



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET VII - STACK

PRAKTIKUM 7 - STACK

Percobaan 1: Penyimpanan Tumpukan Barang dalam Gudang Waktu Percobaan : 90 Menit

```
1 package Jobsheet_7.Percobaan1;
2
3 public class Gudang07 {
4     Barang07 tumpukan[];
5     int size, top;
6
7     public Gudang07(int kapasitas) {
8         size = kapasitas;
9         tumpukan = new Barang07[size];
10        top = -1;
11    }
12    public boolean cekKosong() {
13        if (top == -1) {
14            return true;
15        } else {
16            return false;
17        }
18    }
19    public boolean cekPenuh() {
20        if (top == size - 1) {
21            return true;
22        } else {
23            return false;
24        }
25    }
26    public void tambahBarang(Barang07 brg) {
27        if (!cekPenuh()) {
28            top++;
29            tumpukan[top] = brg;
30            System.out.println("Barang " + brg.nama + " berhasil ditambahkan ke Gudang");
31        } else {
32            System.out.println("Gagal! Tumpukan barang di Gudang sudah penuh");
33        }
34    }
35 }
```

```
1 public Barang07 ambilBarang() {
2     if (!cekKosong()) {
3         Barang07 delete = tumpukan[top];
4         top--;
5         System.out.println("Barang " + delete.nama + " diambil dari Gudang");
6         return delete;
7     } else {
8         System.out.println("Tumpukan barang kosong");
9         return null;
10    }
11 }
12 public Barang07 lihatBarangTeratas() {
13     if (!cekKosong()) { // perbaikan
14         Barang07 barangTeratas = tumpukan[top];
15         System.out.println("Barang teratas : " + barangTeratas.nama);
16         return barangTeratas;
17     } else {
18         System.out.println("Tumpukan barang kosong.");
19         return null;
20    }
21 }
22 public void tampilkanBarang() {
23     if (!cekKosong()) {
24         System.out.println("Rincian tumpukan barang di Gudang : ");
25         for (int i = top; i >= 0; i--) { // perbaikan
26             System.out.printf("Kode %d : %s (Kategori %s)\n", tumpukan[i].kode,
27                 tumpukan[i].nama, tumpukan[i].kategori);
28         }
29     } else {
30         System.out.println("Tumpukan barang kosong.");
31     }
32 }
33 }
34 }
```

```
1 package Jobsheet_7.Percobaan1;
2 import java.util.Scanner;
3
4 public class Utama07 {
5     public static void main(String[] args) {
6         Scanner ayleen07 = new Scanner(System.in);
7         Gudang07 gudang07 = new Gudang07(7);
8
9         while (true) {
10            System.out.println("\nMenu : ");
11            System.out.println("1. Tambah barang");
12            System.out.println("2. Ambil barang");
13            System.out.println("3. Tampilkan tumpukan barang");
14            System.out.println("4. Keluar");
15            System.out.print("Pilih operasi : ");
16            int pilihan = ayleen07.nextInt();
17            ayleen07.nextLine();
18
19            switch (pilihan) {
20                case 1:
21                    System.out.print("Masukkan kode barang : ");
22                    int kode = ayleen07.nextInt();
23                    ayleen07.nextLine();
24                    System.out.print("Masukkan nama barang : ");
25                    String nama = ayleen07.nextLine();
26                    System.out.print("Masukkan nama kategori : ");
27                    String kategori = ayleen07.nextLine();
28
29                    Barang07 barangBaru = new Barang07(kode, nama, kategori);
30                    gudang07.tambahBarang(barangBaru);
31                    break;
32                case 2:
33                    gudang07.ambilBarang();
34                    break;
35                case 3:
36                    gudang07.tampilkanBarang();
37                    break;
38                case 4:
39                    break;
40                default:
41                    System.out.println("Pilihan tidak valid. Silahkan coba lagi.");
42                    break;
43            }
44        }
45    }
46 }
```



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET VII - STACK

```
1 package Jobsheet_7.Percobaan1;  
2  
3 public class Barang07 {  
4     String nama, kategori;  
5     int kode;  
6  
7     public Barang07 (int kode, String nama, String kategori) {  
8         this.kode = kode;  
9         this.nama = nama;  
10        this.kategori = kategori;  
11    }  
12 }
```

