

Modul 7 : ADT Stack

7.1 Waktu Pelaksanaan Praktikum

Durasi kegiatan praktikum = **170 menit**, dengan rincian sebagai berikut (misalkan):

- 15 menit untuk pengerjaan Tes Awal atau wawancara Tugas Pendahuluan
- 60 menit untuk penyampaian materi
- 45 menit untuk pengerjaan tugas / Studi Kasus
- 50 menit **Pengayaan**

7.2 Tujuan

Setelah mengikuti praktikum ini mahasiswa diharapkan dapat:

1. Mampu membuat stack menggunakan array
2. Mampu membuat stack menggunakan linked list.....

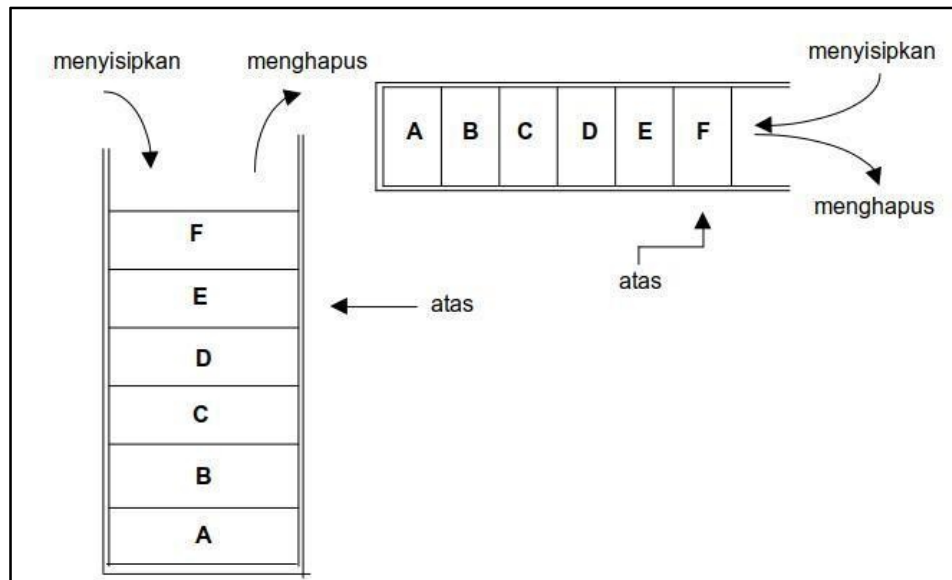
7.3 Alat & Bahan

1. Komputer
2. Java IDE

7.4 Dasar Teori

Stack

Stack atau tumpukan adalah kumpulan data yang hanya bisa dilakukan penambahan (penyisipan) data dan penghapusan data pada salah satu ujung yang sama.



Dengan memperlihatkan ilustrasi-ilustrasi yang disebutkan maka kita bisa melihat bahwa stack merupakan suatu list yang mempunyai karakteristik “masuk terakhir keluar pertama” (last in first out – LIFO).

Operasi dasar pada stack antara lain pop dan push. **Push** adalah operasi menambah/menyisipkan item pada stack. **Pop** adalah proses mengambil item dari stack. Stack bisa dibuat dengan Array atau Linked List. Tabel dibawah memberikan ilustrasi operasi pada stack yang dibuat dengan array.

