

**LAPORAN PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA**

**PRAKTIKUM 14
COLLECTION**



**Oleh:
JESSICA AMELIA
2341760185
SIB 1A/16**

**PROGRAM STUDI SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2023/2024**

Praktikum - Implementasi ArrayList

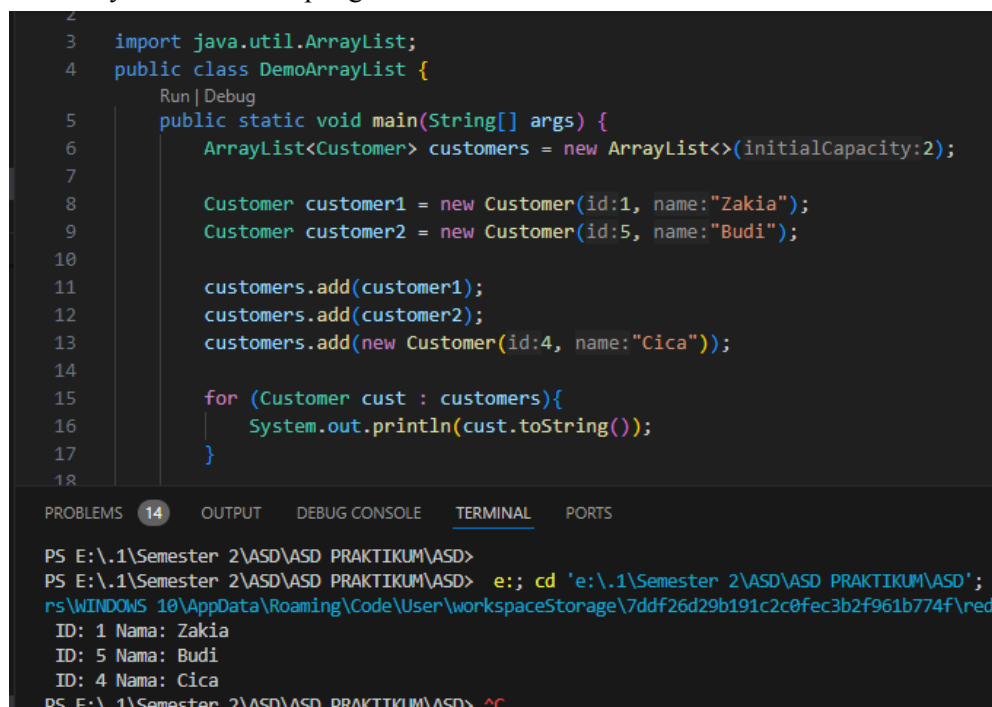
1. Buat file DemoArrayList.java. Lakukan import java.util.ArrayList;
2. Pada fungsi main(), instansiasi collection baru dengan nama customers bertipe ArrayList of Customer dengan size 2. Selanjutnya, buat object customer1 dan customer2 kemudian tambahkan ke dalam ArrayList customers dengan method add.
3. Gunakan looping dengan foreach untuk mencetak data customers



```
1 package PRAKTIKUM14;
2
3 import java.util.ArrayList;
4 public class DemoArrayList {
5     public static void main(String[] args) {
6         ArrayList<Customer> customers = new ArrayList<>(initialCapacity:2);
7
8         Customer customer1 = new Customer(id:1, name:"Zakia");
9         Customer customer2 = new Customer(id:5, name:"Budi");
10
11         customers.add(customer1);
12         customers.add(customer2);
13
14         for (Customer cust : customers){
15             System.out.println(cust.toString());
16         }
17     }
18 }
```

4. Cobalah tambahkan object customer baru ke dalam customers. Apakah object dapat ditambahkan meskipun melebihi kapasitas?

