

Laporan Praktikum ASD Pertemuan 9

Nama: Gegas Anugrah Derajat

Kelas: SIB-1F

Percobaan 7.2

Membuat class, membuat atribut dan konstruktor

```
public class Stack11 {  
    int data[];  
    int size;  
    int top;  
  
    public Stack11(int size) {  
        this.size = size;  
        data = new int [size];  
        top = -1;  
    }  
}
```

Menambahkan method isFull() dan isEmpty()

```
public boolean isFull() {  
    if (top == size-1) {  
        return true;  
    } else {  
        return false;  
    }  
}  
  
public boolean isEmpty() {  
    if (top == -1) {  
        return true;  
    } else {  
        return false;  
    }  
}
```

Menambahkan method push() dan pop()

```
public void push(int dt) {  
    if (!isFull()) {  
        top++;  
        data[top] = dt;  
    } else {  
        System.out.println(x:"Stack penuh");  
    }  
}  
  
public void pop() {  
    if (!isEmpty()) {  
        int x = data[top];  
        top--;  
        System.out.println("Data yang dikeluarkan dari stack: " +x);  
    } else {  
        System.out.println(x:"Stack masih kosong");  
    }  
}
```

Menambahkan method peek(), print() dan clear()

```
public void peek() {  
    System.out.println("Elemen teratas stack: " + data[top]);  
}  
  
public void print() {  
    System.out.println(x: "Isi stack: ");  
    for (int i = top; i >= 0; i--) {  
        System.out.println(data[i] + " ");  
    }  
    System.out.println(x: "");  
}  
  
public void clear() {  
    if (!isEmpty()) {  
        for (int i = top; i >= 0; i--) {  
            top--;  
        }  
        System.out.println(x: "Stack sudah dikosongkan");  
    } else {  
        System.out.println(x: "Stack masih kosong");  
    }  
}
```

Membuat class StackDemo dan mengoperasikan method pada class stack

```
public class StackDemo11 {  
    Run | Debug  
    public static void main(String[] args) {  
        Stack11 stack = new Stack11 (size:10);  
        stack.push(dt:8);  
        stack.push (dt:12);  
        stack.push (dt:18);  
        stack.print ();  
        stack.pop ();  
        stack.peek ();  
        stack.pop ();  
        stack.push (-5);  
        stack.print ();  
    }  
}
```

Hasil

```
S:\Jobsheet9_709181C1\bin - StackDemo11  
Isi stack:  
18  
12  
8  
  
Data yang dikeluarkan dari stack: 18  
Elemen teratas stack: 12  
Data yang dikeluarkan dari stack: 12  
Isi stack:  
-5  
8
```

7.2.3 Pertanyaan

1. Pada method `pop()`, mengapa diperlukan pemanggilan method `isEmpty()`? Apa yang terjadi jika tidak ada pemanggilan `isEmpty()`?
2. Jelaskan perbedaan antara method `peek()` dengan method `pop()` pada class `Stack`.

Jawaban:

1. `isEmpty()` digunakan untuk mengecek apakah stack dalam keadaan kosong, jika stack kosong maka tidak akan bisa mengambil value. Jika method `isEmpty()` tidak ada maka tidak akan bisa mengetahui apakah stack dalam keadaan kosong atau tidak.

2. Method `peek()` digunakan untuk memeriksa elemen paling atas, sedangkan `pop()` digunakan untuk mengambil elemen pada stack paling atas.

Percobaan 7.3

Membuat class `Pakaian`, menambahkan atribut dan konstruktor

```
public class Pakaian {  
    String jenis, warna, merk, ukuran;  
    double harga;  
  
    Pakaian(String jenis, String warna, String merk, String ukuran, double harga) {  
        this.jenis = jenis;  
        this.warna = warna;  
        this.merk = merk;  
        this.ukuran = ukuran;  
        this.harga = harga;  
    }  
}
```

Membuat class `Stack`, menambahkan atribut dan konstruktor

```
public class Stack {  
    int size;  
    int top;  
    Pakaian data[];  
  
    public Stack(int size) {  
        this.size = size;  
        data = new Pakaian[size];  
        top = -1;  
    }  
}
```

