

LAPORAN
PRAKTIKUM INFORMASI DAN STRUKTUR DATA

SEMESTER GANJIL 2022/2023

MODUL 3

Stack



NAMA : ALEXIS DIVASONDA SIGAT NGAING

NPM : 210711407

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ATMA JAYA YOGYAKARTA
TAHUN 2022

Membuat program toko buku dengan Stack (tumpukan). Pertama dibuat dua buah struct. Struct pertama untuk menyimpan data buku dengan nama `Max` dan struct kedua adalah Stack dengan nama `Stack`. Jumlah Stack maksimal 10 tumpukan.

Menu pertama adalah menu inisialisasi. Pengguna tidak dapat mengakses menu lainnya jika belum melakukan inisialisasi. Inisialisasi menggunakan prosedur dengan kode `(*S).top = -1;`. Inisialisasi akan mengosongkan semua Top atau lebih tepatnya menjadikan index arraynya -1 sehingga tidak menunjuk elemen apapun. Kemudian dipanggil di `main.c` : `inisialisasi(&S);`
`inisialisasi(&z_novel);` `inisialisasi(&z_majalah);`
`inisialisasi(&z_biografi);`. Jenis buku juga diinisialisasi.

Menu kedua berfungsi untuk menginputkan data. Inputan akan masuk ke dalam Struct `Max` dengan prosedur `push(&S, max);` dengan kode `(*S).top++;`
`(*S).max[(*S).top] = max;`. Pertama Top akan di naikkan dari index sebelumnya dan data akan masuk ke dalam Stack di paling atas. Terdapat fungsi `isFull(Stack S)` untuk mengecek apakah Stack sudah maksimal atau belum.

Menu ketiga merupakan menu edit data. Pengguna akan menginputkan judul buku yang ingin diedit. Judul buku akan dicek menggunakan fungsi `found(Stack S, string judul)`. Jika judul tidak ditemukan maka fungsi akan mengembalikan -1 dan jika ditemukan maka fungsi akan mengembalikan index array judul buku tersebut. Fungsi `found` akan diubah menjadi integer `index`. Kemudian pengguna akan menginputkan data buku dan data akan dimasukkan ke dalam index Stack yang dicari sebelumnya.

Menu keempat berfungsi untuk menghapus data. Data yang dihapus adalah data Stack yang paling atas. Kodenya adalah sebagai berikut.

```

Max pop(Stack *S) {
    Max temp;

    temp = (*S).max[(*S).top];
    (*S).top--;

    return temp;
}

```

Jika data paling atas diambil, data dibawahnya akan menjadi data yang paling atas. Variabel `temp` berfungsi untuk menyimpan data *temporary* (sementara) yang sebelumnya dikeluarkan dari Stack dan kemudian akan dibalikkan.

Menu kelima merupakan menu untuk menampilkan data. Tampilan data dengan prosedur `show(Stack S)`.

Menu keenam berfungsi untuk menginputkan data secara descending atau dari besar ke kecil. Tumpuannya adalah harga buku. Harga tertinggi akan ditumpuk di Stack teratas. Berikut kodenya.

```

void inputDesc(Stack *S, Max max) {
    int i, n;
    Max value;

    for(i=0; i<stackMAX; ++i) {
        for(n=i+1; n<stackMAX; ++n) {
            if((*S).max[i].harga < (*S).max[n].harga) {
                value = (*S).max[i];
                (*S).max[i] = (*S).max[n];
                (*S).max[n] = value;
            }
        }
    }
}

```

Program akan membandingkan harga sebelumnya dengan harga yang diinputkan terbaru. Harga tertinggi akan dimasukkan ke dalam variabel `value`.

Menu ketujuh merupakan menu untuk menghapus Stack sesuai dengan data yang diinputkan pengguna. Pengguna akan menginputkan judul buku yang ingin dihapus kemudian program akan mencari judul buku tersebut dengan fungsi `found(Stack S, string judul)`. Berikut kodenya.

```

case 7 :
    if(init != -1) {
        if(!isEmpty(S)) {
            printf("\nInput judul buku yang ingin dihapus : "); fflush(stdin); gets(judul);
            index2 = found(S, judul);
            if(found(S, judul) != -1) {
                strcpy(S.max[index2].judul, "-");
                strcpy(S.max[index2].nama, "-");
                strcpy(S.max[index2].jenis, "-");
                S.max[index2].harga = 0;

                pop(&S);
                printf("\n\n\tData berhasil dihapus...");
            }
            else {
                printf("\nStack Kosong [!]");
            }
        }
        else {
            printf("\n\t[!] Inisialisasi terlebih dahulu");
        }
    }
    break;

```

Fungsi found akan diubah menjadi variabel index2 dan program akan menghapus data sesuai dengan array index yang dicari tersebut (contoh : `strcpy(S.max[index2].judul, "-");`). Selanjutnya program akan menjalankan `pop (&S)` ; yang mana akan mengeluarkan Stack data tersebut.

Menu kedelapan merupakan menu cluster. Data Stack akan dibagi sesuai kategori buku. Berikut kodenya.

```

case 8 :
    if(init != -1) {
        if(!isEmpty(S)) {
            int i;

            for(i=0; i != S.top+1; i++) {
                if(strcmpi(S.max[i].jenis, "Novel")==0) {
                    push(&z_novel, max);
                }
                if(strcmpi(S.max[i].jenis, "Majalah")==0) {
                    push(&z_majalah, max);
                }
                if(strcmpi(S.max[i].jenis, "Biografi")==0) {
                    push(&z_biografi, max);
                }
            }
            printf("\nData Novel\n");
            for(i=0; i != z_novel.top+1; i++) {
                printf("\n| %s - %s - %.2f |\n", z_novel.max[i].judul, z_novel.max[i].nama, z_novel.max[i].harga);
            }
            printf("\nData Majalah\n");
            for(i=0; i != z_majalah.top+1; i++) {
                printf("\n| %s - %s - %.2f |\n", z_majalah.max[i].judul, z_majalah.max[i].nama, z_majalah.max[i].harga);
            }
            printf("\nData Biografi\n");
            for(i=0; i != z_biografi.top+1; i++) {
                printf("\n| %s - %s - %.2f |\n", z_biografi.max[i].judul, z_biografi.max[i].nama, z_biografi.max[i].harga);
            }
        }
        else {
            printf("\nStack Kosong [!]");
        }
    }
    else {
        printf("\n\t[!] Inisialisasi terlebih dahulu");
    }
}
break;

```