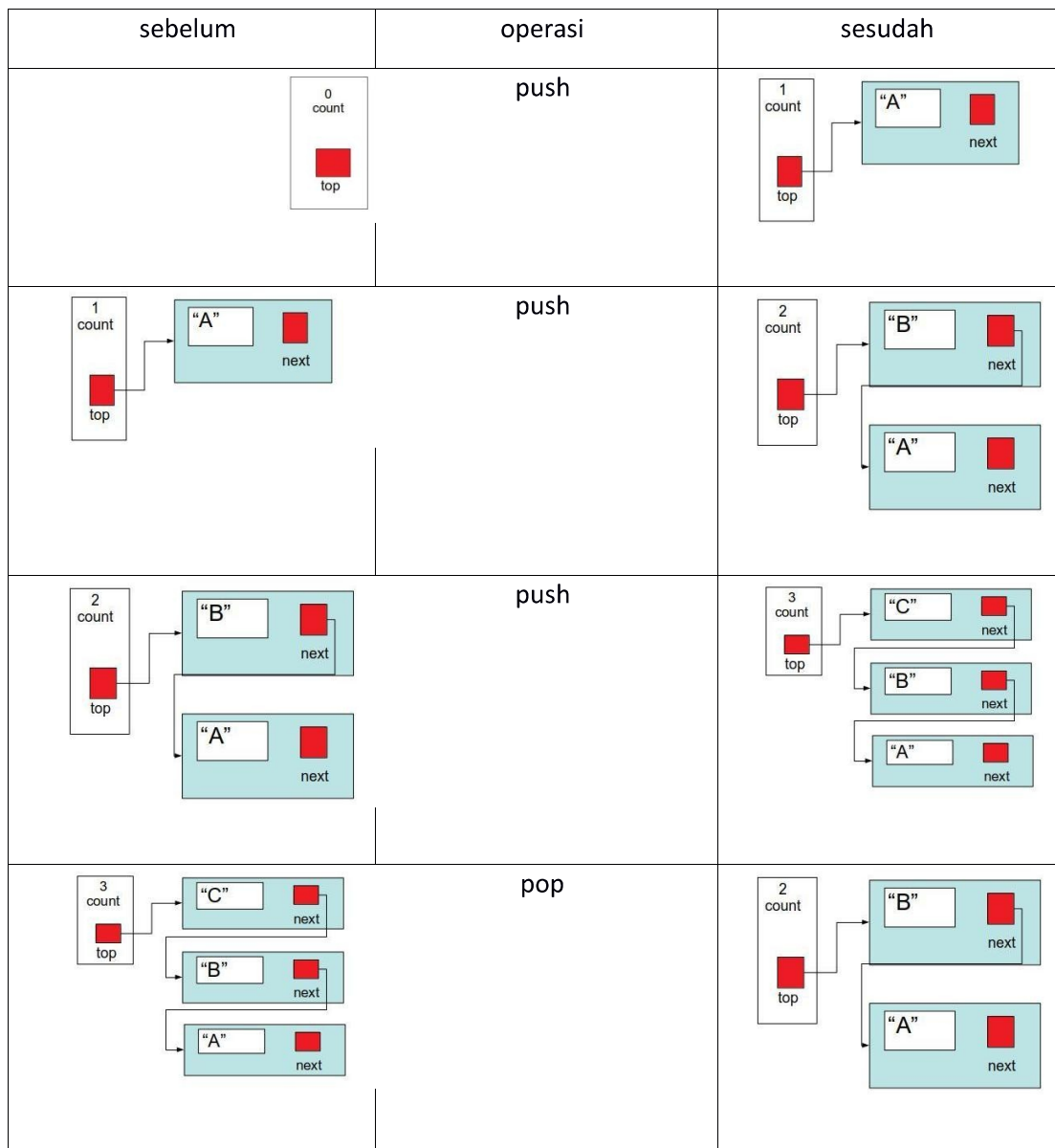
sebelum	operasi	sesudah
[]	push("A")	["A"]
["A"]	push("B")	["A","B"]
["A","B"]	push("C")	["A","B","C"]
["A","B","C"]	pop()	["A","B"]
["A","B"]	push("D")	["A","B","D"]
["A","B","D"]	pop()	["A","B"]

Dalam implementasi stack ini disimpan variabel **count** yang menghitung index atau nomor urut terakhir item dalam stack. Operasi push menambahkan item ke dalam array pada index/urutan **count+1** atau **setelah item terakhir** dalam array. Operasi pop akan mengambil item dari dalam array pada index **count** atau **item terakhir** dalam array.

