

VERSION 1.3
February 27, 2023



[STRUKTUR DATA]

MODUL 2, ARRAY LIST & LINKED LIST

DISUSUN OLEH:
MUHAMMAD SYAUQI AMIQ AMRULLAH
GILANG DWI DARMAWAN

DIAUDIT OLEH:
DIDIH RIZKI CHANDRANEGARA, S.KOM., M.KOM.

PRESENTED BY: TIM LAB-IT
UNIVERSITAS MUHAMMADIYAH MALANG

[STRUKTUR DATA]

PETUNJUK Pengerjaan Modul

Perhatikan petunjuk praktikum dibawah ini:

1. Wajib membaca materi modul, sewaktu – waktu dapat direview oleh asisten saat demo
2. Gunakan referensi yang disediakan modul dan referensi lain pada google (yang kredibel)
3. Latihan praktikum wajib dikerjakan pada praktikum minggu pertama secara bersama – sama di laboratorium dan tidak boleh dijadikan pekerjaan rumah
4. Tugas praktikum boleh dijadikan pekerjaan rumah dan di demokan kepada asisten pada praktikum minggu kedua
5. Memperhatikan kerapihan *source code* termasuk aturan penamaan Class, Method, Variable, File, dan lain - lainnya.
6. Segera lapor kepada asisten jika ada kesalahan pada modul praktikum.

PERSIAPAN MATERI

Mahasiswa diharapkan mempelajari materi sebelum mengerjakan tugas, materi yang tercakup antara lain:

1. Bahasa Pemrograman Java
2. Class dan Object
3. Enkapsulasi
4. Array

TUJUAN

Mahasiswa mampu menguasai dan menjelaskan konsep dari Struktur Data *Array List & Linked List*.

TARGET MODUL

Mahasiswa mampu memahami:

1. Collection tanpa menggunakan generics
2. Collection menggunakan generics
3. ArrayList
4. LinkedList

PERSIAPAN SOFTWARE/APLIKASI

1. Java Development Kit
2. Java Runtime Environment
3. IDE (Intellij IDEA, Eclipse, Netbeans, dll)

REFERENSI MATERI

Youtube

https://www.youtube.com/watch?v=GTjuHxcBeJg&ab_channel=KelasTerbuka
https://www.youtube.com/watch?v=yav_GO-sVbU&ab_channel=SekolahKoding
https://www.youtube.com/watch?v=1nRj4ALuw7A&ab_channel=BroCode

Artikel

<https://codepolitan.com/blog/menggunakan-arraylist-di-java>
<https://www.petanikode.com/java-array/>
<https://www.javatpoint.com/java-arraylist>
<https://www.geeksforgeeks.org/arraylist-in-java/>
https://www.w3schools.com/java/java_arraylist.asp

Note:

Dari referensi tersebut mungkin terdapat sedikit perbedaan satu sama yang lain, cukup pahami konsepnya dan terapkan pada kasus di modul ini.

MATERI POKOK

A. Collection

Collection adalah suatu obyek yang bisa digunakan untuk menyimpan sekumpulan obyek. Kelas Collection merupakan kelas generic, sehingga dapat digunakan untuk menampung objek yang memiliki tipe data berbeda.

a. Collection tanpa menggunakan generics

```
public class CellCollection {
    Cell[] cells;
    int index;
    public void add(Cell c) {
        Cells[index]=c;
        index++;
    }
    public Cell get(int i) {
        return cells[i];
    }
    //more methods...
}
```

Kita juga memerlukan konstruktor untuk menginisialisasi ukuran array.

```
Public
CellCollection(int
size){

        cells = new
        Cell[size];
    }
}
```

b. Collection menggunakan generics

```
public class CellGenericCollection<T> {
    T[] cells;
    int index;
    public void add(T c) {
        cells[index]=c;
        index++;
    }
    public T get(int i) {
        return cells[i];
    }
}
```

B. ArrayList

ArrayList adalah sebuah class untuk array yang resizable, class ini mengimplementasikan Interface List. ArrayList hanya mendukung tipe data object dan tidak mendukung tipe data primitif. Index dalam ArrayList dimulai dari 0 dan di akhiri dengan n-1 .

