Topik: ADT List Linier Sirkuler

Tujuan Praktikum:
- Memahami konsep List Linier, terutama variasi List Linier Sirkuler (Next(Last(L)) = First(L)).
- Memahami implementasinya dalam bahasa C.

PETUNJUK PRAKTIKUM:
1. Untuk setiap file yang Anda buat, cantumkan header sebagai berikut:

```
/* NIM/Nama  :
   Nama file :
   Topik     :
   Tanggal   :
   Deskripsi :                                                    */
```

2. Seluruh file di-upload setelah dikompres menjadi 1 file dengan nama: P08<nim>.zip.
3. Softcopy materi kuliah/diktat, termasuk yang terkait dengan pemrograman dengan Bahasa C dapat dilihat pada situs **http://kuliah.itb.ac.id/app245/course/** pada link **IF2110 Algorithm & Data Structure**.
4. HANYA ADT YANG DAPAT DI-COMPILE YANG AKAN DIPERIKSA. File yang tidak dapat di-*compile* akan otomatis mendapatkan nilai 0.
5. Tugas ini bersifat INDIVIDUAL. Anda dipersilakan membaca dan menggunakan material yang Anda punya, tetapi tidak ada toleransi bagi pencontek. Jika terbukti, baik yang dicontek maupun yang mencontek akan mendapatkan nilai 0.
6. Ikuti petunjuk asisten untuk pengumpulan tugas ini.

SELAMAT BEKERJA.

Soal List Linier Sirkuler

Ambillah ADT List sirkuler dengan representasi fisik berkait dengan pointer yang telah Anda kerjakan sebagai tugas pra-praktikum.
Tambahkan fungsi dan prosedur berikut ini ke ADT tersebut:

- Fungsi **NbX**

```
int NbX (Infotype X, List L);
/* Mengirimkan jumlah kemunculan elemen bernilai X dalam list L;
mengirimkan 0 jika list kosong */
```

- Function **Jarak**

```
int Jarak (List L, Infotype X, Infotype Y);
/* Mengirimkan jarak (banyaknya elemen) antara elemen dengan infotype
X dan elemen dengan infotype Y (tidak termasuk elemen X dan Y
sendiri).
Jika X dan Y tepat bersebelahan, jaraknya adalah 0.
Jika ada lebih dari satu elemen X dan/atau Y di L maka digunakan
elemen X dan Y yang pertama kali ditemukan.
Urutan kemunculan X dan Y bebas, artinya X dapat muncul sebelum atau
sesudah Y. */
/* Prekondisi : X dan Y pasti berbeda dan pasti ada di L, sehingga L
tidak mungkin kosong. */
```

- Prosedur **RemAllX**:

```
void RemAllX (List * L, Infotype X);
/* I.S. L terdefinisi, mungkin kosong */
/* F.S. Tidak ada elemen L yang bernilai X */
/* Menghapus seluruh elemen L yang bernilai X */
```

- Prosedur **SubList**

```
void SubList (List L1, int start, int count, int arah, List * L2);
/* I.S. L1, start, count, dan arah terdefinisi.
        start ≥ 1
        count ≥ 1
        arah bernilai 1 atau -1.
        L2 sembarang. */
/* F.S. Jika arah = 1, maka L2 berisi elemen L1 dari elemen ke-start
      hingga elemen ke-(start+count-1) (sesuai urutan di L1).
        Jika arah = -1, maka L2 berisi elemen L1 dari elemen
      ke-(start+count-1) hingga elemen ke-start (dengan urutan
      terbalik dari urutan di L1).
        Setiap elemen L2 harus dialokasi sebagai elemen baru. Jika
      ada elemen yang gagal dialokasi, maka L2 menjadi kosong dan
      semua elemen yang sudah telanjur dialokasi harus didealokasi.
        Jika start > jumlah elemen L1, L2 kosong.
        Jika start ≤ jumlah elemen L1 dan (start+count-1) > jumlah
      elemen L1, L2 berisi (jumlah elemen L1 – start + 1) elemen
      (kurang dari count).

      Urutan pada List dimulai dari 1
*/
```

- Prosedur **Rotate**:

```
void Rotate (List * L, int N, boolean isLeft);
/* I.S. L terdefinisi, mungkin kosong */
/* F.S. Jika isLeft = true: */
/*         N buah elemen terdepan L dipindah ke belakang. */
```

```
/*          Elemen ke-N+1 pada L awal menjadi First(L) yang baru. */
/*      Jika isLeft = false: */
/*          N buah elemen terbelakang L dipindah ke depan. */
/*          Elemen ke-N dari belakang pada L awal menjadi First(L) */
/*          yang baru. */
/*      Jika N lebih besar dari jumlah elemen L, L tetap */
/* Contoh: Rotate([3,5,7,9,11],2,true) → L = [7,9,11,3,5]
/*          Rotate([3,5,7,9,11],2,false) → L = [9,11,3,5,7]
```

Pekerjaan akan diperiksa menggunakan ADT Asisten. Submitlah **listsirkuler.h** dan **listsirkuler.c** yang dikompres menjadi file *.zip.