Pada tabel dibawah diilustrasikan operasi stack menggunakan array. Perbedaan dengan stack diatas adalah implementasi proses **push** akan menambahkan item pada **array index ke 0**. Proses **pop** akan mengambil item pada **array index ke 0**.

sebelum	operasi	sesudah
[]	push("A")	["A"]
["A"]	push("B")	["B","A"]
["B","A"]	push("C")	["C","B","A"]
["C","B","A"]	pop()	["B","A"]
["B","A"]	push("D")	["D","B","A"]
["D","B","A"]	pop()	["B","A"]

Gambar dibawah memberikan ilustrasi stack menggunakan single linked list. Top merupakan pointer yang menunjuk pada item paling atas yang merupakan item terakhir yang dimasukkan dalam stack. Dalam islutrasi ini operasi **push** pada stack menggunakan operasi **add first** pada single linked list dan operasi **pop** menggunakan operasi **remove first**. Jika operasi **push** menggunakan operasi **add last** maka operasi **pop** akan menggunakan **remove last**.



7.5 Prosedur Praktikum

1. Buatlah file percobaan stack_array.java berikut....

```

1  import java.util.Scanner;
2
3  Public class stack_array {
4      Scanner masuk = new Scanner(System.in);
5
6      int choice, i;
7      char item;
8      char arr_stack[MAX_SIZE];
9      int count = 0;
10     int keluar = 0;
11

```

```

12  Public void push(char item)
13  {
14      if (count == MAX_SIZE)
15      {
16          system.out.print ( "\n# Stack Penuh");
17      }
18      else
19      {
20          .....
21          .....
22          .....
23      }
24  }
25
26  Public void pop()
27  {
28      if (count == 0)
29          system.out.print ( "\n## Stack kosong");
30      else
31      {
32          .....
33          .....
34      }
35  }
36
37  Public void printAll()
38  {
39      system.out.print ( "\n## Stack Size : " + count);
40      for (i = (count - 1); i >= 0; i--)
41          system.out.print ( "\n## No Urut/index : " + i + ", Value : " + arr_stack[i]);
42  }
43
44  Public void menu()
45  {
46      system.out.print ( "\nMasukkan operasi yang akan dilakukan (1:push, 2:pop,
47      3:print) : ");
48      choice=masuk.nextInt();
49      switch (choice)
50      {
51          case 1:
52          {
53              system.out.print ( "\nMasukkan huruf yang akan dipush : ");
54              item= masuk.nextLine();
55              push(item);
56          }
57          break;
58          case 2:pop();
59          break;
60          case 3:printAll();
61          break;
62          default:

```

```

62  system.out.print ( "\n1:push, 2:pop, 3:print\n");
63  keluar = 1;
64  break;
65  }
66  }
67
68  Public static void main()
69  {
70      do{
71          menu();
72      } while (keluar == 0);
73  }
74  }

```

2. Tambahkan kode berikut pada baris 20 sampai 22

```

1      arr_stack[count] = item;
2      System.out.print ( "\n# PUSH No urut/index : " + count
3      + ", Push : " + item);
      count++;

```

3. Tambahkan kode berikut pada baris 32 dan 33

```

1      --count;
2      System.out.print ( "\n##POP No urut/index : " + count
      + ", Value : " + arr_stack[count]);

```

4. Jalankan program dan

- pilih menu push dan masukkan "A"
- pilih menu push dan masukkan "B"
- pilih menu push dan masukkan "C"
- pilih menu print
- pilih menu pop
- pilih menu print
- pilih menu push dan masukkan "D"
- pilih menu print

7.6 Hasil Percobaan

Tuliskan hasil dari percobaan diatas

7.7 Analisis Hasil

Tuliskan Analisis hasil dari percobaan diatas.

7.8 Kesimpulan

Tuliskan kesimpulan dari percobaan diatas

7.9 Latihan

Buat pseudocode untuk operasi pop dan push pada stack menggunakan array dengan ketentuan operasi push akan menambahkan item pada array index ke 0 dan operasi pop akan mengembalikan item pada array index ke 0

7.10 Tugas

Modifikasi program `stack_array.java` sehingga operasi push akan menambahkan item pada array index ke 0 dan operasi pop akan mengembalikan item pada array index ke 0.

7.11 DAFTAR PUSTAKA

- Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, "Data Structures and Algorithms Using Java 6 edition", Wiley, USA, 2014.
- John R. Hubbard, "Scaum's Outline of Data Structures With Java second Edition", McGraw-Hill, New york, 2007.
- Robert Lafore, "Data Structures and Algorithm in Java second Edition", Sams Publishing, Indiana, 2003