



NAMA : Alexander Agung Raya
NIM : 2341720040
NO ABSEN : 03
KELAS : 1F
MATERI : Stack

LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

*FILE NAME =03_Alexander Agung Raya _1F_P8

* Pertemuan 7

Stack

Percobaan 1 :

Barang03 :

```
1 package P8;  
2 public class Barang03 {  
3     int Kode;  
4     String Nama;  
5     String Kategori;  
6  
7     public Barang03 (int Kode, String Nama, String Kategori) {  
8         this.Kode = Kode;  
9         this.Nama = Nama;  
10        this.Kategori = Kategori;  
11    }  
12    public void TampilDataBarang () {  
13        System.out.println("=====");  
14        System.out.println("Nama Barang :");  
15        System.out.println("=====");  
16        System.out.println("Nama Barang : " + Nama);  
17        System.out.println("Kode Barang : " + Kode);  
18        System.out.println("Kategori Barang : " + Kategori);  
19    }  
20 }
```

Gudang03 :

```
1 package P8;  
2 import java.util.Scanner;  
3  
4 public class Gudang03 {  
5     static Barang03[] Tumpukan;  
6     static int Size;  
7     static int Top;  
8  
9     public Gudang03 (int Kapasitas) {  
10        Size = Kapasitas;  
11        Tumpukan = new Barang03[Size];  
12        Top = -1;  
13    }  
14  
15    public boolean CekKosong () {  
16        if (Top == -1) {  
17            return true;  
18        } else {  
19            return false;  
20        }  
21    }  
22  
23    public static boolean CekPenuh () {  
24        if (Top == Size - 1) {  
25            return true;  
26        } else {  
27            return false;  
28        }  
29    }  
30  
31    public static void TambahBarang (Barang03 brg) {  
32        if (!CekPenuh()) {  
33            Tumpukan[Top] = brg;  
34            System.out.println("Barang " + brg.Nama + " berhasil Di Tambahkan");  
35            Top++;  
36            System.out.println("Barang Di Gudang Sudah Penuh.");  
37        }  
38    }  
39  
40    public Barang03 AmbilBarang () {  
41        if (!CekKosong()) {  
42            Barang03 brg = Tumpukan[Top];  
43            Top--;  
44            System.out.println("Barang " + brg.Nama + " diambil dari Gudang.");  
45            return brg;  
46        } else {  
47            System.out.println("Tumpukan Barang Kosong");  
48            return null;  
49        }  
50    }  
51  
52    public Barang03 LihatBarangTeratas () {  
53        if (!CekKosong()) {  
54            Barang03 barangTeratas = Tumpukan[Top];  
55            System.out.println("Barang Teratas : " + barangTeratas.Nama);  
56            return barangTeratas;  
57        } else {  
58            System.out.println("Tumpukan Barang Kosong");  
59            return null;  
60        }  
61    }  
62  
63    public void TampilBarang () {  
64        if (!CekKosong()) {  
65            System.out.println("Tampilan Tumpukan Barang Di Gudang :");  
66            for (int i = 0; i <= Top; i++) {  
67                System.out.printf("Kode M : %d, Kategori : %s\n", Tumpukan[i].Kode, Tumpukan[i].Kategori);  
68            }  
69        } else {  
70            System.out.println("Tumpukan Barang Kosong");  
71        }  
72    }  
73 }
```



NAMA : Alexander Agung Raya
NIM : 2341720040
NO ABSEN : 03
KELAS : 1F
MATERI : Stack

Utama03 :

```
1 package P8;  
2 import java.util.Scanner;  
3  
4 public class Utama03 {  
5     public static void main(String[] args) {  
6         Gudang03 Gudang = new Gudang03(7);  
7         Scanner sc = new Scanner(System.in);  
8         while (true) {  
9             System.out.println("\nMenu :");  
10            System.out.println("1. Tambah Barang");  
11            System.out.println("2. Ambil Barang");  
12            System.out.println("3. Tampilkan Tumpukan Barang");  
13            System.out.println("4. EXIT");  
14            System.out.print("Pilihlah menu :");  
15            int pilih = sc.nextInt();  
16            sc.nextLine();  
17            switch (pilih) {  
18                case 1:  
19                    System.out.print("Masukan Kode Barang : ");  
20                    int kode = sc.nextInt();  
21                    sc.nextLine();  
22                    System.out.print("Masukan Nama Barang : ");  
23                    String Nama = sc.nextLine();  
24                    System.out.print("Masukan Nama Kategori : ");  
25                    String kategori = sc.nextLine();  
26                    Barang03 BarangBaru = new Barang03(kode, Nama, kategori);  
27                    Gudang.TambahBarang(BarangBaru);  
28                    break;  
29                case 2:  
30                    Gudang.AmbilBarang();  
31                    break;  
32                case 3:  
33                    Gudang.TampilkanBarang();  
34                    break;  
35                case 4:  
36                    System.exit(0);  
37                    break;  
38                default:  
39                    System.out.println("Pilihan yang anda masukan salah");  
40            }  
41        }  
42    }  
43 }  
44 }
```