Jawab : Objek dapat ditambahkan. Meskipun memberikan kapasitas 2 awal pada ArrayList ketika membuat objeknya, itu hanya berfungsi sebagai ukuran awal dari array internal yang digunakan oleh ArrayList untuk menyimpan elemen-elemen. Jika kapasitas ini terlampaui, ArrayList secara otomatis akan menyesuaikan ukurannya dengan menambah kapasitas internalnya untuk menampung elemen tambahan.



```
3 import java.util.ArrayList;
4 public class DemoArrayList {
5     public static void main(String[] args) {
6         ArrayList<Customer> customers = new ArrayList<>(initialCapacity:2);
7
8         Customer customer1 = new Customer(id:1, name:"Zakia");
9         Customer customer2 = new Customer(id:5, name:"Budi");
10
11         customers.add(customer1);
12         customers.add(customer2);
13         customers.add(new Customer(id:4, name:"Cica"));
14
15         for (Customer cust : customers){
16             System.out.println(cust.toString());
17         }
18     }
19 }
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS E:\.1\Semester 2\ASD\ASD PRAKTIKUM\ASD>
PS E:\.1\Semester 2\ASD\ASD PRAKTIKUM\ASD> e;; cd 'e:\.1\Semester 2\ASD\ASD PRAKTIKUM\ASD'; &
rs\WINDOWS 10\AppData\Roaming\Code\User\workspaceStorage\7ddf26d29b191c2c0fec3b2f961b774f\redl
ID: 1 Nama: Zakia
ID: 5 Nama: Budi
ID: 4 Nama: Cica
PS E:\.1\Semester 2\ASD\ASD PRAKTIKUM\ASD> ^C

5. Compile dan run kode program, di mana object yang baru ditambahkan? Di awal, di tengah, atau di akhir collection?

Jawab : Object ditambahkan di akhir collection.

6. Untuk menambahkan object baru pada index tertentu, lakukan sebagai berikut

```
ArrayList<Customer> customers = new ArrayList<>(2);

Customer customer1 = new Customer(1, "Zakia");
Customer customer2 = new Customer(5, "Budi");

customers.add(customer1);
customers.add(customer2);

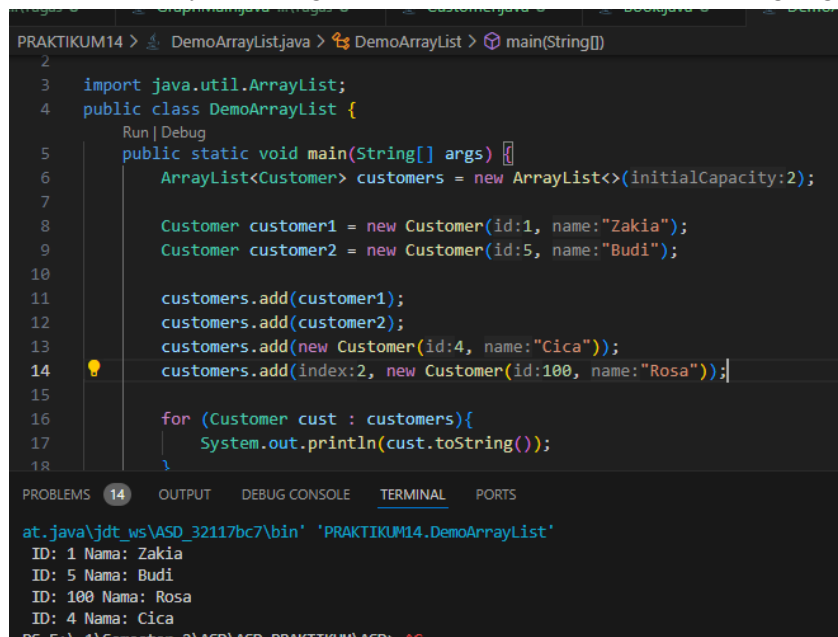
customers.add(new Customer(4, "Cica"));

customers.add(2, new Customer(100, "Rosa"));

for (Customer cust : customers) {
    System.out.println(cust.toString());
}
```

7. Compile dan run kode program. Index pada ArrayList dimulai dari 0 atau 1?

Jawab : Index pada ArrayList dimulai dari 0, lalu pada saat menyisipkan Customer dengan id 100 dan nama "Rosa" pada indeks 2. Elemen yang ada di indeks 2 sebelumnya ("Cica") dan elemen sesudahnya akan digeser ke kanan. Ini menunjukkan bahwa indeks dalam ArrayList di Java dimulai dari 0. Ketika menambahkan elemen pada indeks 2, elemen yang ada di posisi itu dan seterusnya akan digeser ke kanan untuk memberikan ruang bagi elemen baru.



The screenshot shows an IDE window titled 'PRAKTIKUM14' with a file named 'DemoArrayList.java'. The code is as follows:

```
2
3 import java.util.ArrayList;
4 public class DemoArrayList {
5     public static void main(String[] args) {
6         ArrayList<Customer> customers = new ArrayList<>(initialCapacity:2);
7
8         Customer customer1 = new Customer(id:1, name:"Zakia");
9         Customer customer2 = new Customer(id:5, name:"Budi");
10
11         customers.add(customer1);
12         customers.add(customer2);
13         customers.add(new Customer(id:4, name:"Cica"));
14         customers.add(index:2, new Customer(id:100, name:"Rosa"));
15
16         for (Customer cust : customers){
17             System.out.println(cust.toString());
18         }
19     }
20 }
```

The output in the terminal is:

```
at.java\jdt_ws\ASD_32117bc7\bin' PRAKTIKUM14.DemoArrayList'
ID: 1 Nama: Zakia
ID: 5 Nama: Budi
ID: 100 Nama: Rosa
ID: 4 Nama: Cica
```

8. Untuk mengetahui posisi dari suatu objek, gunakan method indexOf()

```
...Tugas U | GraphMain.java ...Tugas U | Customer.java U | Book.java U
PRAKTIKUM14 > DemoArrayList.java > DemoArrayList > main(String[])
4 public class DemoArrayList {
5     public static void main(String[] args) {
8         Customer customer1 = new Customer(id:1, name:"Zakia");
9         Customer customer2 = new Customer(id:5, name:"Budi");
11
12         customers.add(customer1);
13         customers.add(customer2);
14         customers.add(new Customer(id:4, name:"Cica"));
15         customers.add(index:2, new Customer(id:100, name:"Rosa"));
16
17         System.out.println(customers.indexOf(customer2));
18
19         for (Customer cust : customers){
20             System.out.println(cust.toString());
21         }
22     }

```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
at.java\jdt_ws\ASD_32117bc7\bin' 'PRAKTIKUM14.DemoArrayList'
1
ID: 1 Nama: Zakia
ID: 5 Nama: Budi
ID: 100 Nama: Rosa
ID: 4 Nama: Cica

```

9. Untuk mengembalikan object pada index tertentu, gunakan method get()

```
...Tugas U | GraphMain.java ...Tugas U | Customer.java U | Book.java U
PRAKTIKUM14 > DemoArrayList.java > DemoArrayList > main(String[])
4 public class DemoArrayList {
5     public static void main(String[] args) {
8         Customer customer1 = new Customer(id:1, name:"Zakia");
9         Customer customer2 = new Customer(id:5, name:"Budi");
11
12         customers.add(customer1);
13         customers.add(customer2);
14         customers.add(new Customer(id:4, name:"Cica"));
15         customers.add(index:2, new Customer(id:100, name:"Rosa"));
16
17         System.out.println(customers.indexOf(customer2));
18
19         Customer customer = customers.get(index:1);
20         System.out.println(customer.name);
21         customer.name = "Budi Utomo";
22
23         for (Customer cust : customers){

```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
at.java\jdt_ws\ASD_32117bc7\bin' 'PRAKTIKUM14.DemoArrayList'
1
Budi
ID: 1 Nama: Zakia
ID: 5 Nama: Budi Utomo
ID: 100 Nama: Rosa
ID: 4 Nama: Cica

```

10. Cobalah hapus angka 2 saat instansiasi object customers. Apakah ArrayList dapat diinstansiasi tanpa harus menentukan size di awal?

Jawab : Ya, ArrayList di Java dapat diinstansiasi tanpa harus menentukan ukuran awal. Jika ukuran awal tidak ditentukan, ArrayList akan menggunakan kapasitas default awal, yang kemudian akan bertambah secara dinamis sesuai kebutuhan ketika elemen-elemen baru ditambahkan.

```

1 package PRAKTIKUM14;
2
3 import java.util.ArrayList;
4 public class DemoArrayList {
5     Run | Debug
6     public static void main(String[] args) {
7         ArrayList<Customer> customers = new ArrayList<>();
8
9         Customer customer1 = new Customer(id:1, name:"Zakia");
10        Customer customer2 = new Customer(id:5, name:"Budi");
11
12        customers.add(customer1);
13        customers.add(customer2);
14        customers.add(new Customer(id:4, name:"Cica"));
15        customers.add(index:2, new Customer(id:100, name:"Rosa"));
16
17        System.out.println(customers.indexOf(customer2));
18    }
19 }

```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS E:\.1\Semester 2\ASD\ASD PRAKTIKUM\ASD> e;; cd 'e:\.1\Semester 2\ASD\ASD PRAKTIKUM\ASD'
rs\WINDOWS 10\AppData\Roaming\Code\User\workspaceStorage\7ddf26d29b191c2c0fec3b2f961b77
1
Budi
ID: 1 Nama: Zakia
ID: 5 Nama: Budi Utomo
ID: 100 Nama: Rosa
ID: 4 Nama: Cica
PS E:\.1\Semester 2\ASD\ASD PRAKTIKUM\ASD> ^C

