Laporan Tugas Algoritma dan Struktur data

DISUSUN OLEH:

PRIAGUNG SATYAGAMA – 13516089
RENJIRA NAUFHAL DHIAEGANA - 13516014

A. Perubahan pada array.c

1. void MakeEmpty

- ada tambahan parameter int MaxElement untuk menyimpan kapasitas array
- proses nya, ada pengalokasian memori, inisialisasi variabel MaxEl di TabInt
- inisialisasi Neff dengan 0

```
void MakeEmpty (TabInt * T, int MaxElement)
/* I.S. T sembarang */
/* F.S. Terbentuk tabel T kosong dengan kapasitas MaxElement
{
  TI(*T) = (int*) malloc(MaxElement*sizeof(int)+1);
  MaxElmt(*T) = MaxElement;
  Neff(*T) = 0;
}
```

2. int MaxNbEl

- Mengubah return IdxMax diganti jadi return MaxElmt(T)

```
int MaxNbEl (TabInt T)
/* Mengirimkan maksimum elemen yang
{
  return MaxElmt(T);
}
```

3. boolean IsIdxValid

- asalnya return (IdxMin<=i) && (IdxMax>=i)
- menjadi return (IdxMin<=i) && (MaxElmt(T)>=i)

```
boolean IsIdxValid (TabInt T, IdxType i)
/* Mengirimkan true jika i adalah indeks yang
/* yaitu antara indeks yang terdefinisi utk c
{
  return (IdxMin <= i) && (MaxElmt(T) >= i);
}
```

4. void BacaIsi

- kondisi saat N==0 asalnya memanggil fungsi MakeEmpty, menjadi memanggil fungsi Dealokasi

```
void BacaIsi (TabInt * T)
/* I.S. T sembarang */
/* F.S. Tabel T terdefinisi */
/* Proses : membaca banyaknya elemen T dan m
/* 1. Baca banyaknya elemen diakhiri enter,
/*    Pembacaan diulangi sampai didapat N ya
/*    Jika N tidak valid, tidak diberikan pe
/* 2. Jika 0 < N <= MaxNbEl(T); Lakukan N ka
      IdxMin satu per satu diakhiri enter */
/*    Jika N = 0; hanya terbentuk T kosong *
{
  int N, i;
  scanf("%d", &N);
  while(N>MaxNbEl(*T) || N<0){
    scanf("%d", &N);
  }
  if(N==0){
    Dealokasi(T);
  }
  else{
    for(i=IdxMin;i<=N;i++){
      scanf("%d", &Elmt(*T, i));
    }
  }
  Neff(*T)=N;
}
```

5. void BacaIsiTab

- jika input pertama user -9999, memanggil prosedur Dealokasi;

- jika input user bukan -9999, tapi array sudah penuh, maka :

- menyiapkan variabel temp bertipe TabInt dengan memori sebesar MaxElmt dari T (menggunakan prosedur MakeEmpty)

- mengcopy semua elemen di T ke temp (menggunakan prosedur CopyTab)

- membebaskan memori yang digunakan oleh T (menggunakan prosedur Dealokasi)

- menyiapkan variabel T dengan memori MaxElmt(temp)+1 (menggunakan prosedur MakeEmpty)

- mengcopy semua elemen di temp ke T (menggunakan prosedur CopyTab)

- membebaskan memori yang digunakan oleh temp (menggunakan prosedur Dealokasi)

```
{
  ElType x;
  TabInt temp;
  Neff(*T)=0;
  scanf("%d",&x);
  if(x==-9999){
    Dealokasi(T);
  }
  while(x != -9999){
    if(IsFull(*T)){
        //menyiapkan temp
        MakeEmpty(&temp, MaxElmt(*T));
        //mengcopy seluruh elemen T ke temp
        CopyTab(*T, &temp);
        //membebaskan memori di T
        Dealokasi(T);
        //menyiapkan T dengan maks elemen lebih 1
        MakeEmpty(T, MaxElmt(temp)+1);
        //mengcopy kembali seluruh elemen temp ke T
        CopyTab(temp, T);
        //membebaskan memori di temp
        Dealokasi(&temp);
    }
    Neff(*T)++;
    Elmt(*T, Neff(*T)) = x;
    scanf("%d",&x);
  }
}
```