Hasil :

```
Masukan Nama Barang : tikus  
Masukan Nama Kategori : hewan  
Barang tikus berhasil Di Tambahkan  
  
Menu :  
1. Tambah Barang  
2. Ambil Barang  
3. Tampilkan Tumpukan Barang  
4. Tampilkan Barang Teratas  
5. EXIT  
Pilihlah menu : 3  
Rincian Tumpukan Barang Di gudang  
Kode 2 : tikus (kategori hewan)  
Kode 1 : kucing (kategori hewan)  
  
Menu :  
1. Tambah Barang  
2. Ambil Barang  
3. Tampilkan Tumpukan Barang  
4. Tampilkan Barang Teratas  
5. EXIT  
Pilihlah menu : 2  
Barang tikus Diambil Dari Gudang.
```

Pertanyaan :

1. Lakukan perbaikan pada kode program, sehingga keluaran yang dihasilkan sama dengan verifikasi hasil percobaan! Bagian mana saja yang perlu diperbaiki?

Jawab :

```
1 public void TampilkanBarang() {  
2     if (!isEmpty()) {  
3         System.out.println("Rincian Tumpukan Barang Di gudang :");  
4         for (int i = Top; i >= 0; i--) {  
5             System.out.print("Kode No : " + kategori[i] + ", tumpukan[" + i + "] Nama, tumpukan[" + i + "] kategori);  
6         }  
7     } else {  
8         System.out.println("Tumpukan Barang Kosong");  
9     }  
10 }
```

2. Berapa banyak data barang yang dapat ditampung di dalam tumpukan? Tunjukkan potongan kode Programnya!

Jawab :

Gudang03 Gudang = new Gudang03(7);



NAMA : Alexander Agung Raya
NIM : 2341720040
NO ABSEN : 03
KELAS : 1F
MATERI : Stack

3. Mengapa perlu pengecekan kondisi !cekKosong() pada method tampilkanBarang? Kalau kondisi tersebut dihapus, apa dampaknya?

Jawab :

Jika kondisi !CekKosong() dihapus, dampaknya adalah bahwa program akan mencoba untuk menampilkan detail barang tanpa memeriksa apakah tumpukan barang kosong atau tidak. Ini bisa menyebabkan error saat mencoba mengakses elemen array tumpukan dengan indeks negatif atau mengakses indeks yang tidak valid jika tumpukan kosong.

4. Modifikasi kode program pada class Utama sehingga pengguna juga dapat memilih operasi lihat barang teratas, serta dapat secara bebas menentukan kapasitas gudang!

Jawab :

```
Menu :
1. Tambah Barang
2. Ambil Barang
3. Tampilkan Tumpukan Barang
4. Tampilkan Barang Teratas
5. EXIT
Pilihlah menu :3
Rincian Tumpukan Barang Di gudang
Kode 3 : mouse (kategori mouse)
Kode 2 : bebek (kategori bebek)
Kode 1 : ayam (kategori ayam)

Menu :
1. Tambah Barang
2. Ambil Barang
3. Tampilkan Tumpukan Barang
4. Tampilkan Barang Teratas
5. EXIT
Pilihlah menu :4
Barang teratas : mouse
```

5. Commit dan push kode program ke Github

Percobaan 2 :

