

# **LAPORAN PRAKTIKUM MINGGU KE-8**

**“[Stack]”**



Disusun oleh:

**[Daffa Aqila Rahmatullah]**

**[2041720098]**

**D4 TEKNIK INFORMATIKA  
TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG**

**2021**

# **LAPORAN**

## **“PRAKTIKUM”**

### **A. KODE PROGRAM**

#### **7.2.1 Langkah-langkah Percobaan**

```
public class Stack {  
    int size;  
    int top;  
    int data[];  
    public Stack(int size){  
        this.size = size;  
        data = new int[size];  
        top = -1;  
    }  
    public boolean IsEmpty() {  
        if(top == -1)  
        {  
            return true;  
        }  
        else  
        {  
            return false;  
        }  
    }  
    public boolean IsFull() {  
        if(top == size - 1)  
        {  
            return true;  
        }  
        else  
        {  
            return false;  
        }  
    }  
}
```

```

public void push(int dt){
    if(!IsFull())
    {
        top++;
        data[top] = dt;
    } else
    {
        System.out.println("Isi Stack penuh");
    }
}

public void pop(){
    if(!IsEmpty())
    {
        int x = data[top];
        top--;
        System.out.println("Data yang keluar: " + x);
    }
    else
    {
        System.out.println("Stack Masih Kosong");
    }
}

public void peek(){
    System.out.println("elemen Teratas: " + data[top]);
}

public void print(){
    System.out.println("Isi Stack : ");
    for (int i = top; i >= 0; i--)
    {
        System.out.println(data[i] + " ");
    }
    System.out.println("");
}

public void clear(){

```

```

        if(IsEmpty())
        {
            for (int i = top; i >= 0; i--)
            {
                top--;
            }
            System.out.println("Stack Sudah dikosongkan");
        }else
        {
            System.out.println("Gagal! Stack Masih Kosong");
        }
    }
}

public class StackMain {
    public static void main(String[] args) {
        Stack stc = new Stack(5);
        stc.push(15);
        stc.push(27);
        stc.push(13);
        stc.print();
        stc.push(11);
        stc.push(34);
        //    stc.push(90);
        stc.pop();
        stc.peek();
        stc.print();
    }
}

```

### 7.3.1 Langkah-langkah Percobaan

```

public class Buku {
    String judul, namaPengarang;
    int tahunTerbit, jmlHalaman, harga;
    Buku(String jd , String nm, int thn, int hal, int hg){
        judul = jd;
    }
}

```

```

        namaPengarang = nm;
        tahunTerbit = thn;
        jmlHalaman = hal;
        harga = hg;
    }
    int size;
    int top;
    Buku data[];
    public Buku(int size){
        this.size = size;
        data = new Buku[size];
        top = -1;
    }

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

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

```

```

    }

    public void push(Buku bk){
        if(!IsFull())
        {
            top++;
            data[top] = bk;
        } else
        {
            System.out.println("Isi Stack penuh");
        }
    }

    public void pop(){
        if(!IsEmpty())
        {
            Buku x = data[top];
            top--;
            System.out.println("Data yang keluar: " +
x.judul + " " + x.namaPengarang + " "
            + x.tahunTerbit + " " + x.jmlHalaman + "
" + x.harga);
        }
        else
        {
            System.out.println("Stack Masih Kosong");
        }
    }

    public void peek(){
        System.out.println("elemen Teratas: " +
data[top].judul + " " + data[top].namaPengarang + " "
            + data[top].tahunTerbit + " " +
data[top].jmlHalaman + " " + data[top].harga);
    }

```

```

    }

    public void print(){
        System.out.println("Isi Stack : ");
        for (int i = top; i >= 0; i--)
        {
            System.out.println(data[i].judul + " " +
data[i].namaPengarang + " "
            + data[i].tahunTerbit + " " +
data[i].jmlHalaman + " " + data[i].harga);
        }
        System.out.println("");
    }
}
import java.util.Scanner;

/**
 *
 * @author ACHE
 */
public class StackMain {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Buku st = new Buku(8);
        char pilih;
        do
        {
            System.out.println("Judul: ");
            String judul = sc.nextLine();
            System.out.println("Nama Pengarang: ");
            String nama = sc.nextLine();
            System.out.println("Tahun Terbit: ");
            int tahun = sc.nextInt();
            System.out.println("Jumlah Halaman: ");

```

```

        int jml = sc.nextInt();
        System.out.println("Harga: ");
        int harga = sc.nextInt();

        Buku bk = new Buku (judul, nama, tahun, jml,
harga);

        System.out.println("Apakah Anda Akan menambahkan
data baru ke Stack (y/n)? ");
        pilih = sc.next().charAt(0);
        sc.nextLine();
        st.push(bk);
    } while (pilih == 'y');
    st.print();
    st.pop();
    st.peek();
    st.print();
}

}

```

#### **7.4.1 Langkah-langkah Percobaan**

```

public class Postfix {
    int n;
    int top;
    char stack[];

    public Postfix(int total){
        n = total;
        top = -1;
        stack = new char[n];
        push('(');
    }

    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;
        }
    }