```

11. Anda juga dapat menambahkan sekumpulan customer baru ke dalam ArrayList secara sekaligus. Misalnya terdapat ArrayList newCustomers. Tambahkan seluruh object customer sekaligus ke dalam customers.

```

PRAKTIKUM14 > DemoArrayList.java > DemoArrayList > main(String[])
4 public class DemoArrayList {
5     public static void main(String[] args) {
6
7
8
9
10
11
12
13
14
15
16
17
18
19     System.out.println(customer.name);
20     customer.name = "Budi Utomo";
21
22
23     ArrayList<Customer> newCustomers = new ArrayList<>();
24     newCustomers.add(new Customer(id:201, name:"Della"));
25     newCustomers.add(new Customer(id:202, name:"Victor"));
26     newCustomers.add(new Customer(id:203, name:"Sarah"));
27
28     customers.addAll(newCustomers);
29
30     for (Customer cust : customers){
31         System.out.println(cust.toString());
32     }
33 }

```

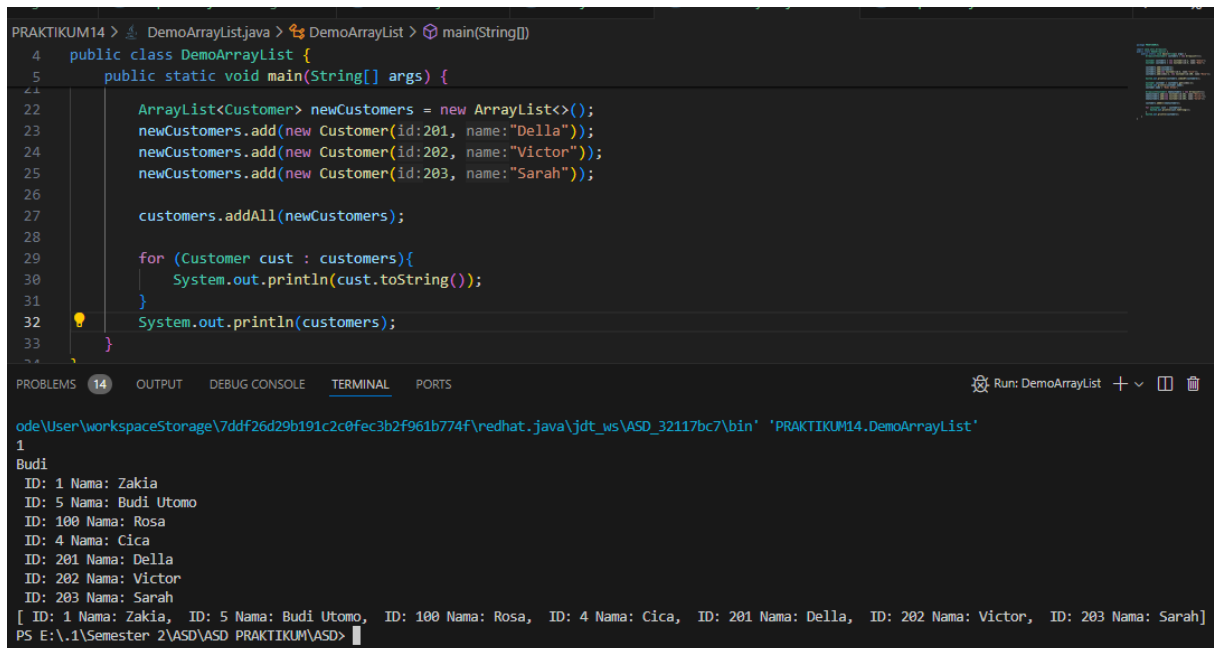
PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS E:\.1\Semester 2\ASD\ASD PRAKTIKUM\ASD> e;; cd 'e:\.1\Semester 2\ASD\ASD PRAKTIKUM\ASD'
rs\WINDOWS 10\AppData\Roaming\Code\User\workspaceStorage\7ddf26d29b191c2c0fec3b2f961b77
1
Budi
ID: 1 Nama: Zakia
ID: 5 Nama: Budi Utomo
ID: 100 Nama: Rosa
ID: 4 Nama: Cica
ID: 201 Nama: Della
ID: 202 Nama: Victor
ID: 203 Nama: Sarah
PS E:\.1\Semester 2\ASD\ASD PRAKTIKUM\ASD> 

```

12. Karena sudah menyediakan method toString(), pengecekan data customers untuk proses debugging juga dapat dilakukan lebih sederhana dengan cara berikut



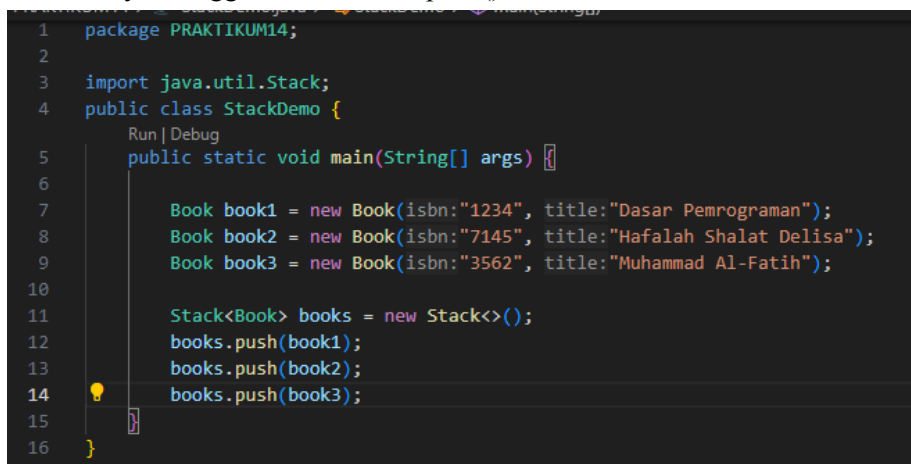
```
PRAKTIKUM14 > DemoArrayList.java > DemoArrayList > main(String[])
4 public class DemoArrayList {
5     public static void main(String[] args) {
22         ArrayList<Customer> newCustomers = new ArrayList<>();
23         newCustomers.add(new Customer(id:201, name:"Della"));
24         newCustomers.add(new Customer(id:202, name:"Victor"));
25         newCustomers.add(new Customer(id:203, name:"Sarah"));
26
27         customers.addAll(newCustomers);
28
29         for (Customer cust : customers){
30             System.out.println(cust.toString());
31         }
32         System.out.println(customers);
33     }
}
```

Run: DemoArrayList