Membuat method isEmpty() dan isFull()

```
public boolean isEmpty() {
    if (top == -1) {
        return true;
    } else {
        return false;
    }
}

public boolean isFull() {
    if (top == size - 1) {
        return true;
    } else {
        return false;
    }
}
```

Membuat method push()

```
public void push(Pakaian pkn) {
    if (!isFull()) {
        top++;
        data[top] = pkn;
    } else {
        System.out.println(x:"Isi stack penuh");
    }
}
```

Membuat method pop() dan peek()

```
public void pop() {
    if (!isEmpty()) {
        Pakaian x = data[top];
        top--;
        System.out.println(
            "Data yang keluar: " + x.jenis + " " + x.warna + " " + x.merk + " " + x.ukuran + " " + x.harga);
    } else {
        System.out.println(x:"Stack masih kosong");
    }
}

public void peek() {
    System.out.println("Elemen teratas: " + data[top].jenis + " " + data[top].warna + " " + data[top].merk + " "
        + data[top].ukuran + " " + data[top].harga);
}
```

Membuat method print() dan clear()

```
public void print() {
    System.out.println(x:"Isi stack: ");
    for (int i = top; i >= 0; i--) {
        System.out.println(data[i].jenis + " " + data[i].warna + " " + data[i].merk + " " + data[i].ukuran + " "
            + data[i].harga + " ");
    }
    System.out.println(x:"");
}

public void clear() {
    if (!isEmpty()) {
        for (int i = top; i >= 0; i--) {
            top--;
        }
        System.out.println(x:"Stack sudah dikosongkan");
    } else {
        System.out.println(x:"Stack masih kosong");
    }
}
```

Membuat class StackMain dan menambahkan kode input data pakaian dan melakukan pemanggilan method print, pop, dan peek

```
import java.util.Scanner;

public class StackMain {

    Run | Debug
    public static void main(String[] args) {
        Stack stk = new Stack(size:5);
        Scanner sc = new Scanner(System.in);
        char pilih;
        do {
            System.out.print(s:"Jenis: ");
            String jenis = sc.next();
            System.out.print(s:"Warna: ");
            String warna = sc.next();
            System.out.print(s:"Merk: ");
            String merk = sc.next();
            System.out.print(s:"Ukuran: ");
            String ukuran = sc.next();
            System.out.print(s:"Harga: ");
            double harga = sc.nextDouble();

            Pakaian p = new Pakaian(jenis, warna, merk, ukuran, harga);
            System.out.print(s:"Apakah anda akan menambahkan data baru ke stack (y/n)? ");
            pilih = sc.next().charAt(index:0);
            sc.nextLine();
            stk.push(p);

        } while (pilih == 'y');
        stk.print();
        stk.pop();
        stk.peek();
        stk.print();
        sc.close();
    }
}
```

Hasil

```
g:\Code\User\workspaceStorage\90b1e7471186d16486264002fff62
Jenis: kaos
Warna: hitam
Merk: nevada
Ukuran: m
Harga: 85000
Apakah anda akan menambahkan data baru ke stack (y/n)? y
Jenis: kemeja
Warna: putih
Merk: styves
Ukuran: xl
Harga: 127000
Apakah anda akan menambahkan data baru ke stack (y/n)? y
Jenis: celana
Warna: biru
Merk: levis
Ukuran: l
Harga: 189500
Apakah anda akan menambahkan data baru ke stack (y/n)? n
Isi stack:
celana biru levis l 189500.0
kemeja putih styves xl 127000.0
kaos hitam nevada m 85000.0

Data yang keluar: celana biru levis l 189500.0
Elemen teratas: kemeja putih styves xl 127000.0
Isi stack:
kemeja putih styves xl 127000.0
kaos hitam nevada m 85000.0
```

7.3.3. Pertanyaan

1. Berapa banyak data pakaian yang dapat ditampung di dalam stack? Tunjukkan potongan kode program untuk mendukung jawaban Anda tersebut!
2. Perhatikan class **StackMain**, pada saat memanggil fungsi push, parameter yang dikirimkan adalah **p**. Data apa yang tersimpan pada variabel **p** tersebut?

```
stk.push(p);
```

3. Apakah fungsi penggunaan **do-while** yang terdapat pada class **StackMain**?
4. Modifikasi kode program pada class **StackMain** sehingga pengguna dapat memilih operasi-operasi pada stack (push, pop, peek, atau print) melalui pilihan menu program dengan memanfaatkan kondisi IF-ELSE atau SWITCH-CASE!