```

```

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;
    }
}

```

```

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;
}

package Praktikum3;

import java.util.Scanner;

public class PostfixMain {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String P, Q;
        System.out.println("Masukkan Ekspresi Matematika
(infix): ");
        Q = sc.nextLine();
        Q = Q.trim();
        Q = Q + ")";
        int total = Q.length();
    }
}

```

```

        Postfix post = new Postfix(total);

        P = post.Konversi(Q);

        System.out.println("Postfix: " + P);

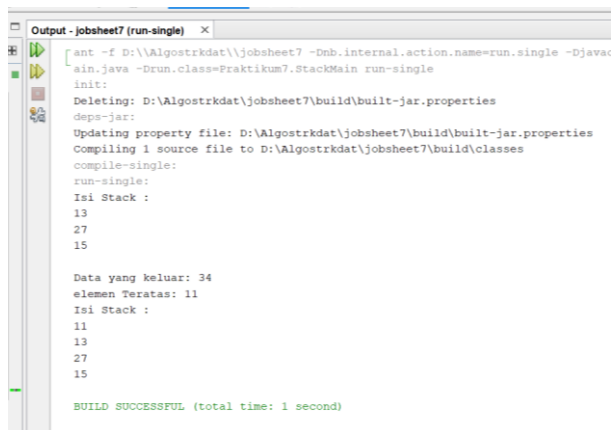
    }

}

```

## B. OUTPUT PROGRAM

### Percobaan 7.2.1



```

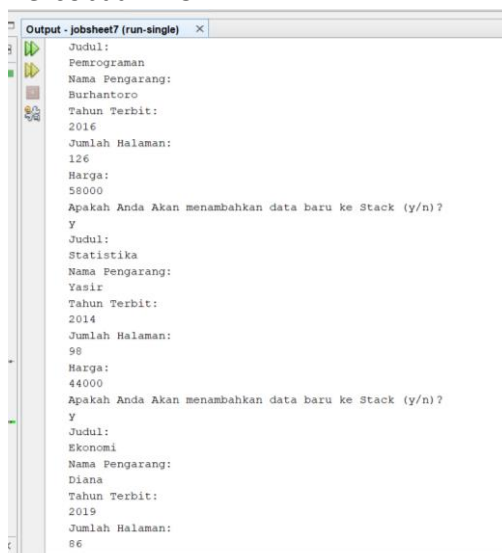
Output - jobsheet7 (run-single) x
[ant -f D:\Algostrkdat\jobsheet7 -Dmb.internal.action.name=run.single -Djava
ain.java -Drun.class=Praktikum7.StackMain run-single
init:
Deleting: D:\Algostrkdat\jobsheet7\build\build-jar.properties
deps-jar:
Updating property file: D:\Algostrkdat\jobsheet7\build\build-jar.properties
Compiling 1 source file to D:\Algostrkdat\jobsheet7\build\classes
compile-single:
run-single:
Isi Stack :
13
27
15

Data yang keluar: 34
elemen Teratas: 11
Isi Stack :
11
13
27
15

BUILD SUCCESSFUL (total time: 1 second)

```

### Percobaan 7.3.1



```

Output - jobsheet7 (run-single) x
Judul:
Pemrograman
Nama Pengarang:
Burhantoro
Tahun Terbit:
2016
Jumlah Halaman:
126
Harga:
58000
Apakah Anda Akan menambahkan data baru ke Stack (y/n)?
y
Judul:
Statistika
Nama Pengarang:
Yasir
Tahun Terbit:
2014
Jumlah Halaman:
98
Harga:
44000
Apakah Anda Akan menambahkan data baru ke Stack (y/n)?
y
Judul:
Ekonomi
Nama Pengarang:
Diana
Tahun Terbit:
2019
Jumlah Halaman:
86
86

```

```

Harga:
47500
Apakah Anda Akan menambahkan data baru ke Stack (y/n)?
n
Isi Stack :
Ekonomi Diana 2019 86 47500
Statistika Yasir 2014 98 44000
Pemrograman Burhantoro 2016 126 58000

Data yang keluar: Ekonomi Diana 2019 86 47500
elemen Teratas: Statistika Yasir 2014 98 44000
Isi Stack :
Statistika Yasir 2014 98 44000
Pemrograman Burhantoro 2016 126 58000

BUILD SUCCESSFUL (total time: 1 minute 12 seconds)

```

### Percobaan 7.4.1

```

run-ranglax:
Masukkan Ekspresi Matematika (infix):
a+b*(c+d-e)/f
Postfix: abcd+e-*f/+
BUILD SUCCESSFUL (total time: 49 seconds)

```

## C. PENJELASAN

### 7.2.3 Pertanyaan

1. Perhatikan class StackMain, apakah fungsi angka 5 pada potongan kode program berikut?

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

Jawab :

2. Lakukan penambahan data ke stack sebanyak dua kali, menggunakan angka 18 dan 40.

Tampilkan hasilnya!

