LATIHAN STRUKTUR DATA

Kasus 1 : Grooming - Queue

A. Deskripsi Kasus

Sebuah toko Grooming kucing, setiap harinya selalu kedatangan customer dengan jumlah yang tidak tentu. Di toko tersebut hanya terdapat 1 petugas. Waktu grooming kucing tergantung dari ukuran kucing (kecil, sedang dan besar). Untuk ukuran kecil, waktu grooming adalah 30 menit, untuk ukuran sedang 45 menit sedangkan untuk yang ukuran besar 60 menit. Ketika ada pelanggan yang datang, kucing akan didaftarkan dan didata :

- waktu kedatangan (dalam satuan menit, menit 0 dimulai ketika toko buka)
- nama kucing
- ukuran kucing (kecil (K), sedang (S), besar (B))

Setelah didata, pelanggan akan dipersilakan menunggu, tetapi jika petugas kosong, kucing akan langsung di grooming.

Tugas anda adalah, buatlah program untuk mendata dan menampilkan proses antrian grooming kucing.

B. Simulasi

Berikut ini adalah simulasi kedatangan pelanggan, berapa lama waktu tunggu pelanggan dan estimasi waktu selesai. Diasumsikan tidak ada waktu jeda dari satu pelanggan ke pelanggan lainnya.

Nama	Waktu Daftar	Ukuran	Waktu Grooming	Waktu Tunggu	Est. Waktu Selesai
Miu	1	K	30	0	31
Jon	4	B	60	27	91
Chen	91	S	45	0	136

C. Cara Pengerjaan

Mission 1

Buatlah tabel perkiraan waktu tunggu dan estimasi waktu selesai berdasarkan data pendaftaran pelanggan berikut ini.

Nama	Waktu Daftar	Ukuran	Waktu Grooming	Waktu Tunggu	Est. Waktu Selesai
Doglas	4	B			
Fulgoso	8	S			

Wawa	70	K			
Dos	75	B			
Fafa	100	S			

Mission 2

Telah disediakan file header dengan spesifikasi struktur data berikut

a. Atribut data

```
/* Definisi elemen dan address */

typedef InfoPengganGrooming infotype;
typedef struct NodeQueue *addrNQ;

typedef struct{

 // silahkan isi sendiri sesuai dengan simulasi di atas

}InfoPengganGrooming;

typedef struct NodeQueue {
    infotype info; // info pelanggan
    addrNQ next;
} NodeQueue;

typedef struct {
    addrNQ Front; // tag antrian depan
    addrNQ Rear; // tag antrian belakang
} Queue;
```

b. Header Modul

```
/***** Manajemen memori *****/
/* Mengirimkan address hasil alokasi sebuah elemen dengan info X.
 * Jika alokasi berhasil, modul mengembalikan P; Info(P) = X,
Next(P) = NULL.
 * P adalah pointer yang menunjuk ke node Queue sebagai hasil
alokasi.
 * Jika alokasi gagal, modul mengembalikan NULL.
 */
addrNQ Alokasi(infotype X);


/* Melakukan dealokasi elemen P (pointer menunjuk ke alamat node
queue).
 * I.S.: P terdefinisi.
 * F.S.: P dikembalikan ke sistem.
 */
```

```
void Dealokasi(addrNQ *P);

/***** Manajemen Queue *****/
/* Membuat sebuah Queue kosong dengan Front(Q) = Nil dan Rear (Q) =
Nil
   I.S. Belum terbentuk Queue
   F.S. Sudah terbentuk Queue
*/
void CreateQueue(Queue *Q);

/* Mengetahui apakah Queue kosong atau tidak.
   mengirimkan 1 jika Queue Kosong yaitu Front(Q) = Nil dan Rear
(Q) = Nil
   Sebaliknya 0 (Queue tidak kosong)
*/
Int IsQueueEmpty(Queue Q);

/* Memasukkan info baru ke dalam Queue dengan aturan FIFO */
/* I.S. Q mungkin kosong atau Q mungkin berisi antrian */
/* F.S. info baru (data) menjadi Rear yang baru dengan node Rear
yang lama mengaitkan pointernya ke node yang baru */
void enQueue(Queue *Q, infotype data);

/* Proses: Mengambil info pada Front(Q) dan mengeluarkannya dari
Queue dengan aturan FIFO */
/* I.S. Q mungkin kosong atau Q mungkin berisi antrian */
/* F.S. info yang diambil = nilai elemen Front pd I.S. */
/* Front(Q) menunjuk ke next antrian atau diset menjadi NIll, Q
mungkin kosong */
void deQueue(Queue *Q, infotype *data);


/* Mengirimkan banyaknya elemen queue jika Q berisi atrian atau
   mengirimkan 0 jika Q kosong
*/
int NBElmt(Queue Q);

#endif // QUEUE_H
```