Jawaban:

1. Ada 5 data yang bisa ditampung

```
Stack stk = new Stack(size:5);
```

2. Data yang telah diinputkan sebelumnya
3. Untuk mengulang input pakaian sampai var pilih bernilai n
- 4.

```
public static void main(String[] args) {
    Stack stk = new Stack(size:5);
    Scanner sc = new Scanner(System.in);
    StackMain main = new StackMain();

    int pil;
    while (true) {
        System.err.println(x:"=====");
        System.err.println(x:"Pilihan");
        System.err.println();
        System.out.println(x:"1. Push");
        System.out.println(x:"2. Pop");
        System.out.println(x:"3. Peek");
        System.out.println(x:"4. Print");
        System.out.println(x:"5. GetMax");
        System.out.println(x:"6. Keluar");
        System.out.print(s:"Pilih salah satu (1/2/3/4/5/6) : ");
        pil = sc.nextInt();
        switch (pil) {
            case 1:
                main.pushPakaian(stk);
                break;
            case 2:
                stk.pop();
                break;
            case 3:
                stk.peek();
                break;
            case 4:
                stk.print();
                break;
            case 5:
                stk.getMax();
                break;
            case 6:
                return;
            default:
                System.out.println(x:"Pilihan tidak tersedia. ");
                break;
        }
    }
    sc.close();
}
```

```
public void pushPakaian(Stack stk) {  
    char pilih;  
    do {  
        System.out.print(s:"Jenis: ");  
        String jenis = sc.next();  
        System.out.print(s:"Warna: ");  
        String warna = sc.next();  
        System.out.print(s:"Merk: ");  
        String merk = sc.next();  
        System.out.print(s:"Ukuran: ");  
        String ukuran = sc.next();  
        System.out.print(s:"Harga: ");  
        double harga = sc.nextDouble();  
  
        Pakaian p = new Pakaian(jenis, warna, merk, ukuran, harga);  
        System.out.print(s:"Apakah anda akan menambahkan data baru ke stack (y/n)? ");  
        pilih = sc.next().charAt(index:0);  
        sc.nextLine();  
        stk.push(p);  
    } while (pilih == 'y');  
    sc.close();  
}
```


Percobaan 7.4

Membuat class Postfix, memberikan atribut dan konstruktor

```
public class Postfix11 {  
    int n;  
    int top;  
    char[] stack;  
  
    public Postfix11(int total) {  
        n = total;  
        top = -1;  
        stack = new char[n];  
        push(c: '(');  
    }  
}
```

Membuat method push, pop, isOperand dan isOperator

```
public void push(char c) {  
    top++;  
    stack[top] = c;  
}  
  
public char pop() {  
    char item = stack[top];  
    top--;  
    return item;  
}  
  
public boolean isOperand(char c) {  
    if ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z') || (c >= '0' && c <= '9') || (c == ' ' || c == '.')) {  
        return true;  
    } else {  
        return false;  
    }  
}  
  
public boolean isOperator(char c) {  
    if (c == '^' || c == '%' || c == '/' || c == '*' || c == '-' || c == '+') {  
        return true;  
    } else {  
        return false;  
    }  
}
```

Membuat method derajat()

```
public int derajat (char c) {  
    switch (c) {  
        case '^':  
            return 3;  
        case '%':  
            return 2;  
        case '/':  
            return 2;  
        case '*':  
            return 2;  
        case '-':  
            return 1;  
        case '+':  
            return 1;  
        default:  
            return 0;  
    }  
}
```


Membuat method konversi()

```
public String konversi (String Q) {
    String P = "";
    char c;
    for (int i = 0; i < n; i++) {
        c = Q.charAt (i) ;
        if (isOperand(c) ) {
            P = P + c;
        }
        if (c == '(') {
            push (c) ;
        }
        if (c == ')') {
            while (stack[top] != '(') {
                P = P + pop ();
            }
            pop ();
        }
        if (isOperator(c)) {
            while (derajat(stack[top]) >= derajat (c)) {
                P = P + pop () ;
            }
            push (c) ;
        }
    }
    return P;
}
```