- **Interface List**

List digunakan untuk menyimpan sekumpulan objek berdasarkan urutan masuk (ordered) dan menerima duplikat. Cara penyimpanannya seperti array, oleh sebab itu memiliki posisi awal dan posisi akhir, menyisipkan objek pada posisi tertentu, mengakses dan menghapus isi list, dimana semua proses ini selalu didasarkan pada urutannya. Class-class yang mengimplementasikan interface List adalah Vector, Stack, Linked List dan Array List.

- **Method untuk ArrayList**

Beberapa method yang ada untuk penggunaan ArrayList

N o.	Class Metthods	Description
1.	<code>boolean add(E e)</code>	Appends the specified element to the end of this list.
2.	<code>void add(int index, E element)</code>	Inserts the specified element at the specified position in this list.
3.	<code>boolean addAll(Collection<? extends E> c)</code>	Appends all of the elements in the specified collection to the end of this list, in the order that they.
4.	<code>boolean addAll(int index, Collection<? extends E> c)</code>	Inserts all of the elements in the specified collection into this list, starting at the specified position.
5.	<code>void clear()</code>	Removes all of the elements from this list.
6.	<code>object clone()</code>	Returns a shallow copy of this ArrayList instance.
7.	<code>boolean contains(Object o)</code>	Returns true if this list contains the specified element.
8.	<code>void ensureCapacity(int minCapacity)</code>	Increases the capacity of this ArrayList instance, if necessary, to ensure that it can hold at least the number of elements specified by the minimum capacity argument.
9.	<code>E get(int index)</code>	Returns the element at the specified position in this list.
10.	<code>int indexOf(Object o)</code>	Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element.
11.	<code>boolean isEmpty()</code>	Returns true if this list contains no elements.
12.	<code>int lastIndexOf(Object o)</code>	Returns the index of the last occurrence of the specified element in this list, or -1 if this list does not contain the element.
13.	<code>E remove(int index)</code>	Removes the element at the specified position in this list.
14.	<code>boolean remove(Object o)</code>	Removes the first occurrence of the specified element from this list, if it is present.

15.	<code>protected void removeRange(int fromIndex, int toIndex)</code>	Removes from this list all of the elements whose index is between fromIndex, inclusive, and toIndex, exclusive.
16.	<code>E set(int index, E element)</code>	Replaces the element at the specified position in this list with the specified element.
17.	<code>int size()</code>	Returns the number of elements in this list.
18.	<code>Object[] toArray()</code>	Returns an array containing all of the

		elements in this list in proper sequence (from first to last element).
19.	<code><T> T[] toArray(T[] a)</code>	Returns an array containing all of the elements in this list in proper sequence (from first to last element); the runtime type of the returned array is that of the specified array.
20.	<code>void trimToSize()</code>	Trims the capacity of this ArrayList instance to be the list's current size.

C. LinkedList

LinkedList dapat diilustrasikan seperti kereta api, dimana kereta api terdiri dari gerbong-gerbong yang saling terhubung yang dapat mengangkut penumpang (Data). Gerbong (Node/Simpul) disini berfungsi untuk menyimpan data. LinkedList merupakan struktur data yang setiap datanya memiliki pointer ke data berikutnya dan data sebelumnya.

Terdapat 3 jenis LinkedList : Single LinkedList , Double LinkedList , Circular LinkedList

- **Single LinkedList**

Setiap node pada LinkedList mempunyai field yang berisi pointer ke node berikutnya dan juga memiliki field yang berisi data. Akhir LinkedList ditandai dengan node terakhir akan menunjuk ke null yang akan digunakan sebagai kondisi berhenti saat pembacaan LinkedList.

- **Double LinkedList**

LinkedList dimana setiap node memiliki 3 field yaitu 1 field pointer yang menunjuk ke pointer berikutnya (next) , 1 field pointer yang menunjuk ke pointer sebelumnya (prev) dan field yang berisi data dari Node tersebut. Pointer next dan prev menunjuk ke null.

