

## Laporan Praktikum

### Tugas Modul 6a

Nama : Muhammad Hadist Rifannan

Kelas : TI 1D

NPM : 240202107

## Manajemen Data Buku

Judul Buku:

Contoh: Belajar PHP

Harga Buku:

Contoh: 50000

Tambah di Depan

Tambah di Belakang

Hapus dari Depan

Hapus dari Belakang

### Daftar Buku:

No	Judul Buku	Harga Buku
1	Belajar PHP	Rp 10.000
2	Belajar Javascript	Rp 12.000

Berikut adalah tampilan awal dari website tersebut. Pengguna dapat menambahkan data buku di depan atau dibelakang lalu juga dapat menghapus data buku dari depan atau dari belakang.

Dengan menggunakan LinkedList kita bisa membuat sistem seperti itu.

Disini terlihat data default yang sudah ada yaitu :

1. Buku Belajar PHP , harga : Rp 10.000
2. Buku Belajar Javascript , harga Rp 12.000

Jika user menambahkan buku dari belakang missal menambahkan buku berjudul Belajar C++

Dengan harga 15000 , maka daftar buku akan bertambah ke bawah.seperti gambar dibawah ini.

#### Daftar Buku:

No	Judul Buku	Harga Buku
1	Belajar PHP	Rp 10.000
2	Belajar Javascript	Rp 12.000
3	Belajar C++	Rp 15.000

Ketika user menambah data buku dari depan missalnya Buku Belajar C# dengan harga Rp 20.000 , maka daftar buku akan menambahkan buku terbaru di awal list .

#### Daftar Buku:

No	Judul Buku	Harga Buku
1	Belajar C#	Rp 20.000
2	Belajar PHP	Rp 10.000
3	Belajar Javascript	Rp 12.000
4	Belajar C++	Rp 15.000

Jika user menghapus dari belakang buku yang sudah ada didalam list maka buku yang terhapus adalah Belajar C++

#### Daftar Buku:

No	Judul Buku	Harga Buku
1	Belajar C#	Rp 20.000
2	Belajar PHP	Rp 10.000
3	Belajar Javascript	Rp 12.000

Jika user menghapus dari depan buku yang sudah ada didalam list maka buku yang terhapus adalah Belajar C#.

### Daftar Buku:

No	Judul Buku	Harga Buku
1	Belajar PHP	Rp 10.000
2	Belajar Javascript	Rp 12.000

### PENJELASAN SOURCE CODE :

```
1 class Buku {
2     public $judul;
3     public $harga;
4
5     public function __construct($h, $j) {
6         $this->harga = $h;
7         $this->judul = $j;
8     }
9 }
```

Berikut merupakan class object yang dibuat untuk memberikan komponen yang dibutuhkan sebuah buku dalam setiap node.

Class ini berisi konstruktor class yang dimana akan mendefinisikan value lalu meng assign variable didalam kelas dengan parameter.

```
1 class BukuNode {
2     public $val;
3     public $next;
4
5     public function __construct($v, $n = null) {
6         $this->val = $v;
7         $this->next = $n;
8     }
9 }
```

Berikut merupakan class dari tiap node yang dihubungkan. Node ini mempunyai komponen value dari class Buku dan mempunyai satu pointer next untuk menghubungkan ke node selanjutnya.

## Fungsi Fungsi yang digunakan :

```
1 function AddBookFront(&$head, $newNode) {  
2     $newNode->next = $head;  
3     $head = $newNode;  
4 }
```

### Fungsi AddBookFront :

Digunakan untuk menambah buku dari depan atau dari node pertama.dengan cara membuat node baru terlebih dahulu dan pointer selanjutnya akan diganti dengan head.Lalu head akan diganti dengan

node terbaru yang telah dibuat.

```
1 function AddBookLast($head, $newNode) {  
2     $current = $head;  
3     while ($current->next != null) {  
4         $current = $current->next;  
5     }  
6     $current->next = $newNode;  
7 }
```

### Fungsi AddBookLast :

Digunakan untuk menambah buku dari belakang linkedlist.caranya adalah dengan melakukan operasi traversal yaitu akses ke setiap node selanjutnya sampai menuju node terakhir.lalu jika node selanjutnya berisi **NULL** maka node selanjutnya akan ditambahkan node baru.

```
1 function DeleteBookFront(&$head) {  
2     if ($head == null) return;  
3     $temp = $head;  
4     $head = $head->next;  
5     unset($temp);  
6 }
```

### Fungsi DeleteBookFront:

Digunakan untuk menghapus data buku dari depan.caranya adalah dengan mengecek terlebih dahulu apakah node paling depan **NULL** atau tidak.

Jika tidak **NULL** maka buat variable temp untuk menampung head sementara lalu pindahkan head ke node selanjutnya dan hapus memori

dari temp.

```
1 function DeleteBookLast($head) {
2     if ($head == null || $head->next == null) {
3         $head = null;
4         return;
5     }
6     $current = $head;
7     while ($current->next->next != null) {
8         $current = $current->next;
9     }
10    unset($current->next);
11    $current->next = null;
12 }
```

### Fungsi DeleteBookLast :

Digunakan untuk menghapus data buku terakhir dalam list.

Dengan cara mengecek terlebih dahulu apakah pointer head atau head selanjutnya **NULL** atau tidak. Setelah itu lakukan traversal dengan mengecek 2 pointer selanjutnya setelah pointer head. dan meng unset pointer head selanjutnya ke **NULL**.

```
1 function PrintAllBook($head) {
2     $output = "";
3     $t = $head;
4     $i = 1;
5     while ($t != null) {
6         $output .= "<tr><td>$i</td><td>{$t->val->judul}</td><td>Rp " . number_format($t->val->harga, 0, ',', '.') . "</td></tr>";
7         $t = $t->next;
8         $i++;
9     }
10    return $output;
11 }
```

### Fungsi PrintAllBook :

Untuk mencetak seluruh node yang ada, dengan melakukan traversal ke seluruh node.

```
1 $head = unserialize($_SESSION['head']);
2
3 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
4     $aksi = $_POST['aksi'] ?? '';
5     $judul = $_POST['judul'] ?? '';
6     $harga = $_POST['harga'] ?? '';
7
8     if ($aksi === 'tambah_depan' && $judul && $harga) {
9         AddBookFront($head, new BukuNode(new Buku($harga, $judul)));
10    } elseif ($aksi === 'tambah_belakang' && $judul && $harga) {
11        AddBookLast($head, new BukuNode(new Buku($harga, $judul)));
12    } elseif ($aksi === 'hapus_depan') {
13        DeleteBookFront($head);
14    } elseif ($aksi === 'hapus_belakang') {
15        DeleteBookLast($head);
16    }
17
18    $_SESSION['head'] = serialize($head);
19 }
```

Kode diatas adalah untuk memilih aksi yang dilakukan user.