Gudang03 :

```
1 // Gudang03.java
2 // Program untuk mengelola gudang barang menggunakan tumpukan.
3 // Author: Alexander Agung Raya
4 // NIM: 2341720040
5 // NO ABSEN: 03
6 // KELAS: 1F
7 // MATERI: Stack
8
9 import java.util.*;
10
11 public class Gudang03 {
12     private static Stack<String> gudang = new Stack<>();
13     private static int kapasitas = 10;
14
15     public static void main(String[] args) {
16         Scanner scanner = new Scanner(System.in);
17         int pilihan;
18
19         do {
20             System.out.println("Menu:");
21             System.out.println("1. Tambah Barang");
22             System.out.println("2. Ambil Barang");
23             System.out.println("3. Tampilkan Tumpukan Barang");
24             System.out.println("4. Tampilkan Barang Teratas");
25             System.out.println("5. EXIT");
26
27             System.out.print("Pilihlah menu: ");
28             pilihan = scanner.nextInt();
29
30             switch (pilihan) {
31                 case 1:
32                     System.out.print("Masukkan nama barang: ");
33                     String barang = scanner.next();
34                     if (gudang.size() < kapasitas) {
35                         gudang.push(barang);
36                         System.out.println("Barang berhasil ditambahkan.");
37                     } else {
38                         System.out.println("Gudang penuh. Tidak dapat ditambahkan.");
39                     }
40                     break;
41                 case 2:
42                     System.out.print("Masukkan nama barang: ");
43                     String barang = scanner.next();
44                     if (gudang.contains(barang)) {
45                         gudang.remove(gudang.indexOf(barang));
46                         System.out.println("Barang berhasil dihapus.");
47                     } else {
48                         System.out.println("Barang tidak ditemukan di gudang.");
49                     }
50                     break;
51                 case 3:
52                     System.out.println("Tumpukan Barang di Gudang:");
53                     if (gudang.isEmpty()) {
54                         System.out.println("Gudang kosong.");
55                     } else {
56                         for (String barang : gudang) {
57                             System.out.println(barang);
58                         }
59                     }
60                     break;
61                 case 4:
62                     System.out.println("Barang Teratas:");
63                     if (gudang.isEmpty()) {
64                         System.out.println("Gudang kosong.");
65                     } else {
66                         System.out.println(gudang.peek());
67                     }
68                     break;
69                 case 5:
70                     System.out.println("EXIT");
71                     break;
72             }
73
74             System.out.print("Pilihlah menu: ");
75             pilihan = scanner.nextInt();
76         } while (pilihan != 5);
77
78         scanner.close();
79     }
80 }
```



NAMA : Alexander Agung Raya
NIM : 2341720040
NO ABSEN : 03
KELAS : 1F
MATERI : Stack

StackKonversi :

```
1 package P8;
2
3 public class StackKonversi {
4     static int tumpukanBiner[] ;
5     static int Size;
6     static int Top;
7
8     public StackKonversi (int Kapasitas) {
9         Size = Kapasitas;
10        tumpukanBiner = new int[Size];
11        Top = -1;
12    }
13    public Boolean CekKosong () {
14        return Top == -1;
15    }
16    public boolean isfull() {
17        return Top==Size-1;
18    }
19    }
20    public void push(int data) {
21        if (isfull()) {
22            System.out.println("Stack penuh ");
23        } else {
24            Top++;
25            tumpukanBiner[Top] = data;
26        }
27    }
28    public int pop() {
29        if (CekKosong()) {
30            System.out.println("Stack Kosong ");
31            return -1;
32        } else {
33            int data = tumpukanBiner[Top];
34            Top--;
35            return data;
36        }
37    }
38 }
```

Hasil :

```
Masukan Nama Barang : tikus
Masukan Nama Kategori : hewan
Barang tikus berhasil Di tambahkan
```

Menu :

1. Tambah Barang
2. Ambil Barang
3. Tampilkan Tumpukan Barang
4. Tampilkan Barang Teratas
5. EXIT

Pilihlah menu :3

Rincian Tumpukan Barang Di gudang

Kode 2 : tikus (kategori hewan)

Kode 1 : kucing (kategori hewan)

Menu :

1. Tambah Barang
2. Ambil Barang
3. Tampilkan Tumpukan Barang
4. Tampilkan Barang Teratas
5. EXIT

Pilihlah menu :2

Barang tikusDiambil Dari Gudang.

Kode unik dalam biner : 10



NAMA : Alexander Agung Raya
NIM : 2341720040
NO ABSEN : 03
KELAS : 1F
MATERI : Stack

Pertanyaan :

1. Pada method konversiDesimalKeBiner, ubah kondisi perulangan menjadi while (kode != 0), bagaimana hasilnya? Jelaskan alasannya!

Jawab :

```
Kode 1 : kucing (kategori hewan)
Menu :
1. Tambah Barang
2. Ambil Barang
3. Tampilkan Tumpukan Barang
4. Tampilkan Barang Teratas
5. EXIT
pilihlah menu :2
Barang tikusdiambil Dari Gudang.
kode unik dalam biner : 10
```

Alasannya adalah bahwa dalam konversi desimal ke biner, kita ingin terus membagi bilangan desimal oleh 2 sampai bilangan desimal tersebut menjadi 0. Oleh karena itu, kita menggunakan kondisi while (kode != 0) karena proses konversi harus terus berlanjut sampai nilai kode mencapai 0.

