



Java Fundamental | Bootcamp

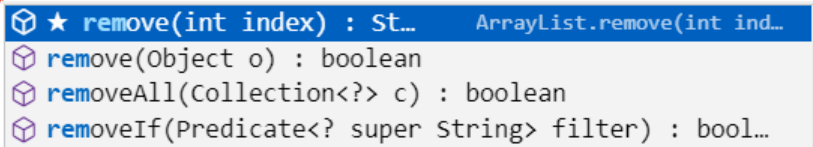
Array List, List, Collection

ArrayList – Remove Data

- Sintaks untuk hapus data → sayuran.re tekan ctrl + space pilih remove(int index) , lalu pilih indexnya yang mau dihapus yang mana

```
public class KelasArrayListDeleteData {
    Run | Debug
    public static void main(String[] args) {
        ArrayList<String> sayuran = new ArrayList<>();
        sayuran.add("BROKOLI");
        sayuran.add("BAYAM");
        for (int i = 0; i < sayuran.size(); i++) {
            System.out.println(sayuran.get(i));
        }
        System.out.println();
        System.out.println("ADD SELEDRI: ");
        sayuran.add("SELEDRI"); //tambah SELEDRI
        for (int i = 0; i < sayuran.size(); i++) {
            System.out.println(sayuran.get(i));
        }

        System.out.println();
        System.out.println("REMOVE BAYAM;");
        sayuran.rem
    }
}
```



ArrayList – Remove Data

```
KelasArrayListDeleteData.java
5 public class KelasArrayListDeleteData {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         ArrayList<String> sayuran = new ArrayList<>();
10        sayuran.add("BROKOLI");
11        sayuran.add("BAYAM");
12        sayuran.add("KANGKUNG");
13        sayuran.add("TOMAT");
14        sayuran.add("TIMUN");
15        for (int i = 0; i < sayuran.size(); i++) {
16            System.out.println(sayuran.get(i));
17        }
18
19        System.out.println();
20        System.out.println("Hapus BROKOLI dan KANGKUNG");
21
22        sayuran.remove(0); // remove index0 yaitu BROKOLI
23        for (int i = 0; i < sayuran.size(); i++) {
24            System.out.println(sayuran.get(i));
25        }
26
27        sayuran.remove(1);
28        // kenapa index1 yg diremove? karena indexnya naik
29        // BAYAM KANGKUNG TIMUN
30        // 0     1     2
31        System.out.println();
32        System.out.println("Sayuran terkini");
33        for (int i = 0; i < sayuran.size(); i++) {
34            System.out.println(sayuran.get(i));
35        }
36    }
37 }
```

```
Console
<terminated> KelasArrayListDeleteData [J
BROKOLI
BAYAM
KANGKUNG
TOMAT
TIMUN

Hapus BROKOLI dan KANGKUNG
BAYAM
KANGKUNG
TOMAT
TIMUN

Sayuran terkini
BAYAM
TOMAT
TIMUN
```

ArrayList – Add Remove - Dynamic

```
public class KelasArrayListDeleteData {  
    Run | Debug  
    public static void main(String[] args) {  
        ArrayList<String> sayuran = new ArrayList<>();  
        sayuran.add("BROKOLI");  
        sayuran.add("BAYAM");  
        for (int i = 0; i < sayuran.size(); i++) {  
            System.out.println(sayuran.get(i));  
        }  
        System.out.println();  
        System.out.println("ADD SELEDRI: ");  
        sayuran.add("SELEDRI"); //tambah SELEDRI  
        for (int i = 0; i < sayuran.size(); i++) {  
            System.out.println(sayuran.get(i));  
        }  
  
        System.out.println();  
        System.out.println("REMOVE BAYAM;");  
        sayuran.remove(1); //hapus BAYAM --> BAYAM ada di Index1  
        for (int i = 0; i < sayuran.size(); i++) {  
            System.out.println(sayuran.get(i));  
        }  
    }  
}
```

