

## Tutorial 5 Thymeleaf – Presentation Layer Design

### Ringkasan dari materi

- Thymeleaf merupakan Java Library berupa XML/XHTML/HTML5 template engine. Thymeleaf digunakan untuk melakukan transformasi dari file XML/XHTML/HTMLtemplate menjadi sebuah file XML/XHTML/HTML yang utuh untuk ditampilkan. Pada tutorial kali ini kita menggunakan HTML5 template engine Thymeleaf. Thymeleaf memungkinkan kita untuk membuat template reusable.
- Kita dapat melakukan iterasi pada view dengan bantuan Thymeleaf dengan menggunakan atribut `th:each`. Atribut tersebut berfungsi untuk melakukan iterasi dari kumpulan data pada suatu objek yang berada di salah satu atribut yang dimiliki suatu kelas Model.
- Kita juga dapat melakukan conditional expression dengan menggunakan `th:if=${...}` dan `th:unless=${...}`. Hal ini dapat memudahkan kita dalam menampilkan suatu tampilan dengan kondisi tertentu.
- Thymeleaf juga memfasilitasi kita untuk menggunakan static file seperti css, javascript, image, dsb. Static file akan disimpan dan diakses di folder static dalam resources.
- Thymeleaf memungkinkan kita untuk reuse source code pada view dengan menggunakan fragment. Fragment adalah bagian-bagian kecil dari view dan bagian tersebut dapat digunakan oleh view lainnya. Pada tutorial ini kita reuse header footer.
- Thymeleaf juga bisa membantu kita untuk membuat handler error tertentu sehingga lebih mudah dimengerti.

### Penggunaan For Looping

**Pertanyaan 1** Apakah value yang dihasilkan dari `${iterationStatus.odd}`?

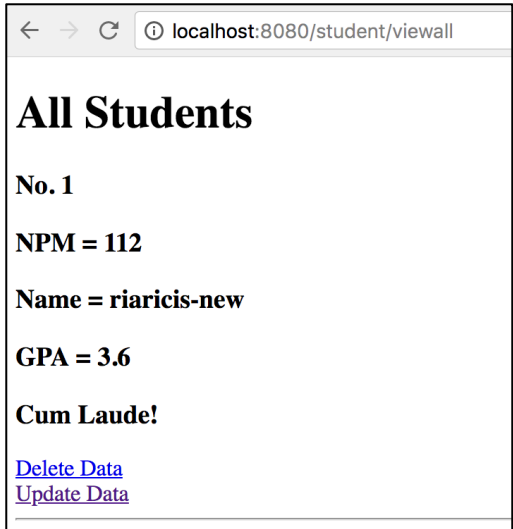
→ Value dari `${iterationStatus.odd}` adalah nilai ganjil tapi tampilan viewAll sama seperti sebelumnya. Hal ini dikarenakan karena value dari `${iterationStatus.odd}` tidak ditampilkan dan tidak digunakan juga untuk menampilkan data dengan counter ganjil, tapi hanya digunakan sebagai value dari atribut class dari `<tr>`.

Nama : Winda Dumaria Simanjuntak

NPM : 1706107075

### Penggunaan Conditional Expression pada Thymeleaf

Berikut hasil dari penggunaan conditional expression pada GPA seperti instruksi pada tutorial.



**All Students**

**No. 1**

**NPM = 112**

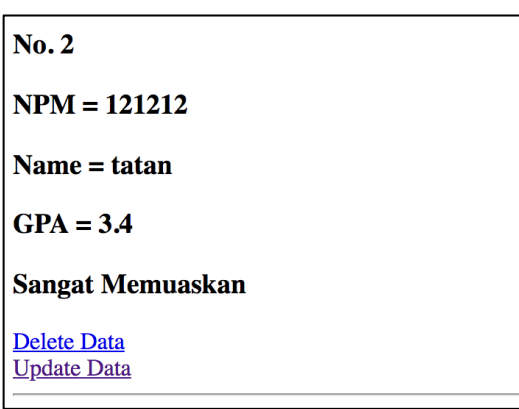
**Name = riaricis-new**

**GPA = 3.6**

**Cum Laude!**

[Delete Data](#)

[Update Data](#)