```
ode\User\workspaceStorage\7ddf26d29b191c2c0fec3b2f961b774f\redhat.java\jdt_ws\ASD_32117bc7\bin' 'PRAKTIKUM14.DemoArrayList'
1
Budi
ID: 1 Nama: Zakia
ID: 5 Nama: Budi Utomo
ID: 100 Nama: Rosa
ID: 4 Nama: Cica
ID: 201 Nama: Della
ID: 202 Nama: Victor
ID: 203 Nama: Sarah
[ ID: 1 Nama: Zakia, ID: 5 Nama: Budi Utomo, ID: 100 Nama: Rosa, ID: 4 Nama: Cica, ID: 201 Nama: Della, ID: 202 Nama: Victor, ID: 203 Nama: Sarah]
PS E:\1\Semester 2\ASD\ASD PRAKTIKUM\ASD>
```

Praktikum - Implementasi Stack

1. Buat file StackDemo.java. Lakukan import java.util.Stack;
2. Pada fungsi main() buat beberapa object bertipe Book.
3. Instansiasi object books bertipe Stack of Book kemudian tambahkan object yang sudah dibuat ke dalamnya menggunakan method push()



```
1 package PRAKTIKUM14;
2
3 import java.util.Stack;
4 public class StackDemo {
5     public static void main(String[] args) {
6
7         Book book1 = new Book(isbn:"1234", title:"Dasar Pemrograman");
8         Book book2 = new Book(isbn:"7145", title:"Hafalah Shalat Delisa");
9         Book book3 = new Book(isbn:"3562", title:"Muhammad Al-Fatih");
10
11         Stack<Book> books = new Stack<>();
12         books.push(book1);
13         books.push(book2);
14         books.push(book3);
15     }
16 }
```

4. Class Stack juga sudah memiliki method pop() dan peek() seperti yang Anda diimplementasikan secara manual pada praktikum sebelumnya