BROKOLI
BAYAM

ADD SELEDRI:
BROKOLI
BAYAM
SELEDRI

REMOVE BAYAM;
BROKOLI
SELEDRI

ArrayList - Integer

- TipeDatanya harus menggunakan yang Besar → Integer

```
public class KelasArrayListInteger {
```

```
Run | Debug
```

```
public static void main(String[] args) {
```

```
    ArrayList<Integer> angkaInteger = new ArrayList<>();
```

```
    angkaInteger.add(999);
```

```
    angkaInteger.add(888);
```

```
    angkaInteger.add(777);
```

```
    for (int i = 0; i < angkaInteger.size(); i++) {
```

```
        System.out.println(angkaInteger.get(i));
```

```
    }
```

```
}
```

```
}
```

999

888

777

ArrayList - Double

```
public class KelasArrayListDouble {
```

Run | Debug

```
public static void main(String[] args) {
```

```
    ArrayList<Double> angkaDouble = new ArrayList<>();
```

```
    angkaDouble.add(999.99);
```

```
    angkaDouble.add(888.88);
```

```
    angkaDouble.add(777.77);
```

```
    for (int i = 0; i < angkaDouble.size(); i++) {
```

```
        System.out.println(angkaDouble.get(i));
```

```
    }
```

```
}
```

999.99

888.88

777.77

ArrayList - Float

```
public class KelasArrayListFloat {  
    Run | Debug  
    public static void main(String[] args) {  
        ArrayList<Float> angkaFloat = new ArrayList<>();  
        angkaFloat.add(999.99F);  
        angkaFloat.add(888.88F);  
        angkaFloat.add(777.77F);  
  
        for (int i = 0; i < angkaFloat.size(); i++) {  
            System.out.println(angkaFloat.get(i));  
        }  
    }  
}
```

999.99
888.88
777.77

ArrayList – Type Of

- ArrayList mempunyai 2 turunan:
 1. List
 2. Collection

ArrayList - List

- Ketik List lalu tekan ctrl + space pilih List – Java.util

```
public class KelasListDasar {
```

Run | Debug

```
public static void main(String[] args) {
```

List



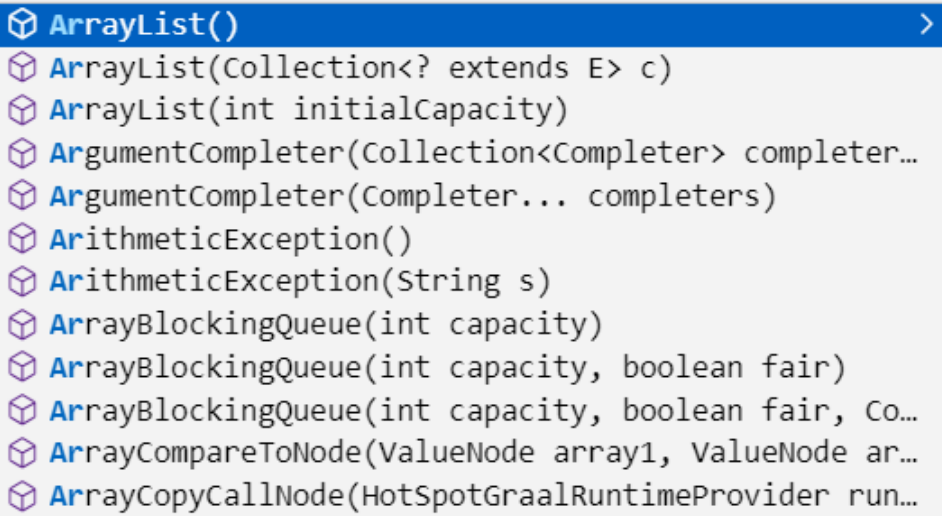
The screenshot shows an IDE window with a code editor and a class completion dropdown menu. The code in the editor is: `public class KelasListDasar {` on the first line, `public static void main(String[] args) {` on the second line, and `}` on the third line. The word `List` is typed at the end of the second line and is underlined with a red wavy line. The dropdown menu is open, showing a list of classes. The first item, `List - java.util`, is highlighted in blue. Other items include `ListAdapter - jdk.nashorn.internal.runtime`, `ListCellRenderer - javax.swing`, `ListDataEvent - javax.swing.event`, `ListDataHandler - javax.swing.plaf.basic.BasicCom...` (two instances), `ListDataListener - javax.swing.event`, `ListEditor - javax.swing.JSpinner`, `ListIterator - java.util`, `ListModel - javax.swing`, and `ListPainter - javax.swing.text.html.StyleSheet`.