Jawab :

Script :

```

public class StackMain {
    public static void main(String[] args) {
        Stack stc = new Stack(5);
        stc.push(15);
        stc.push(27);
        stc.push(13);
        stc.print();
        stc.push(11);
        stc.push(34);
        stc.pop();
        stc.peek();
    }
}

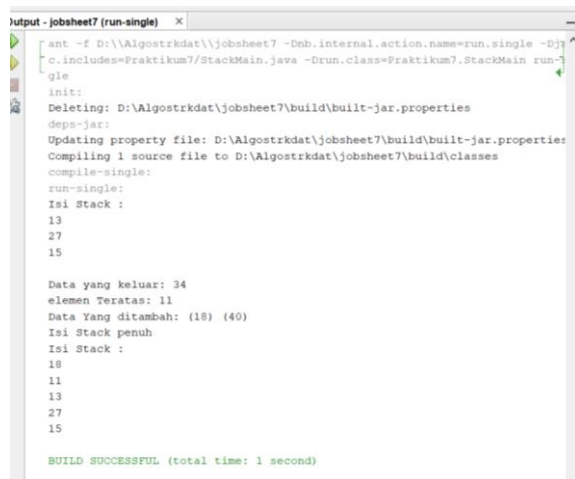
```

```

        int a = 18;
        int b = 40;
        System.out.println("Data Yang ditambah: (" + a+"")
        ("+b+"") ");
        stc.push(a);
        stc.push(b);
        stc.print();
    }
}

```

Running :



```

Output - jobsheet7 (run-single) x
[ant -f D:\Algostrkd\jobsheet7 -Dnb.internal.action.name=run.single -Dj
-c.includes=Praktikum7\StackMain.java -Druntime.class=Praktikum7.StackMain run-
gle
init:
Deleting: D:\Algostrkd\jobsheet7\build\build-jar.properties
deps-jar:
Updating property file: D:\Algostrkd\jobsheet7\build\build-jar.properties
Compiling 1 source file to D:\Algostrkd\jobsheet7\build\classes
compile-single:
run-single:
Isi Stack :
13
27
15

Data yang keluar: 34
elemen Teratas: 11
Data Yang ditambah: (18) (40)
Isi Stack penuh
Isi Stack :
18
11
13
27
15

BUILD SUCCESSFUL (total time: 1 second)

```

3. Pada soal nomor 2, mengapa data yang dimasukkan ke dalam Stack hanya angka 18, sedangkan angka 40 tidak dimasukkan? Jelaskan!

Jawab : karena ukuran Panjang size yang diisi didalam parameter Stack hanya terisi Panjang 5, dan tadi sebelum ditambahkan terdapat 3 panjang dan dimasukkan 2 lagi. dan setelah itu dikeluarkan 1, setelah itu ditambahkan angka 18 dan 40 kenapa yang masuk ke dalam stack hanya angka 18, sedangkan angka 40 tidak dimasukkan, karena Stack Menganut LIFO jadi yang terakhir dimasukkan makan akan pertama kali dikeluarkan, maka data akan masuk kedalam class Stack dengan method Boolean `IsFull()`; dan menampilkan "isi stack penuh"

### 7.3.3 Pertanyaan

1. Perhatikan class StackMain, pada saat memanggil fungsi push, parameter yang dikirimkan adalah bk. Data apa yang tersimpan pada variabel bk tersebut?

```
st.push(bk);
```

Jawab :

Data yang tersimpan yaitu data judul, namaPengarang, TahunTerbit, jmlHalaman, dan harga.

2. Tunjukkan potongan kode program untuk menentukan kapasitas penampungan stack!

Jawab :

```
Buku st = new Buku(8);
```

3. Apakah fungsi penggunaan do-while yang terdapat pada class StackMain?

Jawab : penggunaan do-while untuk mengulang inputan judul, namaPengarang, tahunTerbit, jmlHalaman, dan harga dan jika user mengendaki untuk mengulang inputan judul, namaPengarang, tahunTerbit, jmlHalaman, dan harga.

4. Modifikasi kode program pada class StackMain sehingga pengguna dapat memilih operasi-operasi pada stack (push, pop, peek, atau print) melalui pilihan menu program

Jawab :

Script :

```
public class Stack2Main {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        Scanner input = new Scanner(System.in);  
        Buku st = new Buku(8);  
        char pilih;  
        int ulang = 0;  
        do{  
            System.out.println("");  
            System.out.println(" ~~~ Program Buku~~~");  
            System.out.println("_____");  
            System.out.println("| Pilih menu dibawah ini! |");  
            System.out.println("| 1. Push          |");  
            System.out.println("| 2. Pop           |");  
            System.out.println("| 3. Peek          |");
```

```

System.out.println("| 4. Print          |");
System.out.println("| 5. Keluar          |");
System.out.print("| Pilihah menu : ");
int pilih1 = sc.nextInt();
if(pilih1 == 1){
do
{
System.out.println("Judul: ");
String judul = input.nextLine();
System.out.println("Nama Pengarang: ");
String nama = input.nextLine();
System.out.println("Tahun Terbit: ");
int tahun = sc.nextInt();
System.out.println("Jumlah Halaman: ");
int jml = sc.nextInt();
System.out.println("Harga: ");
int harga = sc.nextInt();

Buku bk = new Buku (judul, nama, tahun, jml, harga);
System.out.println("Apakah Anda Akan menambahkan data baru ke Stack (y/n)? ");
pilih = sc.next().charAt(0);
sc.nextLine();
st.push(bk);
} while (pilih == 'y');
    continue;
}else if(pilih1 == 2){
    st.pop();
    continue;
} else if (pilih1 == 3){
    st.peek();
} else if (pilih1 == 4){
    st.print();