Hasil run program

```
Menu :  
1. Tambah barang  
2. Ambil barang  
3. Tampilkan tumpukan barang  
4. Keluar  
Pilih operasi : 1  
Masukkan kode barang : 21  
Masukkan nama barang : Majalah  
Masukkan nama kategori : Buku  
Barang Majalah berhasil ditambahkan ke Gudang
```

```
Menu :  
1. Tambah barang  
2. Ambil barang  
3. Tampilkan tumpukan barang  
4. Keluar  
Pilih operasi : 1  
Masukkan kode barang : 26  
Masukkan nama barang : Jaket  
Masukkan nama kategori : Pakaian  
Barang Jaket berhasil ditambahkan ke Gudang
```

```
Menu :  
1. Tambah barang  
2. Ambil barang  
3. Tampilkan tumpukan barang  
4. Keluar  
Pilih operasi : 2  
Barang Jaket diambil dari Gudang
```

```
Menu :  
1. Tambah barang  
2. Ambil barang  
3. Tampilkan tumpukan barang  
4. Keluar  
Pilih operasi : 1  
Masukkan kode barang : 33  
Masukkan nama barang : Pizza  
Masukkan nama kategori : Makanan  
Barang Pizza berhasil ditambahkan ke Gudang
```

```
Menu :  
1. Tambah barang  
2. Ambil barang  
3. Tampilkan tumpukan barang  
4. Keluar  
Pilih operasi : 3  
Rincian tumpukan barang di Gudang :  
Kode 33 : Pizza (Kategori Makanan)  
Kode 21 : Majalah (Kategori Buku)
```

```
Menu :  
1. Tambah barang  
2. Ambil barang  
3. Tampilkan tumpukan barang  
4. Keluar  
Pilih operasi : 4
```

Pertanyaan!

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

Jawab : Bagian kode yang salah ditunjukkan pada baris 2 dan baris 16 pada gambar kode dibawah ini, untuk perbaikan kode ditunjukkan pada comment :



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET VII - STACK

```
1 public Barang07 lihatBarangTeratas() {  
2     if (!isEmpty()) {  
3         // seharusnya !cekKosong  
4         Barang07 barangTeratas = tumpukan[top];  
5         System.out.println("Barang teratas : " + barangTeratas.nama);  
6         return barangTeratas;  
7     } else {  
8         System.out.println("Tumpukan barang kosong.");  
9         return null;  
10    }  
11 }  
12 public void tampilkanBarang() {  
13     if(!cekKosong()) {  
14         System.out.println("Rincian tumpukan barang di Gudang : ");  
15         // seharusnya for (int i = top; i >= 0; i--)  
16         for (int i = 0; i <= top; i++) {  
17             System.out.printf("Kode %d : %s (Kategori %s)\n", tumpukan[i].kode,  
18                 tumpukan[i].nama, tumpukan[i].kategori);  
19         }  
20     } else {  
21         System.out.println("Tumpukan barang kosong.");  
22     }  
23 }
```

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

Jawab : Banyak data barang yang ditampung di dalam tumpukan ditentukan oleh nilai kapasitas yang diberikan saat membuat objek Gudang07. Kapasitas tumpukan diinisialisasi dengan nilai 7 pada saat membuat objek Gudang07. Sehingga kapasitas tumpukan adalah 7.

```
1 public Gudang07(int kapasitas) {  
2     size = kapasitas;  
3     tumpukan = new Barang07[size];  
4     top = -1;  
5 }
```

```
1 Gudang07 gudang07 = new Gudang07(7);
```

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

Jawab : Pengecekan kondisi !cekKosong() pada method tampilkanBarang() dilakukan untuk memastikan bahwa tumpukan tidak kosong sebelum mencoba menampilkan isi tumpukan dan mengambil barang tumpukan. Jika kondisi dihapus, maka tidak ada data barang yang dapat ditampilkan sehingga terjadi error saat dijalankan.

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

Jawab :



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET VII - STACK

```
1 Scanner ayleen07 = new Scanner(System.in);  
2  
3 System.out.print("Masukkan kapasitas gudang : "); // Percobaan 1 - Pertanyaan 4  
4 int kapasitas = ayleen07.nextInt();  
5 ayleen07.nextLine();  
6  
7 Gudang07 gudang07 = new Gudang07(kapasitas);
```

```
1 case 3:  
2     gudang07.tampilkanBarang();  
3     break;  
4 case 4:  
5     gudang07.lihatBarangTeratas(); // Percobaan 1 - Pertanyaan 4  
6     break;  
7 case 5:  
8     break;
```