- List - java.util `java.util.List`
- ListAdapter - jdk.nashorn.internal.runtime
- ListCellRenderer - javax.swing
- ListDataEvent - javax.swing.event
- ListDataHandler - javax.swing.plaf.basic.BasicCom...
- ListDataHandler - javax.swing.plaf.basic.BasicCom...
- ListDataHandler - javax.swing.plaf.basic.BasicLis...
- ListDataListener - javax.swing.event
- ListEditor - javax.swing.JSpinner
- ListIterator - java.util
- ListModel - javax.swing
- ListPainter - javax.swing.text.html.StyleSheet

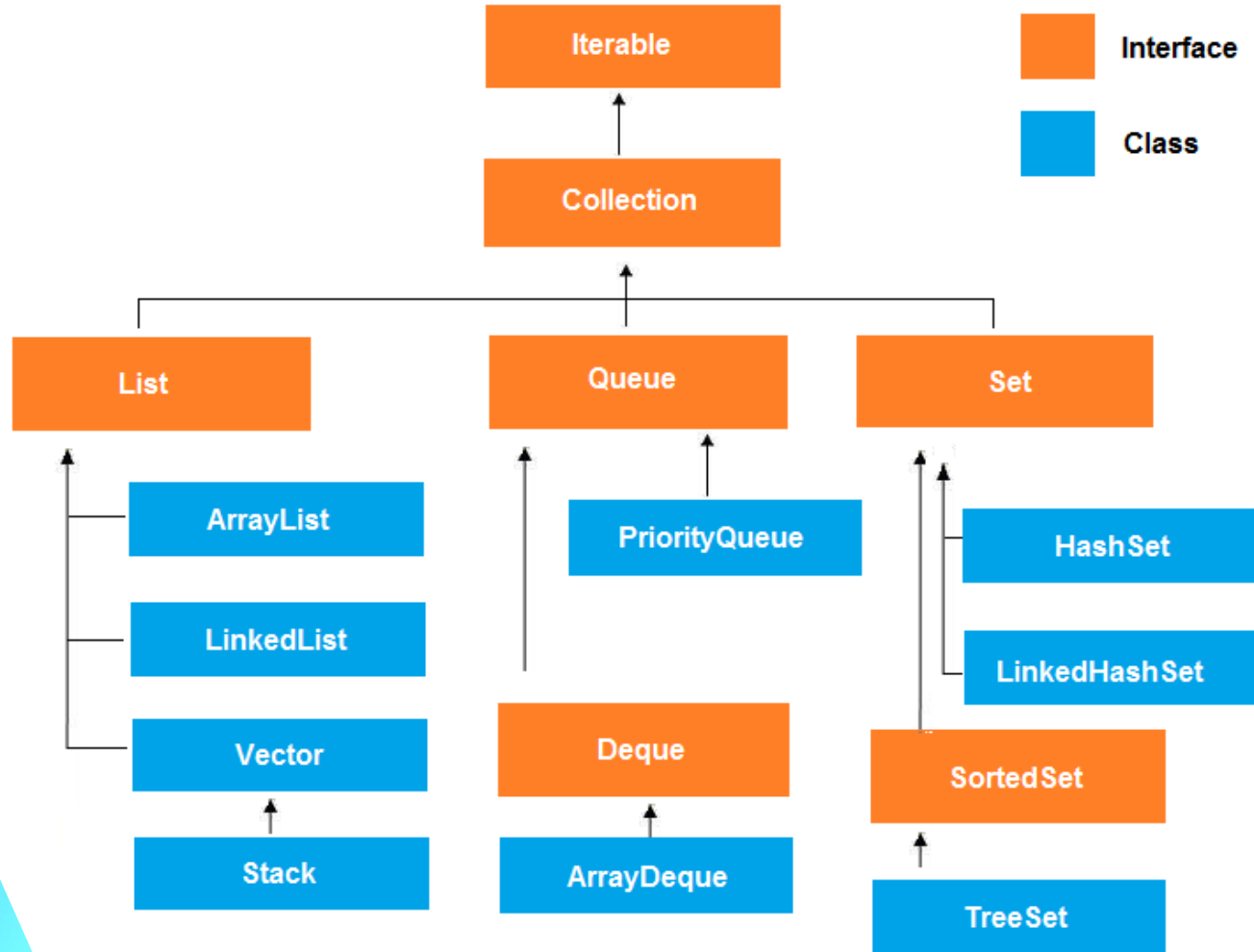
ArrayList - List

- Buat nama variabel arraynya contoh “buahList”
- Ketik new Arr lalu tekan ctrl + space pilih ArrayList – java.util

```
public class KelasListDasar {  
    Run | Debug  
    public static void main(String[] args) {  
        List<Integer> buahList = new Arr;  
    }  
}
