

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA**



**Oleh:
DZULFIKAR MUHAMMAD AL GHIFARI
NIM. 2341760071
SIB-1F / 08
D-IV SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG**

PRAKTIKUM 7

7.2 Percobaan 1

6.2.1 Langkah Langkah percobaan 1

1. Membuat class sorting

```
CodeEditor | Editor | Explain  
public class Stack08 {  
}
```

2. Deklarasi variable dan constructor

```
public class Stack08 {  
    int data[];  
    int size;  
    int top;  
  
    public Stack08(int size) {  
        this.size = size;  
        data = new int[size];  
        top = -1;  
    }  
}
```

3. 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;  
    }  
}
```

4. 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:"Stock masih kosong");  
    }  
}
```

5. Menambahkan method peek

```
CodeEditor | Editor | Explain | Generate README | ...  
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:"");  
}
```

6. Menambahkan 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] + " ");  
    }  
    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");  
    }  
}
```

7. Membuat file StackDemo dan membuat object stack

```
import java.util.Stack;

public class StackDemo08 {
    public static void main(String[] args) {
        Stack08 stack = new Stack08(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();
    }
}
```

VERIFIKASI HASIL PERCOBAAN 7.2.2

```
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
```

PERTANYAAN 3.2.3

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 kondisi kosong, jika stack berisi kosong maka tidak akan bisa mengambil value dan akan mengembalikan pesan stack masih kosong
2. peek() untuk memeriksa elemen paling atas, sedangkan pop() digunakan untuk mengambil elemen pada stack paling atas

7.3 Percobaan 2

7.3.1 Langkah-langkah Percobaan 2

1. Menambahkan class pakaian

```
Pakaian08.java > ...
Codeium: Refactor | Explain
1  ✓ public class Pakaian08 {
2
3  }
4
```

2. Menambahkan attribute dan konstruktor

```
Codeium: Refactor | Explain
public class Pakaian08 {
    String jenis, warna, merk, ukuran;
    double harga;

    Pakaian08(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;
    }
}
```

3. Menambahkan class stack dan memberi attribute dan kosntrukturnya

```
public class Stack08 {
    int size;
    int top;
    Pakaian08 data[];

    public Stack08(int size) {
        this.size = size;
        data = new Pakaian08[size];
        top = -1;
    }
}
```

4. Menambahkan method isEmpty

```
Codeium: Refactor | Explain | Generate
public boolean isEmpty() {
    if (top == -1) {
        return true;
    } else {
        return false;
    }
}
```

5. Menambahkan method isFull

```
Codeium: Refactor | Explain | Generate
public boolean IsFull() {
    if (top == size - 1) {
        return true;
    } else {
        return false;
    }
}
```

6. Menambahkan methos push

```
public void push(Pakaian08 pkn) {
    if (!IsFull()) {
        top++;
        data[top] = pkn;
    } else {
        System.out.println("Isi stack penuh!");
    }
}
```

7. Menambahkan method pop

```
Codeium: Refactor | Explain | Generate
public void pop() {
    if (!IsEmpty()) {
        Pakaian08 x = data[top];
        top--;
        System.out.println("Data yang keluar: " + x.jenis + " " + x.warna + " " + x.merk +
            " " + x.ukuran + " " + x.harga);
    } else {
        System.out.println("Stack masih kosong");
    }
}
```

8. Menambahkan method peek

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

9. Membuat method print

```
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:"");  
};
```

10. Membuat method clear

```
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");  
    }  
}
```

11. Membuat class stack main dan melakukan instansiasi stack

```
Codeium: Reactor | Explain  
public class StackMain08 {  
    Stack08 stk = new Stack08(size:5);  
}
```

12. Deklarasi scanner

```
Scanner sc = new Scanner(System.in);
```

13. Menambahkan input untuk pakaian