**No. 2**

**NPM = 121212**

**Name = tatan**

**GPA = 3.4**

**Sangat Memuaskan**

[Delete Data](#)

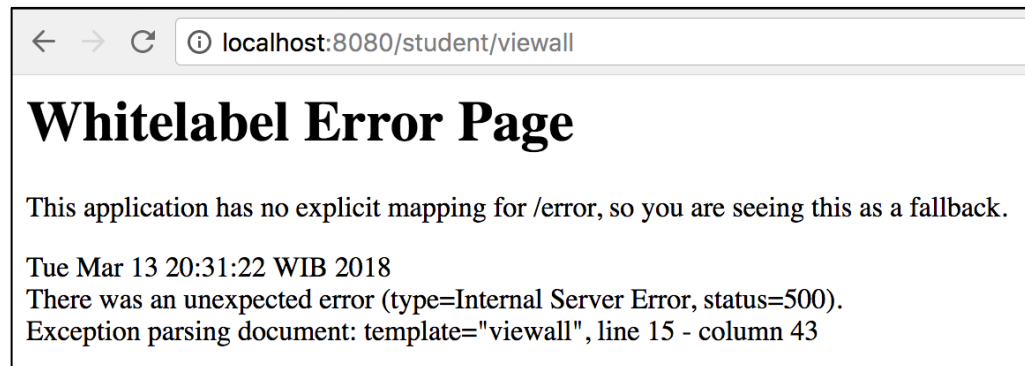
[Update Data](#)

**Pertanyaan 2:** Mengapa condition di dalam th:unless sama dengan di dalam th:if? Jelaskan jawaban Anda.

→ karna untuk bisa menjalankan fungsi dalam th:unless maka condition di th:unless harus sama seperti di th:if. Ketika condition th:if tidak dipenuhi maka akan dicari th:unless yang mempunyai condition yg sama dengan th:if.

**Pertanyaan 3:** Coba ubah condition pada th:unless=\${student.gpa>=3.49} menjadi th:unless=\${student.gpa<=3.48}, lalu run kembali aplikasi Anda dan mengakses halaman yang sama. Apakah terjadi error? Jika iya, kenapa? Jelaskan alasan Anda. Jika tidak error, apakah hasilnya sesuai?

→ Terjadi error karena ketika condition th:if tidak dipenuhi maka akan dicari th:unless yang mempunyai condition yg sama dengan th:if. Akan tetapi tidak ditemukan karna condition pada th:unless berbeda maka fungsi dalam th:unless tersebut tidak bisa dijalankan. Berikut hasil screenshot ketika condition th:unless diubah.



**Whitelabel Error Page**

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Tue Mar 13 20:31:22 WIB 2018

There was an unexpected error (type=Internal Server Error, status=500).

Exception parsing document: template="viewall", line 15 - column 43

**Pertanyaan 4:** Tuliskan cara lain Anda dengan penulisan conditional expression yang berbeda untuk mendapatkan hasil yang sama seperti gambar di atas. Hint: Gunakan ternary operator.

→ Berikut alternative yang dapat digunakan untuk conditional expression:

```
<h3 th:text="${student.getGpa()}>=3.49} ? 'Cum Laude!' : 'Sangat Memuaskan'"></h3>
```

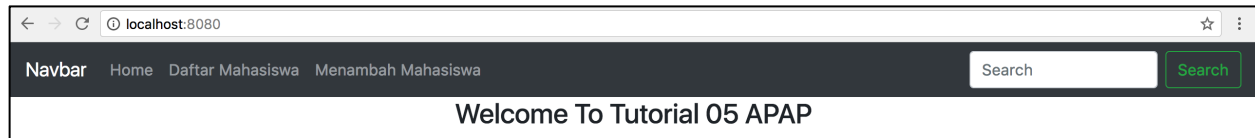
Nama : Winda Dumaria Simanjuntak  
NPM : 1706107075

### Static File

Static file yang digunakan pada tutorial ini adalah css bootstrap. Langkah-langkah yang perlu dilakukan adalah:

1. Static file tsb harus dimasukkan terlebih dulu di dalam folder static pada resource.
2. Panggil pada view menggunakan syntax `<link rel="stylesheet" href="/css/bootstrap.min.css"/>` .