- **Circular LinkedList**

Dalam Circular LinkedList ini masih dibagi lagi menjadi beberapa bagian diantaranya yaitu :

- o Single Circular LinkedList
LinkedList yang pointer next-nya menunjuk ke diri nya sendiri, jika terdiri dari beberapa node maka pointer terakhir nya akan menunjuk ke pointer terdepan nya.
- o Double Circular LinkedList
Double LinkedList yang pointer next nya dan prev nya menunjuk ke diri nya sendiri secara circular.

D. Implementasi penggunaan beberapa method dalam ArrayList

```
import java.util.ArrayList;

public class CthArrayList {

    public static void main(String args[]) {
        ArrayList<Integer> data = new ArrayList<>();
    }
}
```

```

        data.add(1);
        data.add(2);
        data.add(3);
        data.add(4);
        data.add(5);

        System.out.println(data.get(0));
        System.out.println(data.get(1));
        System.out.println(data.get(2));
        System.out.println(data.get(3));
        System.out.println(data.get(4));
    }
}

```

E. Membuat LinkedList Secara Manual

1. Membuat kelas Node

```

public class Node {

    int data;
    Node next;
    Node prev;
    public Node (int data){
        this.data = data;
    }
}

```

2. Membuat Kelas Link

```

public class Link {
    Node head;

    public void add(int data){
        if (head == null){
            head = new Node(data);
        }

        Node current = head;
        while (current.next != null){
            current = current.next;
        }
        current.next = new Node(data);
    }
}

```



```

public void showData(){
    if (head == null){
        System.out.println("Linklist is Empty");
        return;
    }
    Node current = head;
    while (current.next != null){
        current = current.next;
        int data = current.data;
        System.out.println(data);
    }
}

```

3. Membuat kelas Main

```

public class LinkedList {
    public static void main(String[] args) {
        Link myLink = new Link();
        myLink.add(6);
        myLink.add(8);
        myLink.add(4);
        myLink.add(5);
        myLink.showData();
    }
}

```

Tambahan Materi

Mari kita coba untuk membuat code versi bahasa pemrograman Python untuk menambah pengetahuan kita.

A. Arraylist

Berikut ini adalah contoh implementasi arraylist menggunakan list di python:

```

class ArrayList:
    def __init__(self):
        self.array = []
        self.length = 0

    def __len__(self):
        return self.length

    def __getitem__(self, index):
        if index >= self.length:
            raise IndexError('Index out of range')
        return self.array[index]

    def __setitem__(self, index, value):
        if index >= self.length:
            raise IndexError('Index out of range')
        self.array[index] = value

    def append(self, value):
        self.array.append(value)
        self.length += 1

    def insert(self, index, value):
        if index > self.length:
            raise IndexError('Index out of range')
        self.array.insert(index, value)
        self.length += 1

    def remove(self, value):
        self.array.remove(value)
        self.length -= 1

    def pop(self, index=-1):
        value = self.array.pop(index)
        self.length -= 1
        return value

    def index(self, value):
        return self.array.index(value)

    def clear(self):
        self.array.clear()
        self.length = 0

```

Pada contoh di atas, kita membuat kelas ArrayList yang memiliki metode-metode seperti append, insert, remove, dan pop yang serupa dengan metode-metode yang ada di ArrayList pada bahasa pemrograman lain. Kita juga menambahkan metode clear untuk menghapus semua elemen dari arraylist.

Metode `__len__`, `__getitem__`, dan `__setitem__` digunakan untuk mendukung penggunaan operator `len`, `[]`, dan `[] =` pada objek `ArrayList`.

