

## MODUL 6: STACK (TUMPUKAN)

---

### 1.1. Deskripsi Singkat

Stack atau tumpukan adalah suatu struktur data berbentuk list atau daftar yang proses penambahan atau penghapusan datanya hanya melalui satu posisi yaitu item teratasnya atau biasa disebut top. Tumpukan dapat diimplementasikan dengan array atau linked list.

### 1.2. Tujuan Praktikum

- 1) Mahasiswa mampu mengimplementasikan berbagai persoalan tumpukan dengan menggunakan bahasa C
- 2) Mahasiswa dapat melakukan operasi penyisipan (push) maupun penghapusan (pop) item dari tumpukan dengan menggunakan bahasa C

### 1.3. Material Praktikum

Kegiatan pada modul ini memerlukan material berupa software editor dan compiler (atau IDE) untuk bahasa pemrograman C.

### 1.4. Alokasi Waktu

- Tatap muka di kelas (wajib): 50 menit
- Kerja Mandiri: 120 menit pengerjaan mandiri

### 1.5. Kegiatan Praktikum

Pada praktikum kali ini, kita akan membuat sebuah program konversi bilangan desimal ke biner dengan menggunakan tumpukan yang diimplementasikan dengan menggunakan array satu dimensi. Dalam proses pembuatannya, ada 4 tahap, yaitu:

1. Persiapan,
2. Membuat fungsi push,
3. Membuat fungsi pop,
4. Finalisasi.

#### A. Persiapan

Dalam tahapan ini kita harus memahami bagaimana cara mengkonversi bilangan desimal ke biner. Dalam modul 6 ini, kita akan menggunakan algoritma yang sudah umum dalam mengkonversi bilangan desimal ke biner, yaitu bilangan desimal yang akan dikonversi dibagi 2 secara berulang sampai hasil baginya 0.

Contoh 1: Konversi bilangan desimal nilai 50 menjadi bilangan biner.

$$50/2 = 25 \text{ sisa bagi adalah } 0$$

$$25/2 = 12 \text{ sisa bagi adalah } 1$$

$12/2 = 6$  sisa bagi adalah 0

$6/2 = 3$  sisa bagi adalah 0

$3/2 = 1$  sisa bagi adalah 1

$1/2 = 0$  sisa bagi adalah 1

Hasil pembagian tersebut kemudian diurutkan dari yang paling akhir hingga paling awal menjadi 110010.

Jadi Hasil Konversi bilangan desimal 50 menjadi bilangan biner adalah 110010.

Contoh 2: Konversi bilangan desimal 105 menjadi bilangan biner.

$105/2 = 52$  sisa bagi adalah 1

$52/2 = 26$  sisa bagi adalah 0

$26/2 = 13$  sisa bagi adalah 0

$13/2 = 6$  sisa bagi adalah 1

$6/2 = 3$  sisa bagi adalah 0

$3/2 = 1$  sisa bagi adalah 1

$1/2 = 0$  sisa bagi adalah 1

Hasil pembagian tersebut kemudian diurutkan dari yang paling akhir hingga paling awal menjadi 1101001.

Jadi Hasil Konversi bilangan desimal 105 menjadi bilangan biner adalah 1101001.

Menggunakan tumpukan, setiap kali bilangan desimal dibagi dengan 2, sisa bagi disimpan ke dalam tumpukan (push). Setelah selesai, tumpukan di hapus itemnya satu persatu (pop) dan ditampilkan. Itulah hasil konversi bilangan desimal ke biner. Setelah memahami algoritmanya, maka perlu dipersiapkan struktur data tumpukan serta fungsi lain yang akan digunakan dalam proses berikutnya.

```

#include<stdio.h>

typedef struct
{
    int item[50];
    int jml_item;
} stack;

//menyiapkan tumpukan kosong
void initializestack(stack *s)
{
    s->jml_item = 0;
}

//jika tumpukan kosong maka nilai fungsinya 1 (true),
//jika tidak 0 (false)
int isEmpty(stack *s)
{
    return (s->jml_item == 0);
}

//jika tumpukan sudah full(dlm deklarasi stack, field item
//adalah array dg jumlah elemen sebanyak 50)maka nilai
//nilai fungsinya 1, jika tidak 0
int isFull(stack *s)
{
    return (s->jml_item == 50);
}

```

sampai dengan potongan program di atas, kita sudah menyiapkan struktur dari tumpukan dan beberapa fungsi yang nanti akan digunakan dalam proses selanjutnya.

ketik potongan program tersebut pada IDE Anda, lalu simpan dengan nama **praktikum6a.c**.

## B. Pembuatan Fungsi push

Untuk membuat fungsi push, ada dua kondisi yang harus diperhatikan. Kondisi pertama yaitu saat tumpukan sudah penuh maka artinya bilangan yang diinputkan terlalu besar (lebih dari 49 digit bilangan biner). Saat kondisi penuh, fungsi tidak menambahkan item ke dalam tumpukan dan menampilkan pesan bahwa bilangan terlalu besar. Sedangkan kondisi kedua adalah tumpukan belum penuh. Saat seperti ini, fungsi menambahkan item dan jml\_item bertambah 1.

```

void push(int x, stack *s)
{
    if(isFull(&s))
        printf("Bilangan terlalu besar !\n");
    else {
        s->item[s->jml_item]=x; ++(s->jml_item);
    }
}

```

### C. Pembuatan Fungsi pop

Fungsi pop adalah fungsi yang prosesnya berlawanan dengan fungsi push. Seperti fungsi push, fungsi ini juga harus memperhatikan 2 kondisi. Pertama saat tumpukan kosong, maka fungsi akan mengembalikan nilai 0. Kedua saat tumpukan tidak kosong. Dalam kondisi seperti ini, fungsi akan mengembalikan sesuai nilai item dan jml\_item berkurang 1.

```

int pop(stack *s)
{
    if(isEmpty(&s)) return (0);
    else {
        --(s->jml_item);
        return (s->item[s->jml_item]);
    }
}

```

### D. Finalisasi

Langkah terakhir adalah mengatur fungsi-fungsi tersebut sehingga tumpukan bisa menyimpan hasil bagi bilangan yang ingin dikonversi. Setelah itu tumpukan di pop satu persatu itemnya yang merupakan hasil konversi bilangan desimal tersebut. Fungsi akan mengembalikan nilai 0 saat pengguna menginputkan nilai 0 untuk bilangan desimal yang ingin dikonversi.

```

void main() {
    stack tumpukan;
    int bil_desimal;
    int sisa_bagi;

```

```

printf("Program konversi Desimal ke Biner\n\n");
initializestack(&tumpukan);
printf("Masukkan bilangan desimal = ");
scanf("%d", &bil_desimal);

if (bil_desimal==0) printf("Hasil konversi ke biner = 0");
else{
    int n;
    for(n=bil_desimal;n>0;n=n/2){
        sisa_bagi=n%2;
        push(sisa_bagi,&tumpukan);
    }

    int i;
    printf("Hasil konversi ke biner = ");
    for(i=tumpukan.jml_item;i>0;i--){
        printf("%d", pop(&tumpukan));
    }
}
}

```

### 1.6. Penugasan

Simpan ulang program praktikum6a.c dengan nama **praktikum6b\_kelas\_nim.c**, lalu ubah program konversi desimal ke biner di atas menjadi menggunakan linked list.

Unggah file **praktikum6b\_kelas\_nim.c** melalui Google Classroom sesuai dengan batas waktu yang telah ditetapkan.