**LAPORAN PRAKTIKUM**

**ALGORITMA DAN PEMROGRAMAN**

**Pertemuan Ke – 13**



**DISUSUN OLEH :**

**HELDA LUDYA SAFITRI**

**175410186**

**TEKNIK INFORMATIKA**

**STMIK AKAKOM**

**YOGYAKARTA**

**2017**

**PERTEMUAN KE-13**

**ARRAYLIST**

1. **TUJUAN**
2. Mahasiswa dapat membuat sebuah arraylist.
3. Mahasiswa dapat memanipulasi.
4. Mahasiswa dapat menggunakan arraylist dalam loop.
5. **DASAR TEORI**

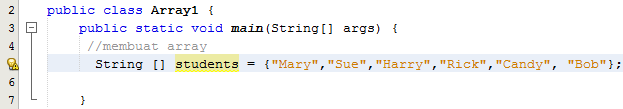
Berbeda dengan Array biasa, ArrayList memiliki sejumlah operasi yang lebih lengkap dan mudah digunakan dibandingkan dengan array biasa. ArrayList merupakan collection yang menjadi bagian dari Java Util. Seperti biasa ArrayList dapat menambah data baru secara dinamis tanpa harus menentukan ukurannya diawal.

Sedangkan Iterator digunakan untuk membuat element-element seperti collection. ListIterator adalah extend dari class Iterator, bisa memudahkan untuk mengambil element-element yang ada di collection dengan cara maju atau mundur.

1. **PEMBAHASAN PRAKTIK**

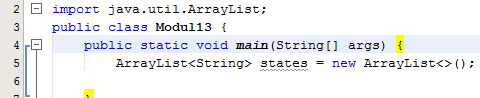
PRAKTIK 1 : Creating an ArrayLists

1. Create Array



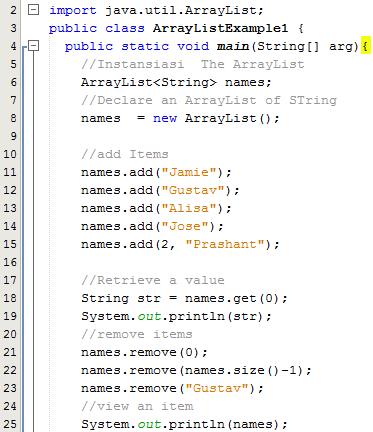
Pada baris ke 5 kita akan mendeklarasikan array dengan nama students dan tipe datanya adalah string dengan nilai seperti di atas yaitu tidak berupa angka.

1. Importing and Declaring Array List

****

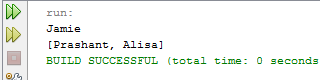
Baris ke 5 artinya mendeklarasikan ArrayList dengan nama variable states dan tipe data string.

1. Bekerja dengan ArrayList



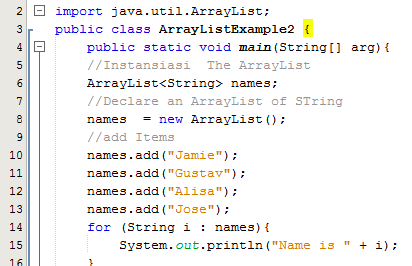
Pada baris ke-6 mendeklarasikan ArrayList dengan nama variable names dan tipe data string, kemudian pada baris 11-14 menambahkan data (add) pada variable names dan nilai data bertipe string sesuai yang sudah dideklarasikan. Selanjutnya pada baris ke-18 **String str = names.get(0);** artinya mendeklarasikan variable str dengan tipe data string yang bernilai sesuai dengan variable names index ke 0 pada baris sebelumnya dan variable str tersebut ditampilkan. Kemudian **names.remove** artinya menghapus nilai pada variable names sesuai indexnya atau nilai dari variable nya sendiri.

Output :



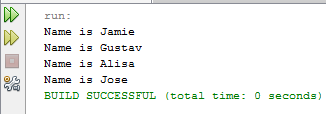
Praktik 2. Traversing an ArrayList

1. Using the for-each loop

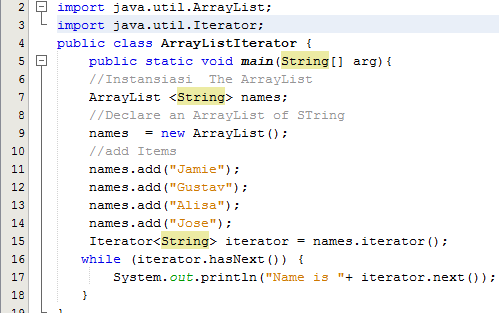


Penjelasan program diatas sama dengan praktik 1c yaitu ada penambahan data (add) pada variable names yang bertipe data string, namun menggunakan **for (String i : names)** yang artinya variable names akan dideklarasikan dengan variable i dalam tipe data string, kemudian ditampilkan variable i yang berisi semua nilai dari variable names.