Berikut hasil dari implementasi css bootstrap pada index.html sesuai dengan instruksi pada tutorial:

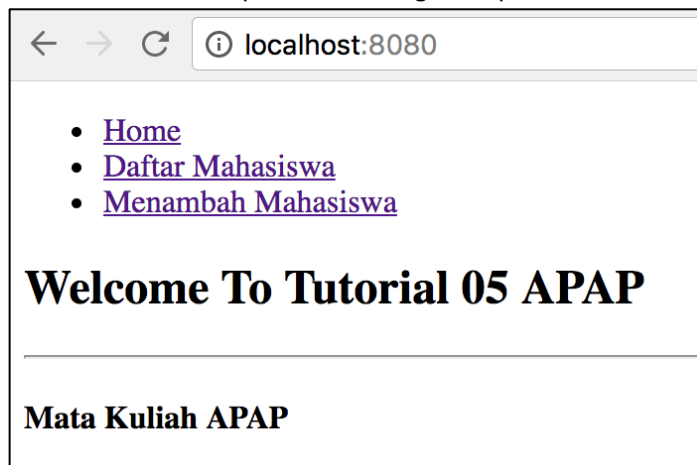


### Fragment

Penggunaan fragment pada tutorial ini diterapkan pada header dan footer. Langkah yang dilakukan adalah:

1. Buat view baru yaitu fragment.html berisi potongan source code yang akan digunakan oleh view lainnya seperti instruksi pada tutorial. Yang menandai potongan tersebut fragment adalah `th:fragment`
2. Panggil fragment yang sudah dibuat sebelumnya pada view yang diinginkan dengan menggunakan `th:replace`. Contohnya: `th:replace="fragment/fragment :: header"`

Berikut hasil dari implementasi fragment pada view index.



### **Pertanyaan 5:**

Apa yang dimaksud dengan `th:replace="fragments/fragment :: header"` dan `th:replace="fragments/fragment :: footer"` pada file index.html yang Anda buat?

→ potongan tersebut digunakan untuk memanggil fragment header pada fragment.html pada folder fragment untuk ditampilkan pada view yang memanggil itu, begitu juga dengan footer potongan tersebut digunakan untuk memanggil fragment footer.

Nama : Winda Dumaria Simanjuntak

NPM : 1706107075

### **Error 404 Not Found Handler**

Thymeleaf dapat memfasilitasi kita untuk membuat view untuk error tertentu seperti error 404 pada tutorial. Berikut contoh implementasi dari pembuatan handler error dengan Thymeleaf.



**Pertanyaan 6:** Apa handler dengan metode ini juga dapat berlaku bagi error lain seperti error 500: internal server error?

→ Tidak. Untuk bisa handle error 500 maka perlu membuat view lain dengan nama 500.html sehingga view tsb akan dipanggil ketika error 500 muncul, sama seperti error 404 karna sudah ada view handlernya maka view tersebut yang ditampilkan jika belum ada view nya maka error yang muncul adalah seperti berikut.



## Latihan

1. Untuk membuat tampilan data lebih rapi kita bisa menggunakan DataTables. Langkah yang perlu dilakukan adalah:

- a. Masukkan jquery, css, dan javascript pada view untuk menampilkan DataTables seperti berikut.

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
<link rel="stylesheet" type="text/css"
href="https://cdn.datatables.net/1.10.13/css/jquery.dataTables.min.css"/>
<script type="text/javascript" charset="utf8"
src="https://cdn.datatables.net/1.10.13/js/jquery.dataTables.min.js"></script>
```

- b. Masukkan potongan kode berikut untuk panggil DataTables tersebut.

```
<script>
$(document).ready(function(){
    $('#table-id').dataTable();
});
</script>
```

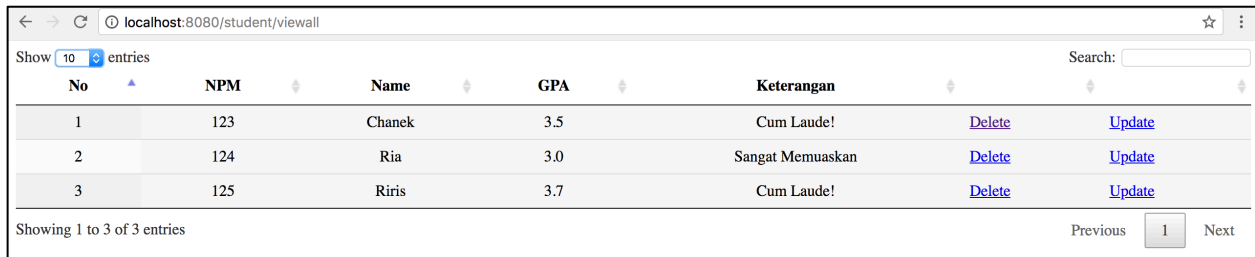
- c. Ubah cara menampilkan data mahasiswa menjadi seperti berikut.

