;
}

```

Referensi

C - Structures. (n.d.). Retrieved April 1, 2024, from https://www.tutorialspoint.com/cprogramming/c_structures.htm

C Structures (structs). (n.d.). Retrieved April 1, 2024, from https://www.w3schools.com/c/c_structs.php

G. (2023, December 20). *Difference between Static Arrays and Dynamic Arrays*. GeeksforGeeks. Retrieved April 1, 2024, from <https://www.geeksforgeeks.org/difference-between-static-arrays-and-dynamic-arrays/>

G. (2021, October 5). *what happens when you don t free memory after using malloc()*. GeeksforGeeks. Retrieved April 1, 2024, from <https://www.geeksforgeeks.org/what-happens-when-you-dont-free-memory-after-using-malloc/>

G. (2023, January 11). *Dynamic Array in C*. GeeksforGeeks. Retrieved April 1, 2024, from <https://www.geeksforgeeks.org/dynamic-array-in-c/>