Contoh penggunaan dari kelas `ArrayList` di atas:

```
arraylist = ArrayList()

arraylist.append(1)
arraylist.append(2)
arraylist.append(3)

print(len(arraylist))  # Output: 3

print(arraylist[0])  # Output: 1

arraylist[1] = 4
print(arraylist[1])  # Output: 4

arraylist.insert(2, 5)
print(arraylist[2])  # Output: 5

arraylist.remove(4)
print(len(arraylist))  # Output: 3

value = arraylist.pop()
print(value)  # Output: 3

print(arraylist.index(2))  # Output: 1

arraylist.clear()
print(len(arraylist))  # Output: 0
```

Dalam contoh di atas, kita membuat objek arraylist dari kelas ArrayList dan menambahkan beberapa elemen menggunakan metode append. Kemudian, kita mengakses elemen pada indeks tertentu menggunakan operator []. Setelah itu, kita memperbarui nilai pada indeks tertentu menggunakan operator []= dan menambahkan elemen baru menggunakan metode insert. Selanjutnya, kita menghapus elemen menggunakan metode remove dan pop, dan mencari indeks elemen tertentu menggunakan metode index.

B. Linkedlist

Berikut ini adalah contoh implementasi linkedlist menggunakan class di Python:

```
class Node:
    def __init__(self, data=None):
        self.data = data
        self.next = None

class LinkedList:
    def __init__(self):
        self.head = None

    def __len__(self):
        count = 0
        node = self.head
        while node:
            count += 1
            node = node.next
        return count

    def __getitem__(self, index):
        if index >= len(self):
            raise IndexError('Index out of range')
        node = self.head
        for i in range(index):
            node = node.next
        return node.data

    def __setitem__(self, index, value):
        if index >= len(self):
            raise IndexError('Index out of range')
        node = self.head
        for i in range(index):
            node = node.next
        node.data = value

    def append(self, value):
        new_node = Node(value)
        if self.head is None:
            self.head = new_node
        else:
            node = self.head
            while node.next:
                node = node.next
            node.next = new_node
```

```

def insert(self, index, value):
    new_node = Node(value)
    if index == 0:
        new_node.next = self.head
        self.head = new_node
    else:
        node = self.head
        for i in range(index - 1):
            node = node.next
            if node is None:
                raise IndexError('Index out of range')
        new_node.next = node.next
        node.next = new_node

def remove(self, value):
    if self.head is None:
        raise ValueError('Value not found')
    elif self.head.data == value:
        self.head = self.head.next
    else:
        node = self.head
        while node.next and node.next.data != value:
            node = node.next
        if node.next is None:
            raise ValueError('Value not found')
        else:
            node.next = node.next.next

def pop(self, index=-1):
    if self.head is None:
        raise IndexError('Index out of range')
    elif index == 0:
        value = self.head.data
        self.head = self.head.next
    else:
        node = self.head
        for i in range(index - 1):
            node = node.next
            if node is None:
                raise IndexError('Index out of range')
        if index == -1:
            index = len(self) - 1
        value = node.next.data
        node.next = node.next.next
    return value

```

```

def index(self, value):
    index = 0
    node = self.head
    while node and node.data != value:
        node = node.next
        index += 1
    if node is None:
        raise ValueError('Value not found')
    else:
        return index

def clear(self):
    self.head = None

```

Pada contoh di atas, kita membuat kelas Node yang merepresentasikan simpul dari linked list, dan kelas LinkedList yang memiliki metode-metode seperti append, insert, remove, dan pop yang serupa dengan metode-metode yang ada di LinkedList pada bahasa pemrograman lain

LATIHAN PRAKTIKUM

LATIHAN 1

Penggunaan interface List pada kelas LinkedList :

Buatlah dua objek List (ArrayList) yaitu objek Hewan dan DeleteHewan. Objek ini berisi hewan-hewan, buatlah sebagian ada yang sama. Lakukan penghapusan data yang terdapat pada objek Hewan yang sama dengan data warna yang terdapat pada objek DeleteHewan, selanjutnya tampilkan.

```

Hewan : [Sapi, Kelinci, Kambing, Unta, Domba]
Hewan yang dihapus:
[Kelinci, Kambing, Unta]

Output Hewan :
[Sapi, Domba]

```

TUGAS

KEGIATAN 1

Buatlah sebuah ArrayList of String dengan nama-nama Hewan, Lalu isilah ArrayList tersebut dengan ketentuan sebagai berikut ini :

Index 0 = Angsa
Index 1 = Bebek
Index 2 = Cicak
Index 3 = Domba
Index 4 = Elang
Index 5 = Gajah

