



# **Modul Praktikum**

## Pemrograman Lanjut

**Laboratium Digital**



**Tim Penyusun Modul**

Tim Asisten Lab Digital 2021

Program Studi S1 Teknik Komputer

Fakultas Teknik

Universitas Indonesia

2021

# KATA PENGANTAR

Praktikum ini adalah bagian dari mata kuliah Pemrograman Lanjut dan Praktikum yang diberikan untuk mahasiswa Semester 2 Program Studi Teknik Komputer. Mata kuliah ini merupakan mata kuliah lanjutan yang diberikan kepada mahasiswa dari serangkaian mata kuliah untuk mendukung 2 CP Prodi, yaitu Mampu merancang algoritma untuk masalah tertentu dan mengimplementasikannya ke dalam pemrograman (C6) dan Mampu memanfaatkan teknologi informasi dan komunikasi (C3).

Capaian Pembelajaran Mata Kuliah (CPMK) dari mata kuliah ini adalah:

1. Mampu merancang program komputer prosedural kompleks dengan struktur data dinamis (C6)
2. Mampu menunjukkan sikap kritis, kreatif, dan inovatif dan menghargai orang lain dalam kelompok untuk memecahkan masalah bersama dan tugas kelompok Pemrograman Lanjut (C3, A3)
3. Mampu menggunakan software pemrogram komputer dengan mahir (C3)

Sedangkan Sub-CPMK yang akan dicapai adalah:

- 1.1. Mampu mengimplementasikan algoritma rekursif ke dalam pemrograman
- 1.2. Mampu mengimplementasikan algoritma searching dan sorting ke dalam pemrograman
- 1.3. Mampu membuat program komputer prosedural kompleks dengan linked list, stack dan queue
- 1.4. Mampu mengimplementasikan multi-threading dan parallel programming
- 1.5. Mampu merancang perangkat lunak sederhana dengan struktur data dinamis
- 2.1. Mampu menunjukkan proses berpikir kritis, kreatif dan inovatif dalam menyelesaikan permasalahan kelompok
- 2.2. Mampu berkomunikasi dengan sopan
- 2.3. Mampu menghargai pendapat orang lain
- 3.1. Mampu menggunakan software pemrogram komputer untuk program kompleks dengan mahir

Pada mata kuliah ini mahasiswa akan mengasah kemampuan cara berpikir dan penyelesaian masalah dengan membuat algoritma, kemudian menerjemahkan algoritma tersebut ke dalam bahasa pemrograman yang dapat dijalankan oleh komputer. Pada mata kuliah ini mahasiswa akan mempelajari Rekursif, Searching, Sorting, Linked list, Stack, Queue, Multi-threading, Parallel programming,, dan pada bagian akhir akan ditutup dengan proyek akhir pemrograman yang dibuat oleh mahasiswa.

Bahasa C merupakan bahasa pemrograman terstruktur, yang membagi program dalam sejumlah blok. Tujuannya adalah untuk mempermudah dalam pembuatan dan pengembangan program. Bahasa C menggunakan standarisasi ANSI (American National Standardization Institute) yang dijadikan acuan oleh para pembuat compiler C. Bahasa C terdiri dari fungsi-fungsi dan setiap program C memiliki fungsi utama yang disebut main. Program akan dieksekusi dimulai dari statement pertama pada fungsi main tersebut.

Akhir kata, diharapkan modul praktikum ini akan dapat menjadi referensi untuk membuat program dalam Bahasa C secara umum, dan menjadi panduan dalam menjalankan praktikum mata kuliah Pemrograman Lanjut, secara khusus.

Depok, Februari 2021

Tim Penyusun Modul

# MODUL 7: LINKED LIST

## Definisi

Linked list merupakan struktur data yang terdiri sekumpulan node yang secara bantu-membantu merepresentasikan sebuah urutan. Linked list juga disebut sebuah metode untuk menyimpan data dengan struktur sehingga sanggup secara otomatis membuat suatu daerah di memory untuk menyimpan data yang dibutuhkan. Menurut cara pengisian dan pengambilan elemennya, linked list dibagi menjadi 2, yaitu:

- Stack (Tumpukan)
- Queue (Antrian)

Linked list juga dapat diimplementasikan untuk membuat beberapa algoritma, salah satunya adalah Tree (Pohon)

## Dynamic Memory Allocation

Untuk membuat linked list, diperlukan dynamic memory allocation untuk mengalokasikan memori secara manual selama program berlangsung. Pada dasarnya, cara kerja dynamic memory allocation mirip seperti array. Perbedaannya adalah dengan mengalokasikan memori secara dinamis, kita tidak perlu mendeklarasikan ukuran array diawal. Oleh karena itu, ukuran arraynya dinamis dan lebih menghemat memori.

Untuk mengalokasikan memori secara dinamis, fungsi library yang biasa digunakan adalah malloc (), calloc (), realloc (), dan free (). Fungsi-fungsi ini terdapat dalam file header <stdlib.h>.

