Pertemuan 10

Waktu : 135 menit

Tujuan Pembelajaran : Mahasiswa mampu menjelaskan teknik pemrograman menggunakan Queue.

Substansi Materi : Queue dengan Double Linked List

Tabulasi Kegiatan Perkuliahan

No	Tahap Kegiatan	Kegiatan Pengajar	Kegiatan Mahasiswa	Media & Alat	Waktu
1	Pendahuluan	1. Membuka pertemuan 2. Mengulang materi pertemuan sebelumnya	Menyimak Bertanya	Papan Tulis	20 Menit
2	Penyajian Materi	1. Queue dengan Double Linked List 2. Latihan Soal	Menyimak Bertanya Menjawab Pertanyaan	Papan Tulis	80 Menit
3	Penutup	1. Menyimpulkan materi pertemuan 2. Memberikan tugas kecil 3. Menutup pertemuan	Menyimak	Papan tulis	35 Menit

MATERI KULIAH

Implementasi QUEUE dengan DOUBLE LINKED LIST

Selain dengan menggunakan array, queue juga dapat dibuat dengan menggunakan linked list. Metoda linked list yang digunakan adalah double linked list. Deklarasi tipe, konstanta, dan variable yang akan digunakan adalah:

```
Type
    Point = ^simpul;
    Simpul = record
        Isi : TipeData;
```

```
        Next : Point;
        End;

    Queue = record
        Head : Point;
        Tail : Point;
        End;
Var
    Q : Queue;
    N : 0 . . Max_Queue;    {Jumlah Antrian}
```

Operasi-Operasi Pembuatan Queue dengan Double Linked List

1. Create

   Procedure create berguna untuk menciptakan QUEUE yang baru dan kosong yaitu dengan cara mengarahkan pointer HEAD dan TAIL kepada NIL.

```
Procedure Create;
Begin
    Q.Head := NIL;
    Q.Tail := Q.Head;
    N := 0;
End;
```

2. Empty

   Function Empty berguna untuk mengecek apakah QUEUE masih kosong atau sudah berisi data. Hal ini dilakukan dengan mengecek apakah head masih menunjuk pada nil atau tidak, jika ya maka kosong.

```
Function Empty : Boolean;
Begin
    If Q.Head = Nil Then
        Empty := True
    Else
        Empty := False;
End;
```

3. Full

   Function Full berguna untuk mengecek apakah QUEUE sudah penuh atau masih bias menampung data dengan cara mengecek apakah N (JumlahQueue) sudah sama dengan Max_Queue atau belum, jika ya maka penuh.

```
Function Full : Boolean;
Begin
    If N = Max_Queue Then
        Full := True
    Else
        Full := False;
End;
```

4. EnQueue

   Procedure EnQueue berguna untuk memasukkan 1 elemen ke dalam QUEUE. Head dan Tail mula-mula menunjuk ke NIL.

```
Procedure EnQueue (Elemen : TipeData);
Var Now : Point;
Begin
```

```
        If Not Full Then
        Begin
                New(Now);
                Now^.Isi := elemen;
                Now^.Next := NIL;
                If Empty Then
                Begin
                        Q.Head := Now;
                        Q.Tail := Now;
                        N := 1;
                End else
                Begin
                        Q.Tail^.next := Now;
                        Q.Tail := Now;
                End;
        End;
End;
```

5. DeQueue

   Procedure DeQueue berguna untuk mengambil 1 elemen dari QUEUE. Jal ini dilakukan dengan cara menghapus satu simpul yang terletak paling depan (head).

```
Procedure DeQueue;
Var Now : Point;
Begin
        If Not Empty Then
        Begin
```

```
            Now := Q.Head;
            Q.Head := Q.Head^.Next;
            Dispose(now);
            Dec (N);
            End;
End;
```