```



ArrayList – List - Hierarchy



ArrayList - List

```
public class KelasListDasar {  
    Run | Debug  
    public static void main(String[] args) {  
        List<String> buahList = new ArrayList<>();  
    }  
}
```

Object Constructor

Object -> Tipe data untuk menentukan data yang akan dimasukkan ke buahList

ArrayList<>() -> adalah Constructor dari class ArrayList yang dia mengimplement List.

ArrayList - List

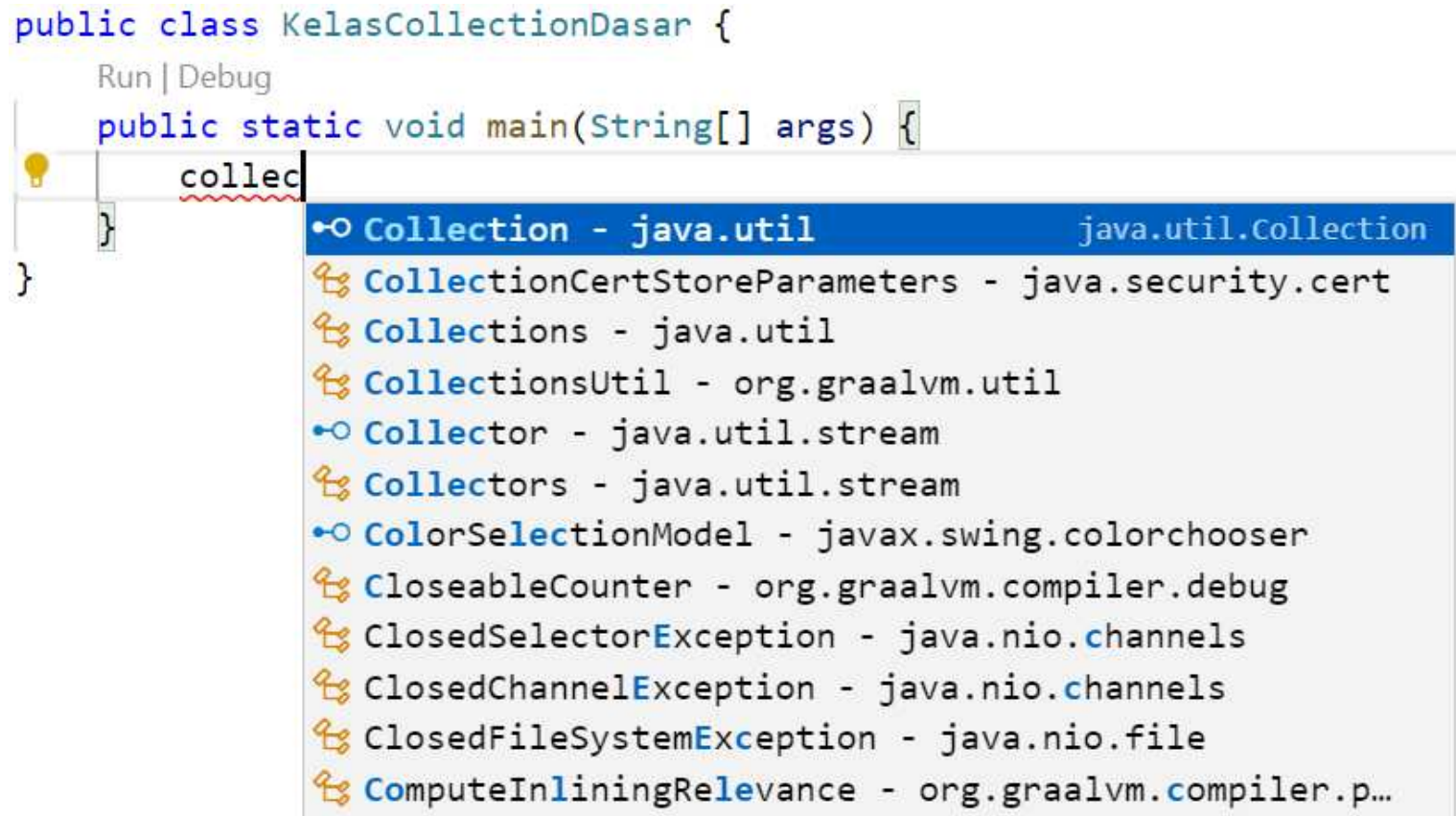
```
public class KelasListInput {  
    Run | Debug  
    public static void main(String[] args) {  
        List<String> buahList = new ArrayList<String>();  
        buahList.add("APEL");  
        buahList.add("ANGGUR");  
  
        for (int i = 0; i < buahList.size(); i++) {  
            System.out.println(buahList.get(i));  
        }  
    }  
}
```