- Malloc()

Nama “malloc” berasal dari memory allocation. Fungsi malloc() menyediakan satu blok memori dengan jumlah byte yang ditentukan. Kemudian fungsi ini akan mengembalikan dalam bentuk pointer of void yang dapat di cast ke pointer dalam bentuk apapun. Syntax dari fungsi malloc() adalah:

```
ptr = (castType*) malloc(size);
```

- Calloc()

Nama “calloc” berasal dari contiguous allocation. Perbedaannya dengan malloc adalah, fungsi malloc () mengalokasikan memori dan membiarkan memori tidak diinisialisasi. Sedangkan, fungsi calloc () mengalokasikan memori dan menginisialisasi semua bit ke nol. Syntax dari fungsi calloc() adalah:

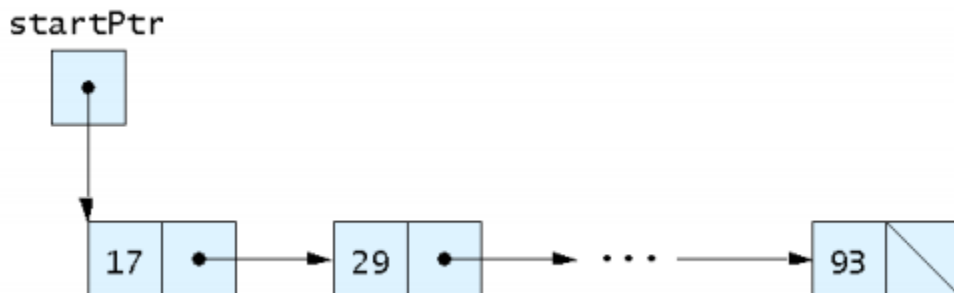
```
ptr = (castType*)calloc(n, size);
```

- Free()

Memori yang dialokasikan secara dinamis yang dibuat dengan calloc() atau malloc() tidak bisa dihapus secara otomatis Kita harus menggunakan fungsi free() untuk menghapus memori yang telah dialokasikan. Syntax dari fungsi free() adalah:

```
free(ptr);
```

## Linked List



Gambar diatas adalah ilustrasi dari linked list. Kita mempunyai beberapa struktur node, yang masing-masing menyimpan data. Pointer startPtr diawal berisi alamat dari node pertama, yang bernilai 17. Setelah itu, pointer yang terdapat pada node pertama berisi alamat dari node kedua, yg bernilai 29. Begitu juga seterusnya, pointer yang ada pada node kedua berisi alamat dari node berikutnya, hingga pointer yang ada pada node terakhir (yang bernilai 93) akan berisi NULL. Hal ini menunjukkan bahwa struktur node yang berisi nilai nextPtr NULL tersebut berada di akhir list.

Langkah dasar dari pembuatan linked list:

1. Buat struktur data untuk sebuah node (terdiri atas field untuk menyimpan data dan alamat/pointer ke node berikutnya)
2. Buat variabel yang merupakan pointer untuk alamat node pertama
3. Saat list kosong, alamat node pertama adalah NULL
4. Alokasikan memori untuk sebuah node, dan simpan pointer-nya
5. Pada node yang dirujuk dengan pointer tersebut, masukkan data, dan pointer ke node berikutnya (NULL)
6. Simpan pointer node ini ke pointer alamat pertama.

Implementasi program:

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

struct node { //langkah 1
    int data;
    struct node * next;
};

int main(){
    struct node * headAddress = NULL; // langkah 2 & 3
    struct node * tmpAddress;
    struct node * iteratorPtr;
    struct node * prevPtr;

    //membuat node pertama
    tmpAddress = malloc(sizeof(struct node)); //langkah 4
    (*tmpAddress).data = 17;
    (*tmpAddress).next = NULL; // langkah 5

    headAddress = tmpAddress; // Langkah 6

    //tambahkan 1 node di depan
    tmpAddress = malloc(sizeof(struct node)); //membuat alamat node
baru
    (*tmpAddress).data = 29;
    (*tmpAddress).next = headAddress;

    headAddress = tmpAddress;

    //tambahkan 1 node di belakang
    tmpAddress = malloc(sizeof(struct node)); //membuat alamat node
baru
    (*tmpAddress).data = 93;
    (*tmpAddress).next = NULL;

    iteratorPtr = headAddress; //iteratorPtr adalah alamat yg akan
```


```

dicek
    while(1){
        if((*iteratorPtr).next == NULL){ //iteratorPtr merujuk ke
node paling akhir
            (*iteratorPtr).next = tmpAddress;
            break;
        }else{
            //alamat utk dicek berikutnya adalah alamat yg
tersimpan di kolom next di rumah di alamat sekarang
            iteratorPtr = (*iteratorPtr).next;
        }
    }

//Print List
if(headAddress != NULL){
    printf("List berisi:\n");
    iteratorPtr = headAddress;
    while(iteratorPtr != NULL){
        printf(" [%d] : %d\n",iteratorPtr,
(*iteratorPtr).data);
        iteratorPtr = (*iteratorPtr).next;
    }
}else{
    printf("List kosong\n");
}
return 0;
}

```

Output Programnya adalah :

 C:\Users\Dar\Desktop\Linked list.exe

```

List berisi:
[1643520] : 29
[1643488] : 17
[1643552] : 93

```