```
char pilih;  
do {  
    System.out.print(s:"Jenis: ");  
    String jenis = sc.nextLine();  
    System.out.print(s:"Warna: ");  
    String warna = sc.nextLine();  
    System.out.print(s:"Merk: ");  
    String merk = sc.nextLine();  
    System.out.print(s:"Ukuran: ");  
    String ukuran = sc.nextLine();  
    System.out.print(s:"Harga: ");  
    double harga = sc.nextDouble();  
  
    Pakaian08 p = new Pakaian08(jenis, warna, merk, ukuran, harga);  
    System.out.print(s:"Apakah Anda akan menambahkan data baru ke stack (yn) ? ");  
    pilih = sc.next().charAt(index:0);  
    sc.nextLine();  
    stk.push(p);  
} while (pilih == 'y');
```

14. Melakukan pemanggilan method

```
stk.print();  
stk.pop();  
stk.peek();  
stk.print();
```

VERIFIKASI HASIL PERCOBAAN 7.3.2

```

Jenis: Kaos
Warna: Hitam
Merk: Nevada
Ukuran: M
Harga: 85000
Apakah Anda akan menambahkan data baru ke stack (yn) ? y
Jenis: Kemeja
Warna: Putih
Merk: Styves
Ukuran: XL
Harga: 127000
Apakah Anda akan menambahkan data baru ke stack (yn) ? y
Jenis: Celana
Warna: Biru
Merk: Levis
Ukuran: L
Harga: 189500
Apakah Anda akan menambahkan data baru ke stack (yn) ? 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

PS E:\1POLINEMA\2Genap 2023-2024\PraktikumAlgoritma\pertemuan9>

```

PERTANYAAN 7.3.3

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 pada stack

```
Stack08 stk = new Stack08(size:5);
```
2. Data yang terkirim adalah data yang telah di inputkan pada sebelumnya

```
Pakaian08 p = new Pakaian08(jenis, warna, merk, ukuran, harga);
```
3. Untuk mengulang input pakaian sampai var pilih bernilai y
4. Dengan mengubahnya menjadi seperti berikut


```

Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    Stack08 stk = new Stack08(size:5);
    StackMain08 main = new StackMain08();
    // main.pushPakaian(stk);

    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. Keluar");
        System.out.print(s:"Pilih salah satu (1/2/3/4/5) : ");
        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:
                return;
            default:
                System.out.println(x:"Pilihan tidak tersedia. ");
                break;
        }
    }
}

```

```

Codeium: Refactor | Explain | Generate Javadoc | X
public void pushPakaian(Stack08 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();

        Pakaian08 p = new Pakaian08(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);
        // Removed unnecessary newline()
        stk.push(p);
    } while (pilih == 'y');
}

```

```

=====
Pilihan
1. Push
2. Pop
3. Peek
4. Print
5. Keluar
Pilih salah satu (1/2/3/4/5) : 1
Jenis: Kaos
Warna: Merah
Merk: pdi
Ukuran: XL
Harga: 45000
Apakah Anda akan menambahkan data baru ke stack (y/n) ? y
Jenis: Celana
Warna: Hijau
Merk: PKB
Ukuran: L
Harga: 50000
Apakah Anda akan menambahkan data baru ke stack (y/n) ? n
=====

```

```

=====
Pilihan
1. Push
2. Pop
3. Peek
4. Print
5. Keluar
Pilih salah satu (1/2/3/4/5) : 2
Data yang keluar: Celana Hijau PKB L 50000.0
=====
Pilihan
1. Push
2. Pop
3. Peek
4. Print
5. Keluar
Pilih salah satu (1/2/3/4/5) : 3
Elemen teratas: Kaos Merah pdi XL 45000.0
=====
Pilihan
1. Push
2. Pop
3. Peek
4. Print
5. Keluar
Pilih salah satu (1/2/3/4/5) : 5
PS E:\POLINEMA\2Genap 2023-2024\PraktikumAlgoritma\pertemuan9>

```

7.4 Percobaan 3

7.4.2 Langkah-langkah Percobaan 3

1. Membuat class postfix dan deklarasi dan membuat construct

```

Codeium: Refactor | Explain
public class Postfix08 {
    int n, top;
    char[] stack;
}

```