APEL
ANGGUR

ArrayList - Collection

- Ketik Col lalu ctrl + space pilih Collection – java.util

```
public class KelasCollectionDasar {  
    Run | Debug  
    public static void main(String[] args) {  
        collec  
    }  
}
```



•	Collection - java.util	java.util.Collection
🔗	CollectionCertStoreParameters - java.security.cert	
🔗	Collections - java.util	
🔗	CollectionsUtil - org.graalvm.util	
•	Collector - java.util.stream	
🔗	Collectors - java.util.stream	
•	ColorSelectionModel - javax.swing.colorchooser	
🔗	CloseableCounter - org.graalvm.compiler.debug	
🔗	ClosedSelectorException - java.nio.channels	
🔗	ClosedChannelException - java.nio.channels	
🔗	ClosedFileSystemException - java.nio.file	
🔗	ComputeInliningRelevance - org.graalvm.compiler.p...	

ArrayList - Collection

```
public class KelasCollectionDasar {  
    Run | Debug  
    public static void main(String[] args) {  
        Collection<String> buahCol = new ArrayList<String>();  
    }  
}
```

ArrayList - Collection

```
public class KelasCollectionInput {  
    Run | Debug  
    public static void main(String[] args) {  
        Collection<String> buahCol = new ArrayList<String>();  
        buahCol.add("KIWI");  
        buahCol.add("BERRY");  
        buahCol.add("ANGGUR");  
  
        System.out.println(buahCol);  
    }  
}
```

[KIWI, BERRY, ANGGUR]

Collection VS List

COLLECTION	LIST
<p>Dalam penggunaan Collection, Collection hanya dapat melakukan penambahan atau penghapusan suatu data. Namun tidak bisa melihat data berdasarkan indexnya.</p>	<p>Pada List memungkinkan kita untuk dapat menambah, menghapus dsb. Selain itu juga memungkinkan untuk melihat data berdasarkan indexnya.</p>

Pengenalan OOP

Object Oriented Programming, adalah suatu Teknik pemrograman yang berorientasi pada object, jadi dalam penggunaannya kita akan membuat banyak object yang akan dipanggil, dibuat, atau di relasikan dengan object lainnya.