Ketika kode menjadi 0, itu berarti tidak ada sisa pembagian yang akan memberikan kontribusi terhadap digit biner yang dihasilkan. Oleh karena itu, tidak perlu lagi melanjutkan proses konversi dan loop berhenti.

2. Jelaskan alur kerja dari method konversiDesimalKeBiner!

Jawab :

Membuat objek stack dari kelas StackKonversi, yang akan digunakan untuk menyimpan sisa-sisa dari pembagian bilangan desimal dengan 2.

Selama nilai kode tidak sama dengan 0, lakukan langkah-langkah berikut:

- a. Hitung sisa pembagian (sisa) dari kode dibagi 2 menggunakan operator modulo %.
- b. Tambahkan sisa ke dalam stack menggunakan metode push.
- c. Bagi nilai kode dengan 2 untuk mendapatkan nilai yang lebih kecil dan proses berulang. Setelah selesai mengisi stack dengan sisa-sisa pembagian, mulai dari digit paling rendah, hapus sisa-sisa tersebut dari stack dan tambahkan ke dalam string biner.

Kembalikan string biner yang berisi representasi biner dari bilangan desimal awal.



NAMA : Alexander Agung Raya
NIM : 2341720040
NO ABSEN : 03
KELAS : 1F
MATERI : Stack

Percobaan 3 :

Postif03 :

```
1 package P8;
2 import java.util.Scanner;
3
4 public class Postfix03 {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         String P;
8         System.out.println("Masukan ekspresi matematika (Infix) :");
9         String Q = sc.nextLine();
10        Q = Q.trim();
11        Q = Q + " ";
12        int total = Q.length();
13        Postfix03 post = new Postfix03(total);
14        P = post.konversi(Q);
15        System.out.println("Postfix: " + P);
16    }
17 }
18
```

Postifmain03 :

```
1 package P8;
2 import java.util.Scanner;
3
4 public class PostfixMain03 {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         String P;
8         System.out.println("Masukan ekspresi matematika (Infix) :");
9         String Q = sc.nextLine();
10        Q = Q.trim();
11        Q = Q + " ";
12        int total = Q.length();
13        Postfix03 post = new Postfix03(total);
14        P = post.konversi(Q);
15        System.out.println("Postfix: " + P);
16    }
17 }
18
```

Hasil :

```
Masukan ekspresi matematika (Infix) :
a+b*(c+d-e)/f
Postfix: abcd+e-*f/+
```



NAMA : Alexander Agung Raya
NIM : 2341720040
NO ABSEN : 03
KELAS : 1F
MATERI : Stack

Pertanyaan :

1. Pada method derajat, mengapa return value beberapa case bernilai sama? Apabila return value diubah dengan nilai berbeda-beda setiap case-nya, apa yang terjadi?

Jawab :

Pada dasarnya, nilai kembalian dari method `derajat(char c)` menentukan prioritas operator dalam ekspresi matematika. Ketika beberapa operator memiliki nilai kembalian yang sama, itu menunjukkan bahwa mereka memiliki tingkat prioritas yang sama dan harus dievaluasi dalam urutan yang sama dalam ekspresi matematika. Jika nilai kembalian diubah sehingga setiap operator memiliki prioritas yang berbeda, itu akan mengubah urutan evaluasi ekspresi matematika, yang dapat menghasilkan hasil yang berbeda dari yang diharapkan, karena aturan konvensional untuk prioritas operator tidak diikuti. Ini menyoroti pentingnya menjaga konsistensi dalam menentukan tingkat prioritas operator agar ekspresi matematika dievaluasi dengan benar.

2. Jelaskan alur kerja method konversi!

Jawab :

Berikut adalah alur kerjanya :

Membuat string kosong P yang akan menyimpan ekspresi postfix hasil konversi.
Menginisialisasi variabel c untuk menyimpan karakter saat ini yang sedang diiterasi.

Melakukan iterasi melalui setiap karakter dalam string masukan Q.

Untuk setiap karakter:

- a. Jika karakter adalah operand (bilangan atau variabel), tambahkan karakter tersebut ke string P.
- b. Jika karakter adalah tanda kurung buka (, dorong karakter tersebut ke dalam stack.
- c. Jika karakter adalah tanda kurung tutup), lakukan langkah-langkah berikut:
Pop karakter dari stack dan tambahkan ke string P sampai menemukan tanda kurung buka (.

Pop tanda kurung buka dari stack.