Output :

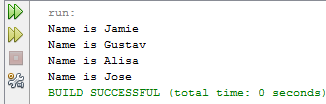
****

1. Introducing Iterator

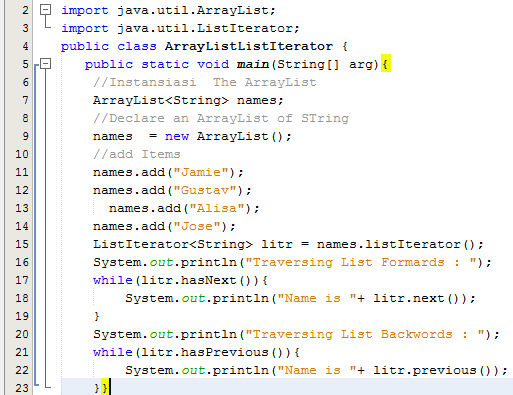


Program tersebut setelah ditambahkan datanya yang dideklarasikan adalah Iterator dengan tipe data string, kemudian menggunakan perulangan while untuk menampilkan variable iterator yang ditambahkan.

Output :

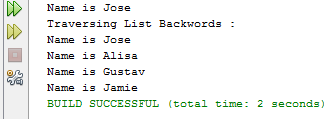


1. Intruducing ListIterator



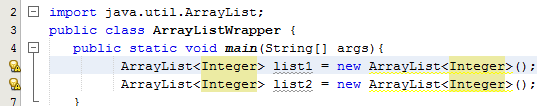
Program tersebut sama dengan praktik sebelumnya menggunakan method add, namun pendeklarasian menggunkan ListIterator kemudian dilanjutkan dengan perulangan while yang akan ditampilkan dari method next yaitu Jose, dan dari method previous akan menampilkan Jose, Alise, Gustav, dan Jamie, karena method previous digunakan untuk menampilkan data sebelumnya.

Output :



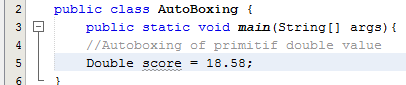
Praktik 3. Using Java Wrapper Class

1. ArrayList and primitive



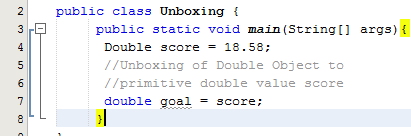
Program diatas merupakan pendeklarasian ArrayList dengan nama variable list 1 dan list 2 dan tipe data integer, namun tipe data selain string dalam ArrayList harus ditulis lengkap (primitive) agar tidak terjadi error.

1. Introducing AutoBoxing



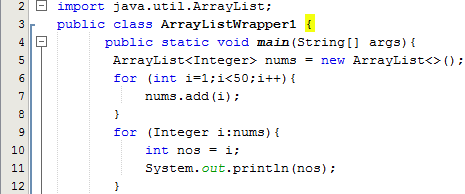
AutoBoxing tersebut menggunakan tipe data primitif yaitu Double dan nama variable score bernilai 18.58.

1. Introducing Unboxing



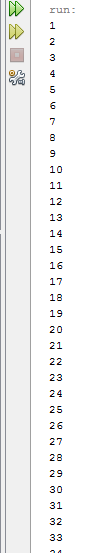
Unboxing tersebut melakukan converse dari tipe data primitive Double dengan nama variable score menjadi tipe data double dengan nama variable goal dan bernilai score atau 18.58.

1. ArrayList and Wrapper Class

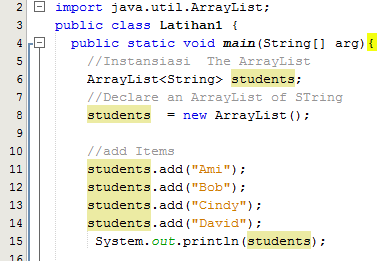


Program tersebut mendeklarasikan ArrayList dengan nama variable nums dan tipe data Integer. Kemudian menggunakan perulangan for untuk tipe data integer dengan nama variable i, dimulai dari i=1 kemudian i+1 namun kurang dari 50, lalu nilai dari variable nums ditambahkan menggunakan method add, selajutnya untk perulangan for variable yang bernama i dengan tipe data Integer nilainya sama dengan variable nums dan diberi nama nos, nos tersebut merupakan nama variable dari nilai i, kemudian nos ditampilkan.