Membuat class PostfixMain

```
import java.util.Scanner;

public class PostfixMain11 {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String P, Q;
        System.out.print(s:"Masukkan ekspresi matematika (infix): ");
        Q = sc.next();
        Q = Q.trim();
        Q = Q + ")";

        int total = Q.length();
        Postfix11 post = new Postfix11(total);
        P = post.konversi(Q);
        System.out.println("Postfix: " + P);
        sc.close();
    }
}
```

Hasil

```
36336665_78767671(bin) - PostfixMain11
Masukkan ekspresi matematika (infix): a+b*(c+d-e)/f
Postfix: abcd+e-*f/+
```

7.4.3. Pertanyaan

1. Perhatikan class **Postfix**, jelaskan alur kerja method **derajat**!
2. Apa fungsi kode program berikut?

```
c = Q.charAt(i);
```
3. Jalankan kembali program tersebut, masukkan ekspresi **5*4^(1+2)%3**. Tampilkan hasilnya!
4. Pada soal nomor 3, mengapa tanda kurung tidak ditampilkan pada hasil konversi? Jelaskan!

Jawaban:

1. Pada saat pemanggilan method derajat akan membawa parameter berupa char. Character akan dilakukan pengecekan menggunakan switch case. pengecekan dilakukan berurutan hingga case terakhir, jika parameter sama dengan case maka akan mereturnkan angka sesuai case dan selanjutnya melakukan pengecekan pada case selanjutnya.

2. Untuk memanggil karakter urutan ke-i pada Q

3.

```
g:\code\user\workspace\storage\9001e7471180d1648626  
Masukkan ekspresi matematika (infix): 5*4^(1+2)%3  
Postfix: 5412+^*3%
```

4. Tanda kurung tidak diperlukan pada hasil konversi, karena sudah diatur urutannya pada konversi.

7.5. Tugas

1. Perhatikan dan gunakan kembali kode program pada **Praktikum 2**. Tambahkan method **getMax** pada class **Stack** yang digunakan untuk mencari dan menampilkan data pakaian dengan harga tertinggi dari semua data pakaian yang tersimpan di dalam stack!
2. Setiap hari Minggu, Dewi pergi berbelanja ke salah satu supermarket yang berada di area rumahnya. Setiap kali selesai berbelanja, Dewi menyimpan struk belanjanya di dalam laci. Setelah dua bulan, ternyata Dewi sudah mempunyai delapan struk belanja. Dewi berencana mengambil lima struk belanja untuk ditukarkan dengan voucher belanja.
Buat sebuah program stack untuk menyimpan data struk belanja Dewi, kemudian lakukan juga proses pengambilan data struk belanja sesuai dengan jumlah struk yang akan ditukarkan dengan voucher. Informasi yang tersimpan pada struk belanja terdiri dari:
 - Nomor transaksi
 - Tanggal pembelian
 - Jumlah barang yang dibeli
 - Total harga bayarTampilkan informasi struk belanja yang masih tersimpan di dalam stack!

Jawaban:

1. Menambahkan method getMax()

```
public void getMax() {
    double hargaMax = 0;
    int indexMax = -1;
    for (int i = top; i >= 0; i--) {
        if (data[i].harga > hargaMax) {
            hargaMax = data[i].harga;
            indexMax = i;
        }
    }
    System.out.println(data[indexMax].jenis + " " + data[indexMax].warna + " " + data[indexMax].merk + " "
        + data[indexMax].ukuran + " "
        + data[indexMax].harga + " ");
    System.out.println();
}
```

