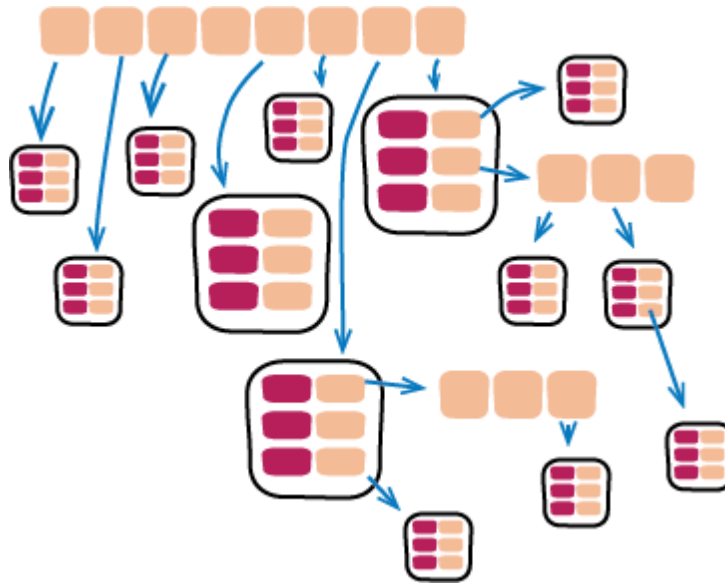


Praktikum Pemrograman I

List (Operasi Add Head & Tail)



<https://martinfowler.com/>

Oleh :

Ade Sukendar [ade.sukendar@unpas.ac.id]



Laboratorium Prodi Teknik Informatika
Universitas Pasundan
2025

Pre-Latihan

- Sebelum dimulai praktikum berdoalah terlebih dahulu supaya memudahkan dalam memahami modul praktikum yang dikerjakan.
- Siapkan projek program di folder yang mudah diingat.
- Setiap latihan yang akan dikerjakan disarankan untuk membuat kelas yang baru dengan *package* yang berbeda.
- Kerjakan latihan dengan **tenang, sungguh-sungguh, tidak mencontek** dan **tidak membuat kegaduhan** yang mengganggu praktikan yang lain.
- Jika ada soal latihan praktikum yang tidak dimengerti tanyakan ke akang/teteh asisten.
- Selamat mengerjakan dan sukses.

List adalah “a finite sequence of zero or more elements” (Alfred & Jeffrey) atau daftar urutan elemen yang terdiri dari nol atau banyak elemen didalamnya. Elemen *list* bertipe *T* maka disebut dengan *list* bertipe *T*, contohnya List bertipe integer, list bertipe bilangan real, list bertipe struktus dan list bertipe lainnya. Penulisan elemen list dipisahkan dengan koma dan berada di dalam tanda kurung seperti di bawah ini:

$$(a_1, a_2, a_3, \dots, a_n)$$

Model data list dapat diimplementasikan dengan tipe data array maupun dengan *linked-list*. Contoh List bilangan prima kurang dari 20 yaitu (2, 3, 5, 7, 11, 13, 17, 19) atau *List* jumlah hari dalam setiap bulan, bukan tahun kabisat yaitu (31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31).

List mempunyai operasi untuk menambah dan menghapus elemen. Operasi tersebut dapat dilakukan di posisi awal list, di akhir list maupun di tengah list. Pembahasan praktikum kali ini mengenai operasi menambahkan elemen list.

Latihan 1

Latihan ini akan memberikan implementasi pembuatan elemen list. Elemen list direpresentasikan dengan Node. Sebuah node terdiri dari atribut data/nilai dan atribut *next*. Atribut *next* akan menunjuk ke node yang lain.

Pseudocode	Bahasa Pemrograman
<pre> class Node { data: tipe data (T) next: Node } </pre>	<pre> public class Node { private int data; private Node next; /** Inisialisasi atribut node */ public Node(int data) { this.data = data; } /** Setter & Getter */ } </pre>

- Contoh setter: `public void setName(String nama) { this.nama = nama; }`
- Contoh getter: `public String getName() { return nama; }`

Latihan 2

Latihan ini akan memberikan implementasi operasi penambahan elemen list di akhir/*tail* dengan notasi algoritma. Operasi ini direpresentasikan dengan fungsi **addTail** dengan parameter data yaitu node yang akan ditambahkan ke List.