```



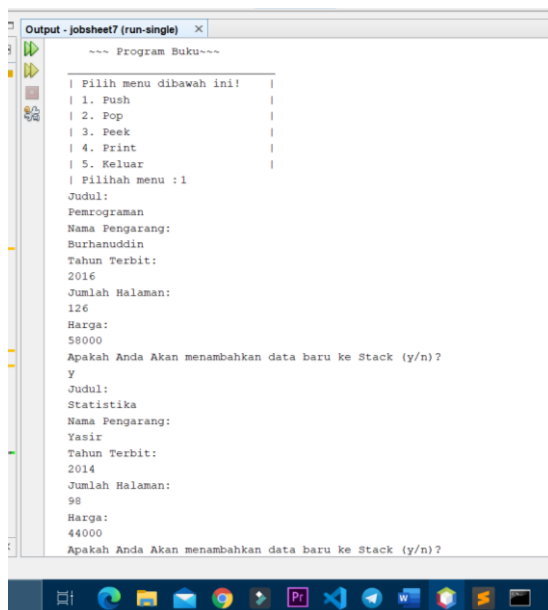
```

    }else if(pilih1 == 5){
        System.out.println("Anda telah keluar dari program");
        System.exit(0);
        break;
    }else{
        System.out.println("Menu yang anda berikan tidak tersedia");
        System.out.println("");
    }
}while(ulang !=0 || ulang<3);
}

}

```

Running :



```

Output - jobsheet7 (run-single) x
~~~ Program Buku~~~
| Pilih menu dibawah ini! |
| 1. Push                 |
| 2. Pop                  |
| 3. Peek                 |
| 4. Print                |
| 5. Keluar               |
| Pilihah menu : 1       |
Judul:
Femrograman
Nama Pengarang:
Burhanuddin
Tahun Terbit:
2016
Jumlah Halaman:
126
Harga:
58000
Apakah Anda Akan menambahkan data baru ke Stack (y/n)?
y
Judul:
Statistika
Nama Pengarang:
Yasir
Tahun Terbit:
2014
Jumlah Halaman:
98
Harga:
44000
Apakah Anda Akan menambahkan data baru ke Stack (y/n)?

```

```
Output - jobsheet7 (run-single) x
Apakah Anda Akan menambahkan data baru ke Stack (y/n)?
y
Judul:
Ekonomi
Nama Pengarang:
Diana
Tahun Terbit:
2019
Jumlah Halaman:
86
Harga:
47500
Apakah Anda Akan menambahkan data baru ke Stack (y/n)?
n

~~~ Program Buku~~~

| Pilih menu dibawah ini! |
| 1. Push                 |
| 2. Pop                  |
| 3. Peek                 |
| 4. Print                |
| 5. Keluar               |
| Pilihah menu : 4        |
Isi Stack :
Ekonomi Diana 2019 86 47500
Statistika Yasir 2014 98 44000
Pemrograman Burhanuddin 2016 126 58000

~~~ Program Buku~~~
```

```
Output - jobsheet7 (run-single) x

~~~ Program Buku~~~

| Pilih menu dibawah ini! |
| 1. Push                 |
| 2. Pop                  |
| 3. Peek                 |
| 4. Print                |
| 5. Keluar               |
| Pilihah menu : 2        |
Data yang keluar: Ekonomi Diana 2019 86 47500

~~~ Program Buku~~~

| Pilih menu dibawah ini! |
| 1. Push                 |
| 2. Pop                  |
| 3. Peek                 |
| 4. Print                |
| 5. Keluar               |
| Pilihah menu : 3        |
elemen Teratas: Statistika Yasir 2014 98 44000

~~~ Program Buku~~~

| Pilih menu dibawah ini! |
| 1. Push                 |
| 2. Pop                  |
| 3. Peek                 |
| 4. Print                |
| 5. Keluar               |
| Pilihah menu : 4        |
```

```
Output - jobsheet7 (run-single) x
| 3. Peek                 |
| 4. Print                |
| 5. Keluar               |
| Pilihah menu : 3        |
elemen Teratas: Statistika Yasir 2014 98 44000

~~~ Program Buku~~~

| Pilih menu dibawah ini! |
| 1. Push                 |
| 2. Pop                  |
| 3. Peek                 |
| 4. Print                |
| 5. Keluar               |
| Pilihah menu : 4        |
Isi Stack :
Statistika Yasir 2014 98 44000
Pemrograman Burhanuddin 2016 126 58000

~~~ Program Buku~~~

| Pilih menu dibawah ini! |
| 1. Push                 |
| 2. Pop                  |
| 3. Peek                 |
| 4. Print                |
| 5. Keluar               |
| Pilihah menu : 5        |
Anda telah keluar dari program
BUILD SUCCESSFUL (total time: 2 minutes 33 seconds)
```

### 7.4.3 Pertanyaan

1. Perhatikan class Postfix, jelaskan alur kerja method derajat!

Jawab : pada method derajat yang ada di class Postfix yaitu untuk mengelompokkan operator, dan operator yang paling Tinggi yaitu operator pangkat(^), lalu operator yang setara yaitu operator Modulo(%) ,Pembagian (/) , dan Perkalian(\*), lalu operator yang paling rendah dan setara yaitu Pengurangan(-) dan Penjumlahan(+). Dan pada operator tersebut akan dikelompokkan menggunakan char c dengan pemilihan switch case.

