

LAPORAN  
PRAKTIKUM INFORMASI DAN STRUKTUR DATA

SEMESTER GANJIL 2022/2023

MODUL 6

*Multilist*



NAMA : ALEXIS DIVASONDA SIGAT NGAING

NPM : 210711407

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

Membuat program untuk sistem yang meng-*handle* pemesanan ruangan private beserta mobil yang terdaftar menggunakan struktur data Multilist. Terdapat 6 menu utama dan 1 menu tambahan (tugas). Menu pertama adalah menu input data ruangan. Terdapat *error handling* pada masing-masing inputan data. Khusus untuk data ID, inputan harus unik atau tidak boleh sama dengan ID yang telah dimasukkan sebelumnya. Pengecekan ini menggunakan fungsi perulangan `bool isParentUnik(Multilist l, int id)` yang akan mengecek semua parent yang ada dan membandingkannya dengan inputan ID yang baru. Khusus untuk inputan Status Ruangan, inputan menggunakan tipe data `bool`. Jika menginputkan angka 1 maka status Aktif atau `true`, jika 0 maka status Tidak Aktif atau `false`. Semua data kemudian disimpan berdasarkan ID. Jika ID bernilai ganjil maka program akan melakukan `insertLastParent(&l, makeDataParent(id, ruangan, status, temp))`; yang dinamakan “input di ruangan lama” dan apabila bernilai genap maka akan melakukan `insertFirstParent(&l, makeDataParent(id, ruangan, status, temp))`; yang dinamakan “input di ruangan baru”.

Menu kedua adalah menu untuk mengedit status ruangan. Program akan meminta inputan ID ruangan yang ingin di ganti statusnya dan ID akan dicari dengan fungsi `findParent(Multilist l, int cariId)`. Fungsi akan mencari data ID dengan melakukan perulangan dengan `parentBantu` yang merupakan `AddressParent` dan akan membandingkan ID dengan ID yang dicari. Jika data ditemukan maka fungsi akan mengembalikan `parentBantu`, jika tidak akan mengembalikan `NULL`. Fungsi akan dibuat sebagai variabel `tempParent` di main program. Apabila ID yang dicari statusnya `true` (Aktif) maka program akan meminta konfirmasi untuk mengganti statusnya menjadi `false` (Tidak Aktif) dan sebaliknya.

Menu ketiga adalah menu hapus data ruangan. Program akan meminta input nama ruangan dan program akan memanggil fungsi `findParentRuangan(l, ruangan)`; yang dialiaskan dengan variabel `tempParent`. Fungsi tersebut akan melakukan pencarian nama ruangan dengan melakukan perulangan dengan `parentBantu` yang merupakan `AddressParent` dan akan membandingkan nama

ruangan dengan nama ruangan yang dicari. Jika nama ruangan ditemukan maka fungsi akan mengembalikan `parentBantu` dan pada main program akan melakukan `deleteAtParent(&l, id);`. Jika tidak ditemukan maka akan mengembalikan `NULL`.

Menu keempat adalah menu input data mobil atau data Child. Program akan meminta inputan ID ruangan dan program akan mencari ID ruangan dengan fungsi `findParent(l, id);` yang dialiaskan dengan `tempParent`. Apabila status ruangan `false` atau tidak aktif maka tidak dapat melakukan input data mobil. Inputan data mobil masing-masing memiliki *error handling* dan khusus untuk inputan nomor plat tidak boleh sama atau harus unik. Pengecekan menggunakan pemanggilan fungsi `!isChildUnik(l, noPlat)`. Data kemudian disimpan pada prosedur `insertFirstChild(l, id, makeDataChild(noPlat, merek, bensin));`.

Menu kelima merupakan menu yang menampilkan jumlah ruangan terbaru. Menu keenam adalah menu tampil data. Data ditampilkan dengan memanggil prosedur `printAll(l);`. Prosedur akan melakukan perulangan hingga list kosong. Di dalamnya juga terdapat prosedur `printAllChild(temp);` yang berfungsi untuk menampilkan data mobil.

Menu ketujuh adalah menu distribusi data. Data mobil atau Child akan dibagi ke dalam Parent yang kosong atau yang isinya lebih sedikit sehingga lebih merata. Distribusi menggunakan prosedur `void distribusi(Multilist *l)`. Pertama prosedur akan melakukan perulangan untuk memasukkan ulang data dengan `insertLastParent(&n, p->data);`. Selanjutnya prosedur akan mengecek apakah terdapat Parent memiliki Child dengan fungsi Boolean. Fungsi akan mengembalikan `firstChild` apabila datanya tidak kosong. Apabila tidak memiliki Child maka program akan `continue` atau lanjut. Selanjutnya seilakukan perulangan kembali untuk mengecek apakah variabel `x` apakah kosong atau tidak. Variabel `x` merupakan data `n.firstParent`. Selanjutnya data Child akan didistribusikan ke ruangan lain dengan `insertLastChild(n, x->data.id, c->data);`. Terakhir variabel `n` menjadi variabel `l` (list).