```

public Postfix08(int total) {
    n = total;
    top = -1;
    stack = new char[n];
    push('(');
}

```

2. Menambahkan method push dan pop

```

Codeium: Refactor | Explain | Generate Javadoc | >
public void push(char c) {
    top++;
    stack[top] = c;
}

Codeium: Refactor | Explain | Generate Javadoc | >
public char pop() {
    char item = stack[top];
    top--;
    return item;
}

```

3. Menambahkan method sort

```

private void sort(int data[], int left, int right) {
    if (left < right) {
        int middle = (left + right) / 2;
        sort(data, left, middle);
        sort(data, middle + 1, right);
        merge(data, left, middle, right);
    }
}

```


4. Membuat method isOperand

```
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;  
    }  
}
```

5. Membuat method isOperator

```
public boolean IsOperator(char c) {  
    if (c == '^' || c == '%' || c == '/' || c == '*' || c == '-' || c == '+') {  
        return true;  
    } else {  
        return false;  
    }  
}
```

6. 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;  
    }  
}
```

7. 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;  
}
```

8. Membuat class PostfixMain

```
public class PostfixMain {  
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        String Q;  
        System.out.println(x: "Masukkan ekspresi matematika (infix) : ");  
        Q = sc.nextLine();  
        Q = Q.trim();  
        Q = Q + " ";  
    }  
} Codeium: Generate (Ctrl+I)
```

9. Instansiasi total, post, dan pemanggilan method konversi

```
int total = Q.length();  
  
Postfix08 post = new Postfix08 (total) ;  
P = post. konversi(Q) ;  
System. out.println ("Posftix: " + P);
```

VERIFIKASI HASIL PERCOBAAN 7.4.3

```
Masukkan ekspresi matematika (infix) :  
a+b*(c+d-e)/f  
Posftix: abcd+e-*f/+
```

PERTANYAAN 7.4.3

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 param berupa char. Character akan dilakukan pengecekan menggunakan switch case. Pengecekan dilakukan berurutan hingga case terakhir (tanpa diakhiri di tengah tengah case), jika parameter sama dengan case maka akan mereturn kan angka sesuai case dan selanjutnya melakukan pengecekan pada case di bawahnya.
2. Untuk memanggil karakter urutan ke i pada Q
- 3.

```
Masukkan ekspresi matematika (infix) :  
5*4^(1+2)%3  
Posftix: 5412+^*3%
```

4. Tanda kurung tidak diperlukan pada hasil konversi, karena sudah di atur urutan nya 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 bayar

Tampilkan informasi struk belanja yang masih tersimpan di dalam stack!

Jawaban

1. Dengan menambahkan method dibawah ini

```
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();
}
```

Dan melakukan perubahan pada main class

```
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;
}
```

Kemudian hasilnya akan seperti dibawah ini

```
=====
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: bres
Ukuran: XL
Harga: 5000
Apakah Anda akan menambahkan data baru ke stack (y/n) ? y
Jenis: celana
Warna: hitam
Merk: HUY
Ukuran: M
Harga: 3000
Apakah Anda akan menambahkan data baru ke stack (y/n) ? y
Jenis: celana
Warna: putih
Merk: HYU
Ukuran: S
Harga: 7500
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 putih HUY S 7500.0
```

2. Membuat class struk dengan method sebagai berikut

```
package tugas;

Codeium: Refactor | Explain
public class Struk08 {
    int noTransaksi, jumlahBarang;
    String tglBeli;
    int totalBayar;
    int size;
    int top;
    Struk08 data[];
    Struk08[] stk;

    Struk08(int no, String tgl, int jb, int tb){
        noTransaksi = no;
        tglBeli = tgl;
        jumlahBarang = jb;
        totalBayar = tb;
    }

    public Struk08(int size){
        this.size = size;
        data = new Struk08[size];
        top = -1;
    }

    Codeium: Refactor | Explain | Generate Javadoc | X
    public boolean isEmpty(){
        return top == -1;
    }