Hasil run program

```
Menu :  
1. Tambah barang  
2. Ambil barang  
3. Tampilkan tumpukan barang  
4. Lihat barang teratas  
5. Keluar  
Pilih operasi : 4  
Barang teratas : Jaket  
  
Menu :  
1. Tambah barang  
2. Ambil barang  
3. Tampilkan tumpukan barang  
4. Lihat barang teratas  
5. Keluar  
Pilih operasi : 3  
Rincian tumpukan barang di Gudang :  
Kode 26 : Jaket (Kategori Pakaian)  
Kode 21 : Majalah (Kategori Buku)
```

5. Commit dan push kode program ke Github

Jawab :

P8_Percobaan 1_Pertanyaan 4_07 ayleenrq committed now	b93a4d2		
P8_Percobaan 1_07 FIX ayleenrq committed 11 minutes ago	a1ed801		
P8_Percobaan 1_07 ayleenrq committed 16 minutes ago	8aa415b		



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET VII - STACK

Percobaan 2: Konversi Kode Barang ke Biner Waktu Percobaan: 30 Menit

```
1 package Jobsheet_7.Percobaan1;
2
3 public class StackKonversi07 {
4     int size;
5     int[] tumpukanBiner;
6     int top;
7
8     public StackKonversi07() {
9         this.size = 32;
10        tumpukanBiner = new int[size];
11        top = -1;
12    }
13    public boolean isEmpty() {
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 (isEmpty()) {
30            System.out.println("Stack kosong.");
31            return -1;
32        } else {
33            int data = tumpukanBiner[top];
34            top--;
35            return data;
36        }
37    }
38 }
```

```
1 public Barang07 ambilBarang() {
2     if (!cekKosong()) {
3         Barang07 delete = tumpukan[top];
4         top--;
5         System.out.println("Barang " + delete.nama + " diambil dari Gudang");
6         System.out.println("Kode unik dalam biner : " + konversiDesimalKeBiner(delete.kode));
7         return delete;
8     } else {
9         System.out.println("Tumpukan barang kosong");
10        return null;
11    }
12 }
```

```
1 public String konversiDesimalKeBiner(int kode) {
2     StackKonversi07 stack = new StackKonversi07();
3     while (kode > 0) {
4         int sisa = kode % 2;
5         stack.push(sisa);
6         kode = kode / 2;
7     }
8     String biner = new String();
9     while (!stack.isEmpty()) {
10        biner += stack.pop();
11    }
12    return biner;
13 }
```



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET VII - STACK

Hasil run program

```
Masukkan kapasitas gudang : 3

Menu :
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat barang teratas
5. Keluar
Pilih operasi : 1
Masukkan kode barang : 13
Masukkan nama barang : Setrika
Masukkan nama kategori : Elektronik
Barang Setrika berhasil ditambahkan ke Gudang

Menu :
1. Tambah barang
2. Ambil barang
3. Tampilkan tumpukan barang
4. Lihat barang teratas
5. Keluar
Pilih operasi : 2
Barang Setrika diambil dari Gudang
Kode unik dalam biner : 1101
```

Pertanyaan!

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

Jawab : Hasilnya akan tetap sama. Kondisi perulangan diubah menjadi while (kode != 0), maka proses konversi akan berlangsung selama nilai kode tidak sama dengan 0

2. Jelaskan alur kerja dari method konversiDesimalKeBiner!

Jawab :

1. Objek StackKonversi07 dibuat untuk menyimpan sisa pembagian.
2. Perulangan while dijalankan selama kode masih lebih besar dari 0. Pada setiap iterasi perulangan, nilai kode dibagi dengan 2 dan sisanya disimpan dalam variabel sisa.
3. Nilai sisa kemudian didorong (push) ke dalam tumpukan (stack) menggunakan method push().
4. Nilai kode diperbarui dengan hasil pembagian (kode / 2).
5. Setelah perulangan dilakukan, String biner dibuat untuk menyimpan hasil akhir.
6. Perulangan while baru dijalankan selama tumpukan tidak kosong. Pada setiap iterasi perulangan, nilai teratas tumpukan diambil menggunakan method pop() dan ditambahkan ke String biner.
7. Setelah perulangan selesai, String biner yang berisi biner dari nilai desimal dikembalikan.