```
<table id="table-id" class="display">
  <thead>
    <tr>
      <th>No</th>
      <th>NPM</th>
      <th>Name</th>
      <th>GPA</th>
      <th>Keterangan</th>
      <th></th>
      <th></th>
    </tr>
  </thead>
  <tbody>
    <tr th:each="student, iterationStatus: ${students}"
    th:class="${iterationStatus.odd} ? 'odd'">
      <td th:text="${iterationStatus.count}" align="center">No. 1</td>
      <td th:text="${student.getNpm()}" align="center"></td>
      <td th:text="${student.getName()}" align="center"></td>
      <td th:text="${student.getGpa()}" align="center"></td>
      <td th:text="${student.getGpa()}>=3.49 ? 'Cum Laude!' : 'Sangat
      Memuaskan'" align="center"></td>
      <td><a th:href="'/student/delete/' + ${student.getNpm()}">
      Delete</a></td>
      <td><a th:href="'/student/update/' + ${student.getNpm()}">
      Update</a></td>
    </tr>
  </tbody>
</table>
```

Nama : Winda Dumaria Simanjuntak

NPM : 1706107075

Berikut hasil dari langkah-langkah untuk mengimplementasikan DataTables seperti di atas.



No	NPM	Name	GPA	Keterangan		
1	123	Chanek	3.5	Cum Laude!	Delete	Update
2	124	Ria	3.0	Sangat Memuaskan	Delete	Update
3	125	Riris	3.7	Cum Laude!	Delete	Update

2. Untuk membuat fragment header dinamis sesuai instruksi pada tutorial berikut langkah-langkah yang dapat dilakukan:

- Buat view baru dengan nama header.html yang berisi dengan fragment untuk menampilkan title sesuai dengan page yang ditampilkan. Kita juga bisa membuat fragment untuk navbar yang sudah diimplementasi pada bagian sebelumnya seperti berikut ini.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <title th:fragment="header(title)" th:text="${title}">Title</title>
  </head>
  <body>
    <div th:fragment="nav">
      <link rel="stylesheet" href="/css/bootstrap.min.css"/>
      <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <a class="navbar-brand" href="#">Navbar</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse"
          data-target="#navbarSupportedContent"
          aria-controls="navbarSupportedContent" aria-expanded="false"
          aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
          <ul class="navbar-nav mr-auto">
            <li class="nav-item">
              <a class="nav-link" href="/">Home</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="/student/viewall">Daftar
Mahasiswa</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="/student/add">Menambah
Mahasiswa</a>
            </li>
          </ul>
          <form class="form-inline my-2 my-lg-0">
            <input class="form-control mr-sm-2" type="text"
placeholder="Search" aria-label="Search" />
            <button class="btn btn-outline-success my-2 my-sm-0"
type="submit">Search</button>
          </form>
        </div>
      </nav>
    </div>
  </body>
</html>
```

Nama : Winda Dumaria Simanjuntak

NPM : 1706107075

- b. Panggil fragment tersebut pada setiap view dengan menambahkan kode seperti berikut agar title yang ditambahkan sesuai dengan page yang ditampilkan.