    Codeium: Refactor | Explain | Generate Javadoc | X
    public boolean isFull(){
        return top == size - 1;
    }

    Codeium: Refactor | Explain | Generate Javadoc | X
    public void push(Struk08 dt){
        if (!isFull()) {
            top++;
            data[top] = dt;
        } else {
            System.out.println("Isi Stack Penuh!");
        }
    }

    Codeium: Refactor | Explain | Generate Javadoc | X
    public void pop(){
        if (!isEmpty()) {
            Struk08 x = data[top];
            top--;
            System.out.println("Data yang Keluar: " + x.noTransaksi + " "
                + x.tglBeli + " " + x.jumlahBarang + " "
                + x.totalBayar + " ");
        } else {
            System.out.println("Stack masih kosong");
        }
    }

    Codeium: Refactor | Explain | Generate Javadoc | X
    public void peek(){
        System.out.println("Elemen teratas: " + data[top].noTransaksi + " "
            + " " + data[top].tglBeli + " " + data[top].jumlahBarang
            + " " + data[top].totalBayar);
    }

    Codeium: Refactor | Explain | Generate Javadoc | X
    public void print(){
        System.out.println("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("");
    }

    Codeium: Refactor | Explain | Generate Javadoc | X
    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");
        }
    }
}
```

Kemudian class main dengan method seperti berikut ini

```
Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    Struk08 stk = new Struk08(size:20);
    Struk08 main = new Struk08Main08();
    // main.pushPakaian(stk);

    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;
        }
    }
}
```

```
Codeium: Refactor | Explain | Generate Javadoc | X
public void pushPakaian(Struk08 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();

        Struk08 d = new Struk08(noTra, tanggal, jumlah, total);

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

```
Codeium: Refactor | Explain | Generate Javadoc | X
public void kupon(Struk08 stk){
    System.out.println(x:"Data yang diambil");
    int i = 0;
    while (i < 5) {
        stk.pop();
        i++;
    }

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


Hasil compile seperti berikut

```
=====
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: 123
Tanggal (dd/mm/yyyy) : 12/02/2034
Jumlah: 5
Total Bayar: 6000
Apakah Anda akan menambahkan data baru ke stack (y/n) ? y
No Belanja: 82
Tanggal (dd/mm/yyyy) : 22/12/2034
Jumlah: 9
Total Bayar: 4000
Apakah Anda akan menambahkan data baru ke stack (y/n) ? y
No Belanja: 45
Tanggal (dd/mm/yyyy) : 23/10/2024
Jumlah: 8
Total Bayar: 9000
Apakah Anda akan menambahkan data baru ke stack (y/n) ? y
No Belanja: 125
Tanggal (dd/mm/yyyy) : 12/12/1093
Jumlah: 5
Total Bayar: 9500
Apakah Anda akan menambahkan data baru ke stack (y/n) ? y
No Belanja: 98
Tanggal (dd/mm/yyyy) : 17/12/4053
Jumlah: 9
Total Bayar: 20340
Apakah Anda akan menambahkan data baru ke stack (y/n) ? y
No Belanja: 812
Tanggal (dd/mm/yyyy) : 22/23/4023
Jumlah: 9
Total Bayar: 56000
Apakah Anda akan menambahkan data baru ke stack (y/n) ? y
No Belanja: 712
Tanggal (dd/mm/yyyy) : 12/03/4565
Jumlah: 89
Total Bayar: 120000
```

```
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: 712 12/03/4565 89 120000
Data yang keluar: 812 22/23/4023 9 56000
Data yang keluar: 98 17/12/4053 9 20340
Data yang keluar: 125 12/12/1093 5 9500
Data yang keluar: 45 23/10/2024 8 9000

Data yang tersisa
Isi stack:
82 22/12/2034 9 4000
123 12/02/2034 5 6000

=====
Pilihan

1. Push
2. Pop
3. Peek
4. Print
5. Kupon
6. Keluar
Pilih salah satu (1/2/3/4/5/6) : 
```