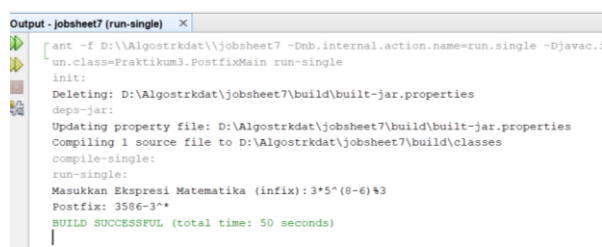
2. Apa fungsi kode program berikut?

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

Jawab : kode script diatas ditujukan untuk membentuk suatu bilangan huruf per karakter dan akan digabung sehingga akan membentuk suatu bentuk String. Dan akan dimasukkan kedalam variable c.

3. Jalankan kembali program tersebut, masukkan ekspresi  $3*5^{(8-6)}\%3$ . Tampilkan hasilnya!

Jawab :



```
Output - jobsheet7 (run-single) x
[ ant -f D:\Algostrkdat\jobsheet7 -Dnb.internal.action.name=run.single -DjavaC...
  un.class=Praktikum3.PostfixMain run-single
  init:
  Deleting: D:\Algostrkdat\jobsheet7\build\build-jar.properties
  deps-jar:
  Updating property file: D:\Algostrkdat\jobsheet7\build\build-jar.properties
  Compiling 1 source file to D:\Algostrkdat\jobsheet7\build\classes
  compile-single:
  run-single:
  Masukkan Ekspresi Matematika (infix): 3*5^(8-6)%3
  Postfix: 3586-3**
  BUILD SUCCESSFUL (total time: 50 seconds)
]
```

4. Pada soal nomor 2, mengapa tanda kurung tidak ditampilkan pada hasil konversi? Jelaskan!

Jawab : program tidak akan tampil jika tanda kurung dimasukkan kedalam sebuah method derajat, karena itu tanda kurung dialihkan kedalam push ke stack dan saat itu akan muncul karakter.

## D. KESIMPULAN

### Percobaan 7.2.1

Dari percobaan praktikum1 diatas bahwa stack membutuhkan Panjang dari array untuk menentukan Panjang elemen, dan elemen yang pertama kali saat dimasukkan akan disimpan kedalam array Stack paling bawah dan sebaliknya saat elemen yang terakhir dimasukkan akan ditaruh atau disimpan kedalam array stack paling atas.

Ada beberapa method operasi yang ada pada script class Stack seperti :

1. **IsEmpty()** , yang digunakan untuk melakukan pengosongan atau melakukan pengurangan pada array stack dan dimulai dari array stack paling atas dan jika ada pengurangan maka data top akan dikurangi -1 dan jika salah tidak akan melakukan apa – apa.
2. **IsFull()**, yang digunakan apakah data yang dimasukkan melebihi dari batas array Stack yang telah ditentukan. Jika benar ada data yang melebihi dari data yang telah ditentukan maka akan di return true if(top == size-1).
3. **Push(int dt)**, method push dengan parameter int dt digunakan untuk melanjutkan dari method IsFul() dan melakukan pengecekan jika data tidak penuh maka data akan bisa ditambah. Dan jika salah maka akan muncul tulisan “Isi stack penuh”.
4. **Method pop()**, digunakan untuk pengecekan data keluar. Jika data IsEmpty tidak keluar maka data akan dikurangi dari elemen paling akhir.
5. **Peek()**, digunakan untuk mengetahui elemen paling atas.
6. **Clear()**, digunakan untuk menghapus atau mengkosongkan semua data pada array stack yang ada isinya.

### Percobaan 7.3.1

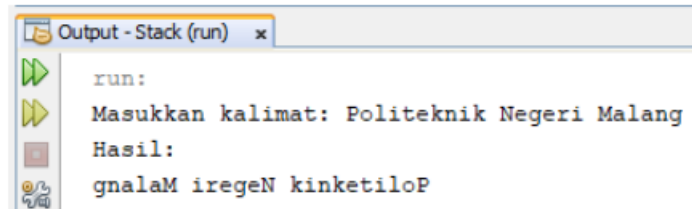
Pada Script diatas dapat disimpulkan bahwa tipe data yang dipakai pada scipt diatas menggunakan data primitive, dan dapat menyimpan tipe data seperti objek atau String. Method push dengan parameter int dt digunakan untuk melanjutkan dari method IsFul() dan melakukan pengecekan jika data tidak penuh maka data akan bisa ditambah. Dan jika salah maka akan muncul tulisan “Isi stack penuh”.