```

9      Book book3 = new Book(isbn:"3562", title:"Muhammad Al-Fatih");
10
11      Stack<Book> books = new Stack<>();
12      books.push(book1);
13      books.push(book2);
14      books.push(book3);
15
16      Book temp = books.peek();
17
18      if(temp != null){
19          System.out.println(temp.toString());
20      }
21
22      Book temp2 = books.pop();
23
24      if(temp2 != null){
25          System.out.println(temp2.toString());
26      }
27

```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS E:\.1\Semester 2\ASD\ASD PRAKTIKUM\ASD> e;; cd 'e:\.1\Semester 2\ASD\ASD PRAKTIKUM\ASD'
xe' '-cp' 'C:\Users\WINDOWS_10\AppData\Roaming\Code\User\workspaceStorage\7ddf26d29b191c2c6
n' 'PRAKTIKUM14.StackDemo'
ISBN: 3562 Title: Muhammad Al-Fatih
ISBN: 3562 Title: Muhammad Al-Fatih
PS E:\.1\Semester 2\ASD\ASD PRAKTIKUM\ASD>

```

5. Mengapa perlu ada pengecekan (temp != null)?

Jawab : Karena untuk menghindari NullPointerException(NPE) ketika mencoba mengakses method toString() pada objek temp. Method peek() dapat mengembalikan null jika stack kosong. Dengan menambahkan pengecekan temp!= null , dapat memastikan bahwa hanya mencoba mengakses metode toString() jika objek tersebut tidak null. Jika objek tersebut null, maka kode dalam blok if tidak akan dieksekusi, sehingga menghindari kemungkinan NPE.

6. Lakukan looping untuk mencetak data buku pada stack

```

4      public class StackDemo {
5          public static void main(String[] args) {
13          books.push(book2);
14          books.push(book3);
15
16          Book temp = books.peek();
17
18          if(temp != null){
19              System.out.println(temp.toString());
20          }
21
22          Book temp2 = books.pop();
23
24          if(temp2 != null){
25              System.out.println(temp2.toString());
26          }
27
28          for (Book book : books){
29              System.out.println(book.toString());
30          }
31      }

```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

& 'C:\Program Files\Java\jdk1.8.0_121\bin\java.exe' '-cp' 'C:\Users
ec3b2f961b774f\redhat.java\jdt_ws\ASD_32117bc7\bin' 'PRAKTIKUM14.Sta
ISBN: 3562 Title: Muhammad Al-Fatih
ISBN: 3562 Title: Muhammad Al-Fatih
ISBN: 1234 Title: Dasar Pemrograman
ISBN: 7145 Title: Hafalah Shalat Delisa
PS E:\.1\Semester 2\ASD\ASD PRAKTIKUM\ASD>

```

7. Jika diperlukan pada proses debugging, books juga dapat dicetak dengan cara berikut

```

4 public class StackDemo {
5     public static void main(String[] args) {
14         books.push(book3);
15
16         Book temp = books.peek();
17
18         if(temp != null){
19             System.out.println(temp.toString());
20         }
21
22         Book temp2 = books.pop();
23
24         if(temp2 != null){
25             System.out.println(temp2.toString());
26         }
27
28         for (Book book : books){
29             System.out.println(book.toString());
30         }
31
32         System.out.println(books);
33     }
}

```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

ing\Code\User\workspaceStorage\7ddf26d29b191c2c0fec3b2f961b774f\redhat.java\jdt_w
ISBN: 3562 Title: Muhammad Al-Fatih
ISBN: 3562 Title: Muhammad Al-Fatih
ISBN: 1234 Title: Dasar Pemrograman
ISBN: 7145 Title: Hafalah Shalat Delisa
[ISBN: 1234 Title: Dasar Pemrograman, ISBN: 7145 Title: Hafalah Shalat Delisa]
PS E:\1\Semester 2\ASD\ASD PRAKTIKUM\ASD>

```

8. Bagaimana cara melakukan pencarian elemen pada stack menggunakan method search()?

Jawab :

```

34         int index = books.search(book2);
35         if (index != -1){
36             System.out.println("Buku ditemukan pada indeks: " + index);
37         } else{
38             System.out.println("x: " + "Buku tidak ditemukan dalam stack");
39         }
40     }
41 }