Melakukan perubahan pada class StackMain

```
int pil;
while (true) {
    System.err.println(x: "=====");
    System.err.println(x: "Pilihan");
    System.err.println();
    System.out.println(x: "1. Push");
    System.out.println(x: "2. Pop");
    System.out.println(x: "3. Peek");
    System.out.println(x: "4. Print");
    System.out.println(x: "5. GetMax");
    System.out.println(x: "6. Keluar");
    System.out.print(s: "Pilih salah satu (1/2/3/4/5/6) : ");
    pil = sc.nextInt();
    switch (pil) {
        case 1:
            main.pushPakaian(stk);
            break;
        case 2:
            stk.pop();
            break;
        case 3:
            stk.peek();
            break;
        case 4:
            stk.print();
            break;
        case 5:
            stk.getMax();
            break;
        case 6:
            return;
        default:
            System.out.println(x: "Pilihan tidak tersedia. ");
            break;
    }
}
```

Hasil

```
Pilihan
1. Push
2. Pop
3. Peek
4. Print
5. GetMax
6. Keluar
Pilih salah satu (1/2/3/4/5/6) : 1
Jenis: kaos
Warna: hitam
Merk: nevada
Ukuran: m
Harga: 85000
Apakah anda akan menambahkan data baru ke stack (y/n)? y
Jenis: celana
Warna: biru
Merk: levis
Ukuran: l
Harga: 189500
Apakah anda akan menambahkan data baru ke stack (y/n)? y
Jenis: kemeja
Warna: putih
Merk: styves
Ukuran: xl
Harga: 127000
Apakah anda akan menambahkan data baru ke stack (y/n)? n
```

```
Pilihan
1. Push
2. Pop
3. Peek
4. Print
5. GetMax
6. Keluar
Pilih salah satu (1/2/3/4/5/6) : 5
celana biru levis l 189500.0
```

```
=====
```

2. Membuat class Struk dan menambahkan method

```

public class Struk11 {
    int noTransaksi, jumlahBarang;
    int totalBayar;
    int size;
    int top;
    String tglBeli;
    Struk11 data[];
    Struk11[] stk;

    Struk11(int no, String tgl, int jb, int tb) {
        noTransaksi = no;
        tglBeli = tgl;
        jumlahBarang = jb;
        totalBayar = tb;
    }

    public Struk11(int size) {
        this.size = size;
        data = new Struk11[size];
        top = -1;
    }

    public boolean isEmpty() {
        return top == -1;
    }

    public boolean isFull() {
        return top == size - 1;
    }

    public void push(Struk11 dt) {
        if (!isFull()) {
            top++;
            data[top] = dt;
        } else {
            System.out.println("X: \"Isi Stack Penuh!\");
        }
    }
}

```

```

public void pop() {
    if (!isEmpty()) {
        Struk11 x = data[top];
        top--;
        System.out.println("Struk yang keluar: " + x.noTransaksi + " "
            + x.tglBeli + " " + x.jumlahBarang + " "
            + x.totalBayar + " ");
    } else {
        System.out.println(x:"Stack masih kosong");
    }
}

public void peek() {
    System.out.println("Elemen teratas: " + data[top].noTransaksi + " "
        + " " + data[top].tglBeli + " " + data[top].jumlahBarang
        + " " + data[top].totalBayar);
}

public void print() {
    System.out.println(x:"Isi stack: ");
    for (int i = top; i >= 0; i--) {
        System.out.println(data[i].noTransaksi + " " + data[i].tglBeli + " "
            + data[i].jumlahBarang + " " + data[i].totalBayar +
            " ");
    }
    System.out.println(x:"");
}

public void clear() {
    if (!isEmpty()) {
        for (int i = top; i >= 0; i--) {
            top--;
        }
        System.out.println(x:"Stack sudah dikosongkan");
    } else {
        System.out.println(x:"Gagal! Stack masih kosong");
    }
}

```

Membuat Class StrukMain dan menambahkan method

```

public class StrukMain11 {
    Scanner sc = new Scanner(System.in);

    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Struk11 stk = new Struk11(size:20);
        StrukMain11 main = new StrukMain11();

        int pil;
        while (true) {
            System.err.println(x:"=====");
            System.err.println(x:"Pilihan");
            System.err.println();
            System.out.println(x:"1. Push");
            System.out.println(x:"2. Pop");
            System.out.println(x:"3. Peek");
            System.out.println(x:"4. Print");
            System.out.println(x:"5. Kupon");
            System.out.println(x:"6. Keluar");
            System.out.print(s:"Pilih salah satu (1/2/3/4/5/6) : ");
            pil = sc.nextInt();
            switch (pil) {
                case 1:
                    main.pushPakaian(stk);
                    break;
                case 2:
                    stk.pop();
                    break;
                case 3:
                    stk.peek();
                    break;
                case 4:
                    stk.print();
                    break;
                case 5:
                    main.kupon(stk);
                    break;
                case 6:
                    return;
                default:
                    System.out.println(x:"Pilihan tidak tersedia. ");
                    break;
            }
        }
    }
}