### Percobaan 7.4.1

Penerapan operator yang ada di Postfix dan akan dipindahkan kedalam stack dan pada bidang aritmatika adalah penulisan ekspresi matematika dan Biasanya, ekspresi matematika ditulis menggunakan notasi infix, dan Setelah persamaan infix terbaca, pindahkan semua isi stack (yang tersisa) ke postfix. namun notasi postfix adalah notasi yang digunakan oleh mesin kompilasi komputer untuk mempermudah proses pengodean, contoh penerapannya pada kalkulator Hp.

## “TUGAS”

1. Buat program dengan menggunakan konsep Stack untuk memasukkan sebuah kalimat, kemudian keluaran yang ditampilkan berupa kalimat dengan urutan karakter terbalik!



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. Informasi yang tersimpan pada struk belanja terdiri dari:

- Nomor transaksi
- Tanggal pembelian
- Jumlah barang yang dibeli
- Total harga bayar

## ~~JAWAB~~

### 1. Tugas1

#### A. Kode Program

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

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

    public boolean IsFull() {
        if(top == size - 1)
```

```

        {
            return true;
        }
        else
        {
            return false;
        }
    }
    public void push(char dt){
        if(!IsFull())
        {
            top++;
            data[top] = dt;
        } else
        {
            System.out.println("Isi Stack penuh");
        }
    }
    public void pop(){
        if(!IsEmpty())
        {
            char x = data[top];
            top--;
            System.out.println("Data yang keluar: " +
x);
        }
        else
        {
            System.out.println("Stack Masih Kosong");
        }
    }
    public void peek(){
        System.out.println("elemen Teratas: " +
data[top]);
    }
    public void print(){
        System.out.println("Isi Stack : ");
        for (int i = top; i >= 0; i--)
        {
            System.out.print(data[i] + "");
        }
        System.out.println("");
    }
    public void clear(){
        if(IsEmpty())
        {
            for (int i = top; i >= 0; i--)
            {
                top--;
            }
        }
    }

```

```

        System.out.println("Stack Sudah
dikosongkan");
    }else
    {
        System.out.println("Gagal! Stack Masih
Kosong");
    }
}
}
package Tugas;

import java.util.Scanner;

public class ReverseMain {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Reverse balik;
        char C;
        System.out.print("Masukkan Kalimat: ");
        String kalimat = sc.nextLine();

        int total = kalimat.length();

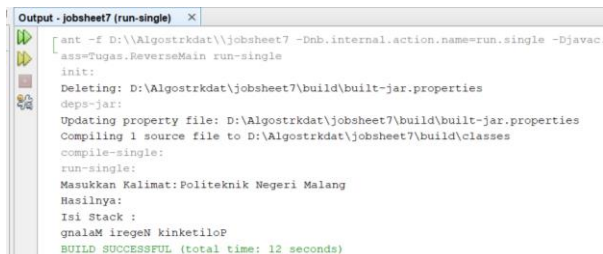
        balik = new Reverse(total);

        for (int i = 0; i < total; i++) {
            C = kalimat.charAt(i);
            balik.push(C);
        }

        System.out.println("Hasilnya: ");
        balik.print();
    }
}

```

## B. Output Program Running :



```

Output - jobsheet7 (run-single) x
[ ant -f D:\Algostrkdat\jobsheet7 -Dnb.internal.action.name=run.single -Djavac.
  ass=Tugas.ReverseMain run-single
  init:
  Deleting: D:\Algostrkdat\jobsheet7\build\build-jar.properties
  deps-jar:
  Updating property file: D:\Algostrkdat\jobsheet7\build\build-jar.properties
  Compiling 1 source file to D:\Algostrkdat\jobsheet7\build\classes
  compile-single:
  run-single:
  Masukkan Kalimat: Politeknik Negeri Malang
  Hasilnya:
  Isi Stack :
  gnalaM iregeN kinketiloP
  BUILD SUCCESSFUL (total time: 12 seconds)

```

## C. Penjelasan

Dari hasil outputan, terdapat 2 class yaitu Class Stack dan class StackMain. Dan didalam class Stack dimasukkan atribut dengan tipe data char dan int. di tipe

data char terdapat array 1d yang digunakan untuk menentukan dan menyimpan Panjang yang akan dimasukkan data menggunakan inputan user scanner. Dan integer digunakan untuk menentukan size dan angka paling atas(top).

Lalu terdapat juga method IsFull(), method push(), dan method print(). Didalam method IsFull menggunakan Boolean yang terdapat 2 pemilihan yaitu true dan false. Jika `if(top == size - 1)` akan di return true . Dan jika else maka akan di return false.

Dan method push digunakan untuk melakukan pengecekan jika data tidak penuh maka data akan bisa ditambah. Dan jika salah maka akan muncul tulisan “Isi stack penuh”.

Dan yang terakhir adalah method print() yang digunakan untuk menampilkan isi yang ada didalam stack.

Dan masuk ke class StackMain yang digunakan untuk menginput data menggunakan inputan dan inputan tersebut di cari lengthnya dengan menggunakan variable total dengan tipe data integer dan kemudian melakukan perulangan for untuk push dari String ke char dan hasil nya dikeluarkan.

#### **D. Kesimpulan**

Kesimpulan yang saya dapat yaitu pada kasus ini bagaimana cara membalikan kalimat menggunakan String sebelum saya tau caranya menggunakan String saya menggunakan Char tapi menurut saya menggunakan Char kurang efektif karena harus mendeklarasikan satu-satu.

## **2. Tugas no 2**

### **A. Kode Program**