1. Buatlah object kosong untuk menambahkan ketentuan diatas yang bertipe data String. Boleh menggunakan Library.
2. Tambahkan elemen lagi "Badak" dan "Bebek", hitung jumlah elemen "Bebek" dan tampilkan posisi index dari elemen "Bebek" pada object kosong yang telah dibuat.

```
Output :  
[Angsa, Bebek, Cicak, Domba, Elang, Gajah, Badak, Bebek]  
Bebek = 2  
Posisi Index : 1, 4
```

3. Hapus posisi "Bebek" yang pertama.

```
Output :  
[Angsa, Cicak, Domba, Elang, Gajah, Badak, Bebek]
```

4. Tampilkan elemen pada index ke-0 dan ke-2, selanjutnya hapus index ke-0.

```
Output :  
Index ke-0:Angsa  
Index ke-2 :  
Domba  
[Cicak, Domba, Elang, Gajah, Badak, Bebek]
```

5. Ubahlah index ke-0 dari "Cicak" Menjadi "Ular", selanjutnya tambahkan elemen baru pada index ke-2 dengan "Itik".

```
Output :  
[Ular, Domba, Elang, Gajah, Badak, Bebek]  
[Ular, Domba, Itik, Elang, Gajah, Badak, Bebek]
```

6. Hapus elemen diantara index ke-1 dan ke-5.


```
Output :  
[Ular, Domba, Badak, Bebek]
```

7. Tampilkan elemen pertama dan terakhir.

```
Output :  
Elemen Pertama : Ular  
Elemen Terakhir : Bebek
```

8. Tampilkan jumlah elemen pada ArrayList.
9. Carilah posisi index dari "Badak".
10. Sampaikan pemahamanmu mengenai ArrayList kepada asisten masing-masing

KEGIATAN 2

Buatlah sebuah program LinkedList secara manual **tidak diperkenankan menggunakan library** dengan ketentuan dibawah ini :

Ketika user menambahkan Node baru bernilai n pada LinkedList, maka program akan memeriksa apakah nilai dari n tersebut mempunyai nilai lebih besar daripada data yang lain. Sehingga program LinkedList yang dibuat dapat mengurutkan node dari yang terkecil hingga terbesar.

Contoh :

```
LinkedList list = new LinkedList() list.add(8);  
list.add(7);  
list.add(1);  
list.add(4);  
list.add(6);  
list.add(2);  
list.add(3); list.printList();
```

Output :



```
Output - Modul2ku (run) x  
run:  
before : 8 7 1 4 6 2 3  
after : 1 2 3 4 6 7 8  
BUILD SUCCESSFUL (total time: 0 seconds)
```

CATATAN

Kegiatan 1 : Gunakan Library untuk mengerjakan

Kegiatan 2 : Buat Linkedlist secara manual tidak diperkenankan menggunakan library

Aturan umum penulisan bahasa JAVA agar mudah di koreksi oleh asisten:

1. Untuk nama kelas,interface,enum, dan yang lainnya biasakan menggunakan gaya CamelCase (diawali dengan huruf besar pada tiap kata untuk mengganti spasi) seperti: Kursi , JalanRaya, ParkiranGedung, dan lain seterusnya.
 2. Untuk penulisan nama method, dan attribute diawali dengan huruf kecil di awal kata dan menggunakan huruf besar untuk kata setelahnya, seperti: getNamaJalan, namaJalan, harga, setNamaJalan, dan lain seterusnya.
 3. Jika menggunakan IDE IntelliJ jangan lupa untuk memformat penulisan kode agar terlihat rapi menggunakan menu code -> show reformat file dialog -> centang semua field dan klik ok. Silahkan dikerjakan tanpa **copy – paste**.
-

DETAIL PENILAIAN TUGAS

Kriteria	Nilai
Semua Ketentuan Pada Tugas Praktikum Terpenuhi Saat Demo dan Kerapihan Code Serta Tidak Ada Plagiasi	40
Mengerjakan Latihan Praktikum	40
Presensi Kehadiran	20