- d. Jika karakter adalah operator, lakukan langkah-langkah berikut:

Selama prioritas operator di tumpukan lebih besar atau sama dengan prioritas operator saat ini (c), pop operator dari tumpukan dan tambahkan ke string P.

Dorong operator saat ini (c) ke dalam tumpukan.

Setelah selesai iterasi, kembalikan string P yang berisi ekspresi postfix hasil konversi.

3. Pada method konversi, apa fungsi dari potongan kode berikut?

`C = Q.charAt(i);`

Jawab :

Kode ini digunakan dalam proses iterasi melalui setiap karakter dalam string Q untuk menganalisis dan memproses ekspresi matematika yang dimasukkan oleh pengguna, seperti mendeteksi operand, operator, dan tanda kurung. Setelah karakter tersebut diambil, kode selanjutnya akan mengevaluasi karakter tersebut



NAMA : Alexander Agung Raya
NIM : 2341720040
NO ABSEN : 03
KELAS : 1F
MATERI : Stack

untuk mengambil tindakan yang sesuai, seperti menambahkan operand ke ekspresi postfix atau memproses operator dan tanda kurung.

Latihan :

Perhatikan dan gunakan kembali kode program pada Percobaan 1. Tambahkan dua method berikut pada class Gudang:

- Method lihatBarangTerbawah digunakan untuk mengecek barang pada tumpukan terbawah
- Method cariBarang digunakan untuk mencari ada atau tidaknya barang berdasarkan kode barangnya atau nama barangnya

Jawab :

```
1 import java.util.Scanner;
2 import java.util.Stack;
3
4 public class Gudang {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7         Scanner sc = new Scanner(System.in);
8         while (true) {
9             System.out.println("Menu :");
10            System.out.println("1. Tambah Barang");
11            System.out.println("2. Ambil Barang");
12            System.out.println("3. Tampilkan Tumpukan Barang");
13            System.out.println("4. Tampilkan Barang Teratas");
14            System.out.println("5. Tampilkan Barang Terbawah");
15            System.out.println("6. Cari Barang");
16            System.out.println("7. EXIT");
17            System.out.print("Pilihlah menu :");
18            int pilih = sc.nextInt();
19            switch (pilih) {
20                case 1:
21                    System.out.println("Masukan Nama Barang :");
22                    String kode = sc.next();
23                    System.out.println("Masukan Nama Barang :");
24                    String nama = sc.next();
25                    System.out.println("Masukan Kode Barang :");
26                    String kategori = sc.next();
27                    Barang b = new Barang(kode, nama, kategori);
28                    Gudang.tambahBarang(b);
29                    break;
30                case 2:
31                    Gudang.amilBarang();
32                    break;
33                case 3:
34                    Gudang.tampilkanBarang();
35                    break;
36                case 4:
37                    Gudang.lihatBarangTeratas();
38                    break;
39                case 5:
40                    Gudang.lihatBarangTerbawah();
41                    break;
42                case 6:
43                    System.out.println("Masukan Nama Barang :");
44                    String kodeBarang = sc.next();
45                    Barang b = Gudang.cariBarang(kodeBarang);
46                    if (b == null) {
47                        System.out.println("Barang tidak ditemukan");
48                    } else {
49                        System.out.println("Barang ditemukan");
50                        System.out.println("Kode barang : " + b.kode);
51                        System.out.println("Nama barang : " + b.nama);
52                        System.out.println("Kategori : " + b.kategori);
53                    }
54                    break;
55                case 7:
56                    System.exit(0);
57                    break;
58                default:
59                    System.out.println("Pilihlah yang ada dalam daftar");
60            }
61        }
62    }
63 }
```

Hasil :

```
6. Cari Barang
7. EXIT
Pilihlah menu :4
Barang Teratas : burung

Menu :
1. Tambah Barang
2. Ambil Barang
3. Tampilkan Tumpukan Barang
4. Tampilkan Barang Teratas
5. Tampilkan Barang Terbawah
6. Cari Barang
7. EXIT
Pilihlah menu :5
Barang Terbawah : kucing

Menu :
1. Tambah Barang
2. Ambil Barang
3. Tampilkan Tumpukan Barang
4. Tampilkan Barang Teratas
5. Tampilkan Barang Terbawah
6. Cari Barang
7. EXIT
Pilihlah menu :6
Masukan Nama Barang : kucing
Barang Ditemukan
Kode Barang : 1
Nama Barang : kucing
Kategori : kucing
```




NAMA : Alexander Agung Raya
NIM : 2341720040
NO ABSEN : 03
KELAS : 1F
MATERI : Stack