```
public class Belanja {
    String tanggal, barang, nomorTransaksi;
    int totalHarga, jumlahBarang;
    Belanja(String tgl, String b, String nt, int th,
int jml){
        tanggal = tgl;
        barang = b;
        nomorTransaksi = nt;
        totalHarga = th;
        jumlahBarang = jml;
    }
}

public class BelanjaServices1 {
    int size;
    int top;
    Belanja data[];
    BelanjaServices1(int size){
        this.size = size;
        data = new Belanja[size];
        top = -1;
    }

    public boolean IsEmpty() {
        if(top == -1)
        {
            return true;
        }
    }
}
```



```

        }
        else
        {
            return false;
        }
    }

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

    public void push(Belanja bk){
        if(!IsFull())
        {
            top++;
            data[top] = bk;
        } else
        {
            System.out.println("Isi Stack penuh");
        }
    }

    public void pop(){
        if(!IsEmpty())
        {
            Belanja x = data[top];
            top--;
            System.out.println("id struk      nama
barang      jumlah Barang      Total harga
Tanggal");
            System.out.println(x.nomorTransaksi+"
"+x.barang+"      \t\t"+x.jumlahBarang+"
\t"+x.totalHarga+"      "+x.tanggal);
        }
        else
        {
            System.out.println("Stack Masih Kosong");
        }
    }

    public void peek(){
        System.out.println("id struk      nama barang
jumlah Barang      Total harga      Tanggal");
    }

```

```

        System.out.println(data[top].nomorTransaksi+"
"+data[top].barang+"
\t\t"+data[top].jumlahBarang+"
\t"+data[top].totalHarga+"
"+data[top].tanggal);
    }

    public void print() {
        System.out.println("Isi stack:");
        for (int i = top; i >= 0; i--) {
            System.out.println("id struk      nama
barang      jumlah Barang      Total harga
Tanggal");
            System.out.println(data[i].nomorTransaksi+"
"+data[i].barang+"      \t\t"+data[i].jumlahBarang+"
\t"+data[i].totalHarga+"      "+data[i].tanggal);
        }
        System.out.println("-----
-----");
    }

    void namaBubbleSortAZ() {
        BelanjaServices1 bl;
        for (int i=0; i<data.length; i++){
            for (int j=1; j<data.length-i; j++){
                if (data[j].barang.compareTo(data[j-
1].barang)<0) {
                    Belanja tmp = data[j];
                    data[j] = data[j-1];
                    data[j-1] = tmp;
                }
            }
        }
    }

    void tampil(){
        BelanjaServices1 bl;
        System.out.println("no Rekening      Nama
NamaIbu      Phone      Email");
        for (int i = 0; i < data.length; i++) {
            System.out.println("id struk      tanggal
nama barang      total      harga");
            System.out.println(data[i].nomorTransaksi+"
"+data[i].barang+"      "+data[i].jumlahBarang+"
"+data[i].totalHarga+"      "+data[i].tanggal);
        }
    }
}

```

## B. Output Program

```
Output - jobsheet7 (run-single) X
Deleting: D:\Algostrkdat\jobsheet7\build\build-jar.properties
deps-jar:
Updating property file: D:\Algostrkdat\jobsheet7\build\build-jar.properties
Compiling 1 source file to D:\Algostrkdat\jobsheet7\build\classes
compile-single:
run-single:

    ~~~ Program Belanja~~~

| Pilih menu dibawah ini! |
| 1. Tampil Struk         |
| 2. Ambil Struk          |
| 3. Ambil Struk Paling Atas |
| 0. Keluar               |
| Pilihah menu : 1        |
Isi stack:
id struk  nama barang      jumlah Barang      Total harga      Tanggal
TR98      Lemari           1                  2500000          02-23-2021
id struk  nama barang      jumlah Barang      Total harga      Tanggal
TR90      Kompor           1                  1100000          02-16-2021
id struk  nama barang      jumlah Barang      Total harga      Tanggal
TR83      shampo           9                  1000000          02-09-2021
id struk  nama barang      jumlah Barang      Total harga      Tanggal
TR76      laptop            1                  8000000          02-02-2021
id struk  nama barang      jumlah Barang      Total harga      Tanggal
TR62      kamera            1                  5000000          01-22-2021
id struk  nama barang      jumlah Barang      Total harga      Tanggal
TR55      Gorden            1                  2000000          01-15-2021
id struk  nama barang      jumlah Barang      Total harga      Tanggal
TR49      Sepatu            1                  1200000          01-08-2021
-----
```

