Pertemuan 8

Waktu : 135 menit

Tujuan Pembelajaran : Mahasiswa mampu menjelaskan teknik pemrograman menggunakan Stack.

Substansi Materi : Stack

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. Stack dengan Linked List 2. Deklarasi stack dengan linked list 3. Operasi-operasi stack dengan linked list	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

Stack dengan Single Linked List

Implementasi stack dengan Single Linked List hampir sama dengan stack pada array. Keunggulannya terletak pada penggunaan alokasi memory yang dinamis sehingga menghindari pemborosan memory. Misalnya pada stack dengan array disediakan tempat untuk stack berisi 150 elemen, sementara ketika dipakai oleh user hanya diisi 50 elemen, maka telah terjadi pemborosan memori untuk sisa 100 tempat elemen yang tak terpakai. Dengan penggunaan linked list maka tempat yang disediakan akan sesuai dengan banyaknya elemen yang mengisi stack. Karena itu pula dalam linked list tidak dikenal

dengan istilah Full, sebab biasanya program tidak menentukan jumlah elemen stack yang mungkin ada.

Deklarasi tipe, dan variable yang akan dipakai dalam penjelasan operasi stack.

```
Type
    TipeData = Byte;
    Point = ^Simpul;
    Simpul = record
        Isi : TipeData;
        Next : Point;
    End;
Var
    Top : Point;
```

Operasi-operasi pada Stack dengan Single Linked List

- Create : Membuat stack baru yang masih kosong

```
Procedure Create;
  Begin
      Top := nil;
  End;
```

- Empty : Fungsi untuk menentukan apakah stack yang ada masih kosong atau tidak.

```
Function Empty : Boolean;
  Empty := False;
  if Top = nil then
```

```
        Empty := True;
    End;
```

- Push : Memasukan elemen baru ke dalam stack

```
    Procedure Push(elemen:TipeData);
    Var Now : Point;
      Begin
          New(Now);
          Now^.isi := elemen;
          If empty then
            Now^.next := nil;
          Else
            Now^.next := top;
          Top:= Now;
      End;
```

- Pop : Mengambil elemen teratas dari stack.

```
    Procedure Pop(elemen:TipeData);
      Var
          Now : Point;
          Begin
          If not Empty then
            Begin
                Elemen := Now^.isi;
                Now := Top;
                Top := Top^.next;
```

```
            Dipose(Now);
        End;
    End;
```

- Clear : Mengosongkan stack (Jika top = 0, maka stack dianggap kosong)

```
Procedure Clear;
Var
  Trash = TipeData;
  Begin
      While not empty do Pop (Trash);
  End;
```