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET VII - STACK

Percobaan 3: Konversi Notasi Infix ke Postfix Waktu Percobaan: 90 Menit

```
1 package Jobsheet_7.Percobaan1;
2
3 public class Postfix07 {
4     public int n, top;
5     public char[] stack;
6
7     public Postfix07(int total) {
8         n = total;
9         top = -1;
10        stack = new char[n];
11        push('(');
12    }
13    public void push(char c) {
14        top++;
15        stack[top] = c;
16    }
17    public char pop() {
18        char item = stack[top];
19        top--;
20        return item;
21    }
22    public boolean IsOperand(char c) {
23        if ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z') ||
24            (c >= '0' && c <= '9') || (c == '.' || c == ',')) {
25            return true;
26        } else {
27            return false;
28        }
29    }
30    public boolean IsOperator(char c) {
31        if (c == '^' || c == '%' || c == '/' || c == '*' ||
32            c == '-' || c == '+') {
33            return true;
34        } else {
35            return false;
36        }
37    }
38    public int derajat(char c) {
39        switch (c) {
40            case '^':
41                return 3;
```

```
1         case '%':
2             return 2;
3         case '/':
4             return 2;
5         case '*':
6             return 2;
7         case '-':
8             return 1;
9         case '+':
10            return 1;
11        default:
12            return 0;
13    }
14 }
15 public String konversi(String Q) {
16     String P = "";
17     char c;
18     for (int i = 0; i < n; i++) {
19         c = Q.charAt(i);
20         if (IsOperand(c)) {
21             P = P + c;
22         }
23         if (c == '(') {
24             push(c);
25         }
26         if (c == ')') {
27             while (stack[top] != '(') {
28                 P = P + pop();
29             }
30             pop();
31         }
32         if (IsOperator(c)) {
33             while (derajat(stack[top]) >= derajat(c)) {
34                 P = P + pop();
35             }
36             push(c);
37         }
38     }
39     return P;
40 }
41 }
```

```
1 package Jobsheet_7.Percobaan1;
2 import java.util.Scanner;
3
4 public class PostfixMain07 {
5     public static void main(String[] args) {
6         Scanner ayleen07 = new Scanner(System.in);
7         String P, Q;
8
9         System.out.println("Masukkan ekspresi matematika (infix) : ");
10        Q = ayleen07.nextLine();
11        Q = Q.trim();
12        P = Q + " ";
13
14        int total = Q.length();
15
16        Postfix07 post = new Postfix07(total);
17        P = post.konversi(Q);
18        System.out.println("Postfix : " + P);
19    }
20 }
```

Hasil run program



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET VII - STACK

```
n07'  
Masukkan ekspresi matematika (infix) :  
a+b*(c+d-e)/f  
Postfix : abcd+e-*f/+  
PS C:\Users\Admin\OneDrive - ypt.or.id\Documents\K  
ikum_AlgoritmaStrukturData_07> |
```

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 method derajat, beberapa case memiliki nilai return yang sama karena operator-operator tersebut memiliki prioritas yang setara. Nilai return digunakan untuk menentukan prioritas operasi matematika. Jika nilai return diubah maka urutan operasi akan berubah dan mempengaruhi hasil konversi menjadi postfix.

2. Jelaskan alur kerja method konversi!

Jawab :

1. Inisialisasi string P untuk menyimpan hasil konversi postfix.
2. Pada setiap iterasi, mengambil karakter dari Q pada indeks i dan disimpan dalam variabel c.
3. Jika c adalah operand (huruf, angka, spasi, atau titik), maka c akan ditambahkan ke string P.
4. Jika c adalah tanda kurung buka “(”, maka c akan di-push ke stack.
5. Jika c adalah tanda kurung tutup “)”, maka semua operand dan operator akan di-pop dari stack dan ditambahkan ke P sampai ditemukan tanda kurung buka “(”. Tanda kurung buka “(” juga di-pop dari stack.
6. Jika c adalah operator, maka semua operator dengan prioritas yang sama atau lebih tinggi akan di-pop dari stack dan ditambahkan ke P lalu c akan di-push ke stack.
7. Setelah semua karakter diproses, P akan berisi hasil konversi postfix.

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

```
c = Q.charAt(i);
```

Jawab : Potongan kode tersebut berfungsi untuk mengambil karakter pada indeks i dari string Q (ekspresi infix) dan disimpan dalam variabel c. Kemudian variabel c akan digunakan dalam proses konversi infix ke postfix pada iterasi saat ini.

Link GitHub Pribadi :

https://github.com/ayleenrq/Prak_AlgoritmaStrukturData_07.git