```
~~~~~ Program Belanja~~~~~

| Pilih menu dibawah ini! |
| 1. Tampil Struk         |
| 2. Ambil Struk          |
| 3. Ambil Struk Paling Atas |
| 0. Keluar               |
| Pilihah menu : 3        |
id struk  nama barang      jumlah Barang      Total harga      Tanggal
TR98      Lemari           1                  2500000          02-23-2021
~~~~~ Program Belanja~~~~~

| Pilih menu dibawah ini! |
| 1. Tampil Struk         |
| 2. Ambil Struk          |
| 3. Ambil Struk Paling Atas |
| 0. Keluar               |
| Pilihah menu : 1        |
Isi stack:
id struk  nama barang      jumlah Barang      Total harga      Tanggal
TR98      Lemari           1                  2500000          02-23-2021
id struk  nama barang      jumlah Barang      Total harga      Tanggal
TR90      Kompor           1                  1100000          02-16-2021
id struk  nama barang      jumlah Barang      Total harga      Tanggal
TR83      shampo           9                  1000000          02-09-2021
id struk  nama barang      jumlah Barang      Total harga      Tanggal
TR76      laptop            1                  8000000          02-02-2021
id struk  nama barang      jumlah Barang      Total harga      Tanggal
TR62      kamera            1                  5000000          01-22-2021
id struk  nama barang      jumlah Barang      Total harga      Tanggal
TR55      Gorden            1                  2000000          01-15-2021
id struk  nama barang      jumlah Barang      Total harga      Tanggal
TR49      Sepatu            1                  1200000          01-08-2021
~~~~~ Program Belanja~~~~~

| Pilih menu dibawah ini! |
| 1. Tampil Struk         |
| 2. Ambil Struk          |
| 3. Ambil Struk Paling Atas |
```

```

| Pilih menu dibawah ini! |
| 1. Tampil Struk |
| 2. Ambil Struk |
| 3. Ambil Struk Paling Atas |
| 0. Keluar |
| Pilihah menu : 2 |
id struk nama barang jumlah Barang Total harga Tanggal
TR98 Lemari 1 2500000 02-23-2021
id struk nama barang jumlah Barang Total harga Tanggal
TR90 Kompor 1 1100000 02-16-2021
id struk nama barang jumlah Barang Total harga Tanggal
TR83 shampo 9 1000000 02-09-2021
id struk nama barang jumlah Barang Total harga Tanggal
TR76 laptop 1 8000000 02-02-2021
id struk nama barang jumlah Barang Total harga Tanggal
TR62 kamera 1 5000000 01-22-2021

~~~ Program Belanja~~~

| Pilih menu dibawah ini! |
| 1. Tampil Struk |
| 2. Ambil Struk |
| 3. Ambil Struk Paling Atas |
| 0. Keluar |
| Pilihah menu : 1 |
Isi stack:
id struk nama barang jumlah Barang Total harga Tanggal
TR55 Gorden 1 2000000 01-15-2021
id struk nama barang jumlah Barang Total harga Tanggal
TR49 Sepatu 1 1200000 01-08-2021

~~~ Program Belanja~~~

| Pilih menu dibawah ini! |
| 1. Tampil Struk |
| 2. Ambil Struk |
| 3. Ambil Struk Paling Atas |
| 0. Keluar |
| Pilihah menu : 3 |
id struk nama barang jumlah Barang Total harga Tanggal
TR55 Gorden 1 2000000 01-15-2021

~~~ Program Belanja~~~

| Pilih menu dibawah ini! |
| 1. Tampil Struk |
| 2. Ambil Struk |
| 3. Ambil Struk Paling Atas |
| 0. Keluar |
| Pilihah menu : 0 |
Anda telah keluar dari program
BUILD SUCCESSFUL (total time: 24 seconds)

```

### C. Penjelasan

Didalam script diatas, menggunakan 3 class yang berbeda yaitu class Belanja, class BelanjaServices1, dan class Main.

Lalu di class BelanjaServices1 terdapat beberapa atribut seperti int size, int top, dan StrukBelanjaan data[]. Dan ada beberapa method juga didalam nya seperti method IsFull() yang digunakan untuk apakah data yang dimasukkan melebihi dari batas array Stack yang telah ditentukan. Lalu method push dengan parameter BelanjaServices1 bl yang digunakan untuk melakukan pengecekan di class BelanjaServices1 yang akan diisi apakah nilai ada ruang untuk array, jika ada maka akan ditambahkan kedalam array stack dan jika sudah penuh maka ada tulisan data sudah penuh. Lalu lanjut kedalam method pop yang digunakan untuk melakukan pengambilan pada data array stack yang sudah di inputkan kedalam dan pada kasus ini saya memanggil sebanyak 5 kali. Lalu lanjut method peek() yang digunakan untuk menampilkan data yang paling atas dari data array Stack. Lalu method print() digunakan untuk menampilkan data pada array stack seluruhnya.

Lanjut di class Main, sudah saya deklarasikan datanya jadi tinggal panggil .

### D. Kesimpulan

pada kasus ini bagaimana caranya menyimpan stack dan mengeluarkan stack sebanyak 5 kali tetapi menurut saya acara mengambil Stack yang saya buat ini kurang efektif karena harus mendeklarasikan satu-satu, mungkin kalau ada waktu saya akan mencoba mengambil Stack yang menurut saya lebih efektif