```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

c0fec3b2f961b774f\redhat.java\jdt_ws\ASD_32117bc7\bin' 'PRAKTIKUM14.StackDemo'
ISBN: 3562 Title: Muhammad Al-Fatih
ISBN: 3562 Title: Muhammad Al-Fatih
ISBN: 1234 Title: Dasar Pemrograman
ISBN: 7145 Title: Hafalah Shalat Delisa
[ISBN: 1234 Title: Dasar Pemrograman, ISBN: 7145 Title: Hafalah Shalat Delisa]
Buku ditemukan pada indeks: 1
PS E:\1\Semester 2\ASD\ASD PRAKTIKUM\ASD>

```

Praktikum - Implementasi TreeSet

1. Buat file TreeSetDemo.java kemudian import java.util.TreeSet;
2. Tambahkan fungsi main() kemudian instansiasi object TreeSet of String. Tambahkan beberapa nilai bertipe String ke dalam TreeSet


```
PRAKTIKUM14 > TreeSetDemo.java > ...
1  package PRAKTIKUM14;
2
3  import java.util.TreeSet;
4  public class TreeSetDemo {
5      Run | Debug
6      public static void main(String[] args) {
7          TreeSet<String> fruits = new TreeSet<>();
8
9          fruits.add(e:"Mangga");
10         fruits.add(e:"Apel");
11         fruits.add(e:"Jeruk");
12         fruits.add(e:"Jambu");
13
14         for (String temp : fruits){
15             System.out.println(temp);
16         }
17     }

```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
C:\Program Files\Java\jdk1.8.0_121\bin\java.exe' '-cp' 'C:\Users\WINDO
2f961b774f\redhat.java\jdt_ws\ASD_32117bc7\bin' 'PRAKTIKUM14.TreeSetDe
Apel
Jambu
Jeruk
Mangga
PS E:\1\Semester 2\ASD\ASD_PRAKTIKUM\ASD>

```

3. Cetak data pada ts dengan looping

```
PRAKTIKUM14 > TreeSetDemo.java > ...
1  package PRAKTIKUM14;
2
3  import java.util.TreeSet;
4  public class TreeSetDemo {
5      Run | Debug
6      public static void main(String[] args) {
7          TreeSet<String> fruits = new TreeSet<>();
8
9          fruits.add(e:"Mangga");
10         fruits.add(e:"Apel");
11         fruits.add(e:"Jeruk");
12         fruits.add(e:"Jambu");
13
14         for (String temp : fruits){
15             System.out.println(temp);
16         }
17     }

```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
C:\Program Files\Java\jdk1.8.0_121\bin\java.exe' '-cp' 'C:\Users\WINDO
2f961b774f\redhat.java\jdt_ws\ASD_32117bc7\bin' 'PRAKTIKUM14.TreeSetDe
Apel
Jambu
Jeruk
Mangga
PS E:\1\Semester 2\ASD\ASD_PRAKTIKUM\ASD>

```

4. Compile dan run program. Mengapa urutan yang ditampilkan berbeda dengan urutan penambahan data ke dalam TreeSet fruits?

Jawab : Ketika menggunakan TreeSet di Java, elemen-elemen yang ditambahkan ke dalam set secara otomatis diurutkan berdasarkan urutan alami (natural ordering) elemen tersebut. Karena Tree yang diimplementasikan merupakan self-balancing binary search tree

Binary search tree adalah binary tree dengan aturan left-child < parent < rightchild, sehingga data tidak disimpan sesuai urutan penambahan. Untuk String, urutan alami adalah urutan leksikografis atau alfabetis. Oleh karena itu, meskipun elemen-elemen ditambahkan ke dalam TreeSet dalam urutan tertentu, mereka akan disimpan dan ditampilkan dalam urutan yang sudah diurutkan.

5. Tambahkan kode program sebagai berikut:

```

15     }
16
17     System.out.println("First: " + fruits.first());
18     System.out.println("Last: " + fruits.last());
19
20     fruits.remove(o:"Jeruk");
21     System.out.println("Setelah remove " + fruits);
22
23     fruits.pollFirst();
24     System.out.println("Setelah poll first " + fruits);
25
26     fruits.pollLast();
27     System.out.println("Setelah poll last " + fruits);
28
29 }
30 }

```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

a.exe' '-cp' 'C:\Users\WINDOWS_10\AppData\Roaming\Code\User\workspaceStorage\
\bin' 'PRAKTIKUM14.TreeSetDemo'
Apel
Jambu
Jeruk
Mangga
First: Apel
Last: Mangga
Setelah remove [Apel, Jambu, Mangga]
Setelah poll first [Jambu, Mangga]
Setelah poll last [Jambu]
PS E:\1\Semester 2\ASD\ASD PRAKTIKUM\ASD>