Tugas Anda : Buatlah Header Modul Queue yang dimodifikasi sesuai kasus, dengan perubahan sebagai berikut.

Perubahan Modul

Nama Method	Perubahan	Keterangan
`CreateQueue`	`BuatAntrian`	Membuat Antrian baru (dilakukan diawal ketika toko buka)
`IsQueueEmpty`	`CekAntrianKosong`	Mengecek apakah antrian kosong.
`enQueue`	`MasukAntrian`	Pelanggan yang sudah didata, maka akan dimasukkan ke antrian
`deQueue`	`MulaiGrooming`	Pelanggan yang akan digrooming dikeluarkan dari antrian.

Tambahan Modul

Nama Method	Spesifikasi	Parameter	Tipe Modul
`HitungWaktuSelesai`	Menghitung waktu estimasi selesai layanan grooming pada setiap customer berdasarkan informasi waktu tunggu, waktu grooming dan waktu daftar. *{silangkan buat logic rumus sendiri untuk proses kalkulasi}*	`NodeQueue sebagai customer, Char sbg Estimasi waktu selesai pada antrian sebelumnya` int (output)	fungsi
`HitungWaktuTunggu`	Menghitung waktu tunggu grooming pada setiap customer. *{silangkan buat logic rumus sendiri untuk proses kalkulasi}*	`NodeQueue ,` `---` *`tambahkan lagi parameter yang dibutuhkan`* int (output)	fungsi

`HitungWaktuGrooming`	Menghitung waktu estimasi grooming berdasarkan ukuran kucing I.S : Ukuran kucing dalam inisial karakter K, S dan B (lihat di soal) F.S : Waktu grooming dalam bentuk int	Char (input) Int (output)	Function

Lakukan perubahan komentar berdasarkan keterangan, spesifikasi, parameter atau kembalian sesuai tabel diatas. Jika ingin menambahkan file header lagi silahkan.

Mission 3 - Implementasikan semua header modul yang ada pada file header

Mission 4 - Buatlah main driver dengan alur sebagai berikut:
1. Buat antrian kosong
2. Input data customer Grooming Kucing
3. Masukan Pelanggan ke antrian
4. Hitung Estimasi Waktu selesai pelanggan pada setiap customer
5. Print Informasi pelayanan grooming kucing