```



```

public void pushPakaian(Struk11 stk) {
    char pilih;
    do {
        System.out.print(s:"No Belanja: ");
        int noTra = sc.nextInt();
        System.out.print(s:"Tanggal (dd/mm/yyyy) : ");
        String tanggal = sc.next();
        System.out.print(s:"Jumlah: ");
        int jumlah = sc.nextInt();
        System.out.print(s:"Total Bayar: ");
        int total = sc.nextInt();

        Struk11 d = new Struk11(noTra, tanggal, jumlah, total);

        System.out.print(s:"Apakah Anda akan menambahkan data baru ke stack (y/n) ? ");
        pilih = sc.next().charAt(index:0);
        stk.push(d);
    } while (pilih == 'y');
}

public void kupon(Struk11 stk) {
    System.out.println(x:"Struk yang diambil");
    int i = 0;
    while (i < 5) {
        stk.pop();
        i++;
    }

    System.out.println();
    System.out.println(x:"Data yang tersisa");
    stk.print();
}
}

```

Hasil

```

Pilihan
1. Push
2. Pop
3. Peek
4. Print
5. Kupon
6. Keluar
Pilih salah satu (1/2/3/4/5/6) : 1
No Belanja: 12
Tanggal (dd/mm/yyyy) : 12/12/2323
Jumlah: 4
Total Bayar: 30400
Apakah Anda akan menambahkan data baru ke stack (y/n) ? y
No Belanja: 33
Tanggal (dd/mm/yyyy) : 12/2/2323
Jumlah: 3
Total Bayar: 40000
Apakah Anda akan menambahkan data baru ke stack (y/n) ? y
No Belanja: 21
Tanggal (dd/mm/yyyy) : 12/3/2323
Jumlah: 4
Total Bayar: 40000
Apakah Anda akan menambahkan data baru ke stack (y/n) ? y
No Belanja: 332
Tanggal (dd/mm/yyyy) : 23/12/1223
Jumlah: 45
Total Bayar: 1000000
Apakah Anda akan menambahkan data baru ke stack (y/n) ? y
No Belanja: 25
Tanggal (dd/mm/yyyy) : 13/12/2324
Jumlah: 4
Total Bayar: 120000

```

```

Total Bayar: 120000
Apakah Anda akan menambahkan data baru ke stack (y/n) ? y
No Belanja: 32
Tanggal (dd/mm/yyyy) : 13/12/2322
Jumlah: 5
Total Bayar: 120000
Apakah Anda akan menambahkan data baru ke stack (y/n) ? y
No Belanja: 55
Tanggal (dd/mm/yyyy) : 23/01/2425
Jumlah: 6
Total Bayar: 100000
Apakah Anda akan menambahkan data baru ke stack (y/n) ? y
No Belanja: 33
Tanggal (dd/mm/yyyy) : 13/04/2024
Jumlah: 6
Total Bayar: 45500
Apakah Anda akan menambahkan data baru ke stack (y/n) ? n
=====
Pilihan
1. Push
2. Pop
3. Peek
4. Print
5. Kupon
6. Keluar
Pilih salah satu (1/2/3/4/5/6) : 5
Data yang diambil
Data yang keluar: 33 13/04/2024 6 45500
Data yang keluar: 55 23/01/2425 6 100000
Data yang keluar: 32 13/12/2322 5 120000
Data yang keluar: 25 13/12/2324 4 120000
Data yang keluar: 332 23/12/1223 45 1000000

Data yang tersisa
Isi stack:
21 12/3/2323 4 40000
33 12/2/2323 3 40000
12 12/12/2323 4 30400

```