Output :

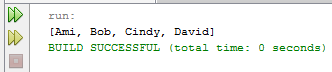
****

1. **LATIHAN**
2. Latihan 1

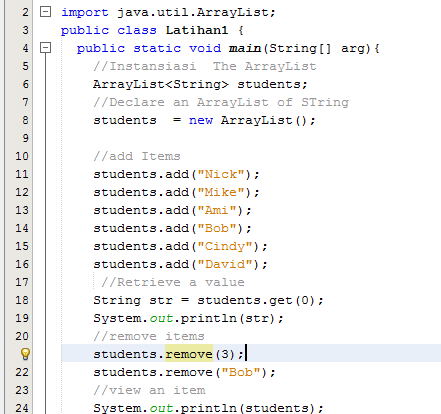


Program diatas mendeklarasikan ArayList dengan nama variable students dan tipe data string, kemudian menambahkan data (add) dengan nilai yang bertipe string yaitu Ami, Bob, Cindy, David, setelah itu nilai tersebut ditampilkan hanya dengan memanggil nama variablenya (students) .

Output :

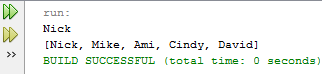


1. Latihan 2

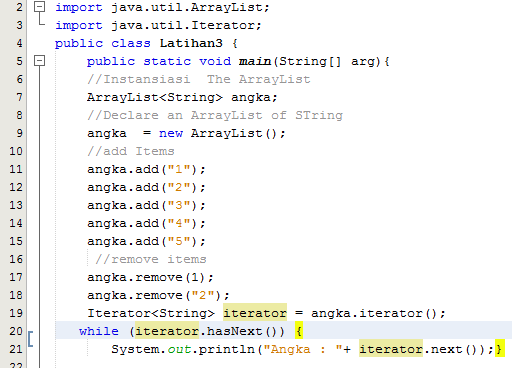


Program diatas sama dengan latihan 2, namun setelah data ditambahkan (add), ada salah satu data yang dihapus dengan menggunakan method remove yaitu pada baris ke 21 dan 22, maka secara otomatis data yang sudah dihapus tidak ditampilkan pada variable students

Output :

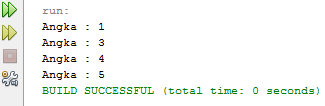


1. Latihan 3

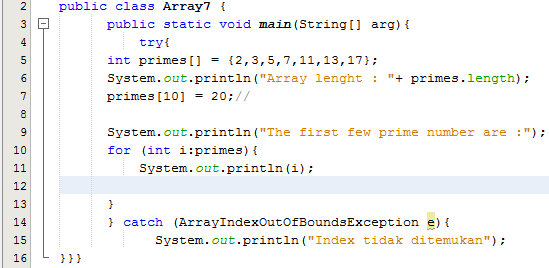


Program diatas mendeklarasikan ArrayList dengan nama variable angka dan tipe data String, kemudian data ditambahkan dengan nilai 1-5, namun data atau nilai pada index ke-1 yaitu angka 2 dihapus menggunakan method remove. Kemudian mendeklarasikan Iterator dengan nama variable iterator dan tipe data string, dan menggunakan perulangan while, setelah itu ditampilkan tanpa angka 2 yang merupakan bilangan genap.

Output :



1. **TUGAS**
2. Contoh Program



1. Tipe data Error

*Syntax* merupakan suatu aturan penulisan yang sudah ditetapkan pada struktur elemen - elemen dalam bahasa pemrograman.  
  
Jadi *syntax error* ini adalah kesalahan dalam *coding* karena aturan penulisan yang tidak sesuai atau kesalahan pada konstruksi kode, misalnya :

* Salah dalam menuliskan *keyword* Java
* Tidak menggunakan tanda kurung kurawal untuk pernyataan
* Tidak menggunakan tanda atau karakter yang sesuai, misalnya tidak menggunakan tanda " ", untuk tipe data String.

Syntax error ini mudah ditelusuri atau ditemukan karena *compiler* akan memberi tahu kita di mana letak kesalahan dalam penulisan kode program

*Runtime error* merupakan salah satu jenis *error* yang terjadi karena program secara tiba-tiba dihentikan ketika program tersebut sedang berjalan. *Runtime error* ini dapat terjadi bila program tidak bisa menangani operasi yang tidak mungkin untuk dilakukan.

Logic error terjadi bila program berjalan dan tidak memberikan hasil seperti yang diinginkan.

1. **KESIMPULAN**

Berdasarkan pembahasa praktikum dapat disimpulkan bahwa ArrayList memiliki sejumlah operasi yang lebih lengkap dan mudah digunakan dibandingkan dengan array biasa. ArrayList merupakan collection yang menjadi bagian dari Java Util

\

1. **LISTING**

Terlampir