Kasus 2 : Parking - Stack

A. Deskripsi Kasus

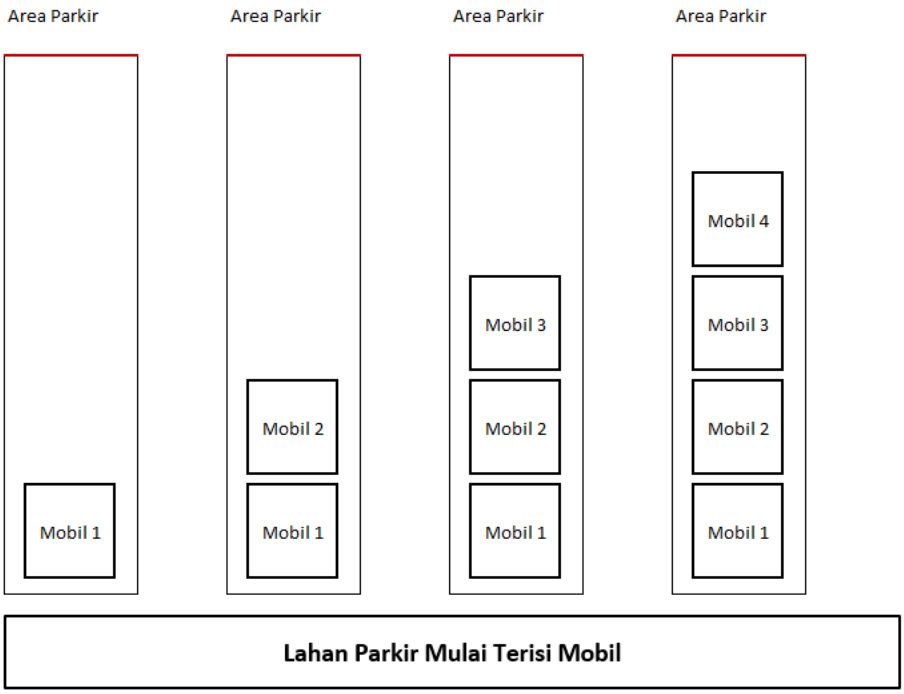
Terdapat sebuah tanah dengan luas area sempit dijadikan sebagai lahan parkir dan ada petugas parkir yang mengelolanya. Area tersebut hanya mampu menampung 1 jalur mobil saja, sehingga hanya ada satu pintu untuk keluar/masuk mobil. Setiap ada mobil yang mau parkir, posisi mobil selalu berada di ujung dekat pintu. Jika ada mobil yang ingin keluar, mobil yang berada dekat pintu keluar harus keluar terlebih dahulu. Alhasil keadaan tersebut menyebabkan mobil bertumpuk.

Pada kasus tertentu, pemilik mobil tidak ingin menunggu mobil yang menghalanginya. Sehingga petugas parkir harus memindahkan mobil penghalang menuju area luar parkir dan memasukannya lagi ke lahan parkir dengan posisi urutan mobil sama.

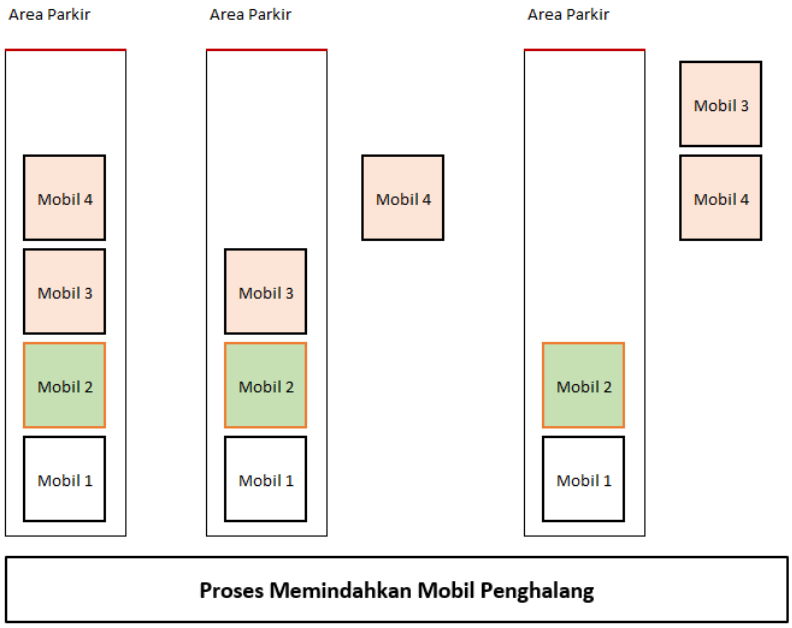
Tugas Anda adalah membantu petugas parkir untuk mengatur keluar atau masuk mobil di area parkir tersebut.