```

6. Apa yang dilakukan oleh method first(), last(), remove(), pollFirst(), dan pollLast()?

Jawab :

- Method first() mengembalikan elemen pertama (terkecil), yaitu "Apel".
- Method last() mengembalikan elemen terakhir (terbesar), yaitu "Mangga".
- Method remove() menghapus elemen "Jeruk" dari **TreeSet**, sehingga **TreeSet** sekarang berisi ["Apel", "Jambu", "Mangga"].
- Method pollFirst() menghapus dan mengembalikan elemen pertama, yaitu "Apel", sehingga **TreeSet** sekarang berisi ["Jambu", "Mangga"].
- Method pollLast() menghapus dan mengembalikan elemen terakhir, yaitu "Mangga", sehingga **TreeSet** sekarang berisi ["Jambu"].

Praktikum – Sorting

Selain pencarian data menggunakan method get() atau search(), Java Collection Framework juga menyediakan fungsi untuk melakukan pengurutan data. Untuk melakukan pengurutan data String, int, atau primitive data type lain, Anda dapat melakukan cara berikut:

```

ArrayList<String> daftarSiswa = new ArrayList<>();
daftarSiswa.add("Zainab");
daftarSiswa.add("Andi");
daftarSiswa.add("Rara");
Collections.sort(daftarSiswa);

System.out.println(daftarSiswa);

```

Jawab :

```
1 package PRAKTIKUM14;
2
3 import java.util.ArrayList;
4 import java.util.Collections;
5 public class SortingDemo {
6     public static void main(String[] args) {
7         ArrayList<String> daftarSiswa = new ArrayList<>();
8         daftarSiswa.add(e:"Zainab");
9         daftarSiswa.add(e:"Andi");
10        daftarSiswa.add(e:"Rara");
11        Collections.sort(daftarSiswa);
12
13        System.out.println(daftarSiswa);
14    }
15 }
```

Run | Debug

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS E:\.1\Semester 2\ASD\ASD PRAKTIKUM\ASD> & 'C:\Program Files\Java\jdk1.8. e\User\workspaceStorage\7ddf26d29b191c2c0fec3b2f961b774f\redhat.java\jdt_ws [Andi, Rara, Zainab]

PS E:\.1\Semester 2\ASD\ASD PRAKTIKUM\ASD>

Khusus untuk collection of objects, pengurutan data harus menentukan berdasarkan atribut mana pengurutan dilakukan. Misalnya ingin dilakukan pengurutan data customer pada praktikum di atas berdasarkan atribut name. Gunakan kode program berikut (berikan nama variabel c1 dan c2 secara random. Anda bisa menggunakan nama variabel lainnya)

```
customers.sort((c1,c2)->c1.name.compareTo(c2.name));

System.out.println(customers);
```

Jawab : pada class DemoArrayList.java

```
4 public class DemoArrayList {
5     public static void main(String[] args) {
6         newCustomers.add(new Customer(12123, "Nama: Budi Utomo"));
7
8         customers.addAll(newCustomers);
9
10        for (Customer cust : customers){
11            System.out.println(cust.toString());
12        }
13        System.out.println(customers);
14
15        customers.sort((c1,c2)->c1.name.compareTo(c2.name));
16        System.out.println(customers);
17    }
18 }
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

1
Budi
ID: 1 Nama: Zakia
ID: 5 Nama: Budi Utomo
ID: 100 Nama: Rosa
ID: 4 Nama: Cica
ID: 201 Nama: Della
ID: 202 Nama: Victor
ID: 203 Nama: Sarah
[ID: 1 Nama: Zakia, ID: 5 Nama: Budi Utomo, ID: 100 Nama: Rosa, ID: 4 Nama: Cica, ID: 201 Nama: Della, ID: 202 Nama: Victor, ID: 203 N
ama: Sarah]
[ID: 5 Nama: Budi Utomo, ID: 4 Nama: Cica, ID: 201 Nama: Della, ID: 100 Nama: Rosa, ID: 203 Nama: Sarah, ID: 202 Nama: Victor, ID: 1 N
ama: Zakia]
PS E:\.1\Semester 2\ASD\ASD PRAKTIKUM\ASD>