- Buatlah kelas **StrukturList** kemudian tambahkan atribut HEAD dengan tipe data Node
- Tambahkan fungsi dibawah ini di kelas **StrukturList**. Fungsi addTail di bawah dikonversi ke dalam bahasa pemrograman

Algoritma addTail	Program addTail
<pre>procedure addTail(data: integer) deklarasi posNode, curNode: Node {current node} deskripsi newNode ← new Node(data) IF (isEmpty()) THEN HEAD ← newNode ELSE curNode ← HEAD WHILE (curNode <> null) DO posNode ← curNode curNode ← curNode.next ENDWHILE posNode.next ← newNode ENDIF</pre>	<pre>public void addTail(int data) { Node posNode=null, curNode=null; Node newNode = new Node(data); if (isEmpty()) { HEAD = newNode; } else { curNode = HEAD; while (curNode != null) { posNode = curNode; curNode = curNode.getNext(); } posNode.setNext(newNode); } }</pre>

- *function isEmpty()* harus diimplementasikan dengan melakukan apakah list kosong atau tidak (HEAD != null)!

Latihan 3

Latihan ini akan memberikan implementasi untuk menampilkan elemen list. Elemen list yang ditampilkan ke layar diawali dari nilai HEAD.

Algoritma	Bahasa Pemrograman
<pre>procedure displayElement() deklarasi curNode: Node deskripsi curNode ← HEAD; WHILE (curNode <> null) DO print(curNode.data) curNode ← curNode.next ENDWHILE</pre>	<pre>public void displayElement() { Node curNode = HEAD; while (curNode != null) { System.out.print(curNode.getData() + " "); curNode = curNode.getNext(); } }</pre>

Latihan 4

Latihan ini akan memberikan penggunaan operasi penambahan elemen di akhir list dan kemudian menampilkan setiap elemen yang terdapat di list. Buatlah kelas **ListTest** berikut fungsi main untuk mengeksekusi program. Konversikan urutan instruksi berikut di bawah ini ke fungsi tersebut!

Urutan Instruksi	Program
<ol style="list-style-type: none">1. Create list dengan keyword new2. Tambah elemen 3 di akhir list3. Tambah elemen 4 di akhir list4. Tambah elemen 5 di akhir list5. Tampilkan elemen list	<pre>public class ListMain { public static void main(String[] args) { StrukturList list = new StrukturList(); list.addTail(3); list.addTail(4); list.addTail(5); System.out.println("Elemen: "); list.display(); } }</pre>

Outputnya adalah: **Elemen: 3 4 5**

Tes-1

Lakukan seperti diatas dengan output elemen list seperti berikut:

- a. 3 2 1
- b. 1 4 5 7

Latihan 5

Latihan ini akan memberikan implementasi operasi penambahan elemen list di awal/head. Operasi ini direpresentasikan dengan *procedure* **addHead** dengan parameter data yang akan ditambahkan. Tambahkan atribut **HEAD** dengan tipe data Node kelas **StrukturList**.

Algoritma addHead	Bahasa Pemrograman
<pre>procedure addHead(data: integer) deskripsi newNode ← new Node(data); IF(HEAD = null) THEN HEAD ← newNode ELSE newNode.next ← HEAD HEAD ← newNode ENDIF</pre>	<pre>public void addHead(int data) { Node newNode = new Node(data); if (isEmpty()) { HEAD = newNode; } else { newNode.setNext(HEAD); HEAD = newNode; } }</pre>

Tes-2

Latihan ini akan memberikan penggunaan operasi penambahan elemen di awal list dan kemudian menampilkan setiap elemen yang terdapat di list. Konversikan urutan instruksi berikut di bawah ini ke dalam Bahasa pemrograman!

Urutan Instruksi	Output
<ol style="list-style-type: none">1. Create list dengan keyword new2. Tambah elemen 5 di awal list3. Tambah elemen 4 di awal list4. Tambah elemen 3 di awal list5. Tampilkan elemen list	3 4 5

Tes-3

Lakukan seperti diatas dengan output elemen list seperti berikut:

- a. 3 2 1
- b. 1 4 5 7