B. Ilustrasi

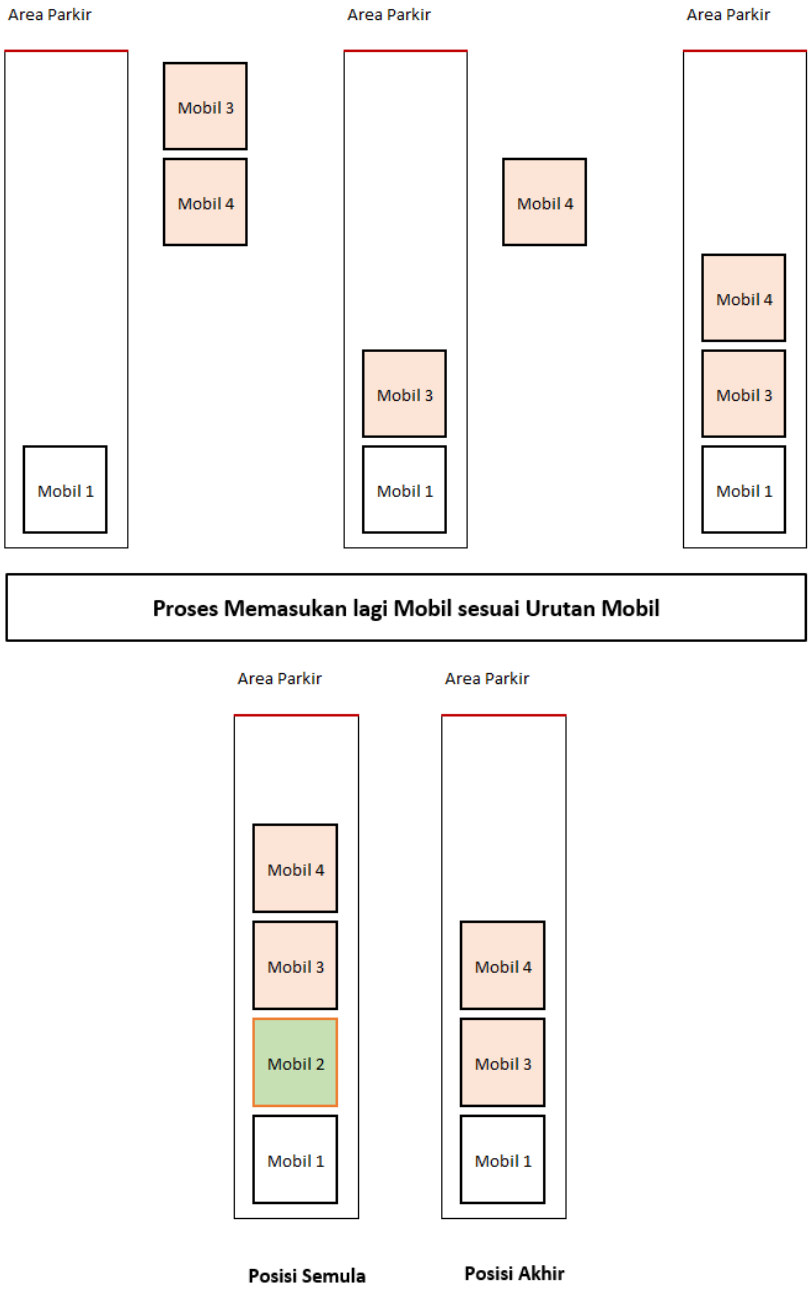
Diketahui sebuah lahan parkir yang memiliki 1 buah pintu keluar/masuk dan jumlah mobil yang akan diparkirnya berbeda-berda setiap hari. Proses parking mobil yang masuk ke area lahan parkir ditunjukkan pada Gambar dibawah ini.



Selanjutnya jika mobil 4 akan keluar parkir, maka mobil tersebut maju ke arah pintu parkir dan keluar dari area parkir tanpa ada masalah. Sehingga yang tersisa adalah mobil 1, mobil 2 dan mobil 3. Namun jika ada mobil yang akan keluar adalah mobil 2 terdapat 2 mobil yang menghalanginya menuju pintu keluar, maka bagaimanakah caranya? Silahkan simak ilustrasi dibawah ini.



Pada ilustrasi tersebut, terlihat bahwa petugas parkir harus memindahkan mobil 3 dan mobil 4. Kemudian memasukan satu persatu lagi ke dalam lahan parkir. Pada akhir proses hanya tersisa mobil 1, mobil 3 dan mobil 4 dengan urutan yang sama seperti semula.



C. Cara Pengerjaan

Mission 1 - Buatlah abstraksi pemahaman saudara terhadap kasus yang diberikan
Mission 2 - Modifikasi header stack linked list
Mission 3 - Implementasikan semua header modul yang ada pada file header
Mission 4 - Buatlah alur pada main driver sesuai dengan persoalan
Mission 5 - Implementasikan main driver sesuai dengan alur mission 4.
Note. Cara pengerjaan setiap mission caranya sama seperti pada kasus pertama (Queue)