6. InverseTab

- memanggil prosedur MakeEmpty untuk menyiapkan memori untuk TOut

```
TabInt InverseTab (TabInt T)
/* Menghasilkan tabel dengan urutan tempat yang te
/* elemen pertama menjadi terakhir, */
/* elemen kedua menjadi elemen sebelum terakhir, d
/* Tabel kosong menghasilkan tabel kosong */
{
  TabInt TOut;
  MakeEmpty(&TOut, MaxElmt(T));
  int i;
  for(i=GetLastIdx(T);i>=GetFirstIdx(T);i--){
    Elmt(TOut, GetLastIdx(T)-i+1) = Elmt(T, i);
  }
  Neff(TOut) = NbElmt(T);
  return TOut;
}
```

7. TabInt PlusTab, MinusTab, KaliTab

- menambahkan pemanggilan prosedur MakeEmpty untuk mengalokasikan memori untuk variabel THasil

```
TabInt PlusTab (TabInt T1, TabInt T2)
/* Prekondisi : T1 dan T2 berukuran sama dan tidak
/* Mengirimkan  T1+T2, yaitu setiap elemen T1 dan
{
  TabInt THasil;
  int i;

  // mengalokasikan memori untuk THasil
  MakeEmpty(&THasil, MaxElmt(T1));

  for(i=GetFirstIdx(T1); i<=GetLastIdx(T1); i++){
    Elmt(THasil, i) = Elmt(T1, i) + Elmt(T2, i);
  }
  Neff(THasil) = Neff(T1);

  return THasil;
}
```

```
TabInt MinusTab (TabInt T1, TabInt T2)
/* Prekondisi : T1 dan T2 berukuran sama dan tidak
/* Mengirimkan T1-T2, yaitu setiap elemen T1 dikura
{
  TabInt THasil;
  int i;

  // mengalokasikan memori untuk THasil
  MakeEmpty(&THasil, MaxElmt(T1));

  for(i=GetFirstIdx(T1); i<=GetLastIdx(T1); i++){
    Elmt(THasil, i) = Elmt(T1, i) - Elmt(T2, i);
  }
  Neff(THasil) = Neff(T1);

  return THasil;
}
```

```
TabInt KaliTab (TabInt T1, TabInt T2)
/* Prekondisi : T1 dan T2 berukuran sama dan tidak
/* Mengirimkan T1 * T2 dengan definisi setiap eleme
{
  TabInt THasil;
  int i;

  // mengalokasikan memori untuk THasil
  MakeEmpty(&THasil, MaxElmt(T1));

  for(i=GetFirstIdx(T1); i<=GetLastIdx(T1); i++){
    Elmt(THasil, i) = Elmt(T1, i) * Elmt(T2, i);
  }
  Neff(THasil) = Neff(T1);

  return THasil;
}
```

8. void Dealokasi

   - parameter input/output : TabInt

   - proses nya : membebaskan memori, MaxEl = 0, Neff = 0

```
void Dealokasi (TabInt *T)
/* I.S. T Terdefinisi*/
/* F.S. TI(T) dikembalikan k
{
    free(TI(*T));
    MaxElmt(*T) = 0;
    Neff(*T) = 0;
}
```

B. perubahan pada array.h

- Menghapus konstanta IdxMax
- Mengubah TI dalam TabInt menjadi bertipe pointer to ElType
- Menambahkan komponen dari TabInt yaitu MaxEl bertipe integer yang berfungsi untuk menyimpan total elemen yang bisa ditampung di array
- Menambahkan selector MaxElmt(T) untuk mengakses MaxEl dalam TabInt
- Menambahkan prototype prosedur Dealokasi
- Menambahkan parameter input MaxElmt di prosedur MakeEmpty

```c
/*  Kamus Umum */
#define IdxMin 1
/* Indeks minimum array */
#define IdxUndef -999
/* Indeks tak terdefinisi*

/* Definisi elemen dan kol
typedef int IdxType;  /* t
typedef int ElType;   /* t
typedef struct {
  ElType *TI; /* memori te
  int Neff;
  int MaxEl;  /* >=0, bany
} TabInt;
```

```c
/* ********** SELEKTOR ********** */
#define Neff(T)   (T).Neff
#define TI(T)     (T).TI
#define Elmt(T,i) (T).TI[(i)]
#define MaxElmt(T) (T).MaxEl

/* ********** KONSTRUKTOR ********** */
/* Konstruktor : create tabel kosong  */
void MakeEmpty (TabInt * T, int MaxElmt);
/* I.S. T sembarang */
/* F.S. Terbentuk tabel T kosong dengan kap

void Dealokasi (TabInt *T);
/* I.S. T Terdefinisi */
/* F.S. TI(T) dikembalikan ke system, MaxEl
```