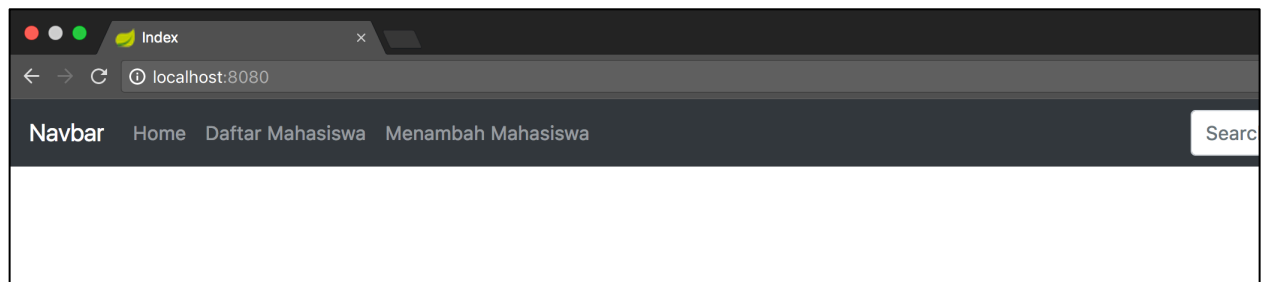
```
<head>
  <title th:replace="fragments/header ::header(title=${title})"></title>
  .....
</head>
<body>
  <div th:replace="fragments/header :: nav"></div>
  .....
</body>
```

- c. Jika dilihat pada potongan kode yang ditambahkan di atas perlu atribut model yang perlu dikirimkan dari controller. Dengan demikian kita perlu menambahkan atribut title seperti berikut.

```
model.addAttribute("title", "Delete");
```

Berikut hasil implementasi dari fragment header dinamis berdasarkan langkah-langkah di atas.

index.html



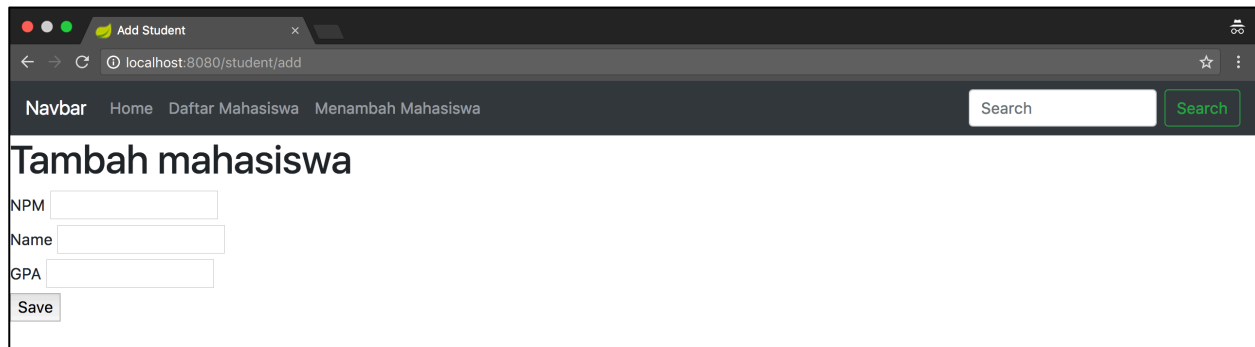
viewall.html

No	NPM	Name	GPA	Keterangan
1	123	Chanek	3.5	Cum Laude!
2	124	Ria	3.0	Sangat Memuaskan
3	125	Riris	3.7	Cum Laude!

Nama : Winda Dumaria Simanjuntak

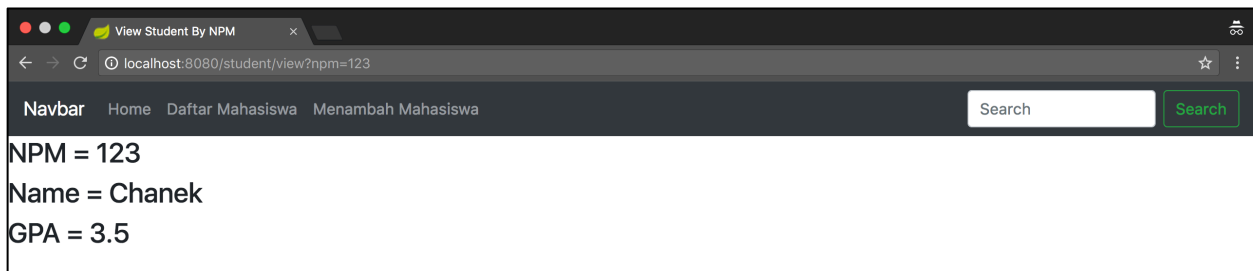
NPM : 1706107075

form-add.html

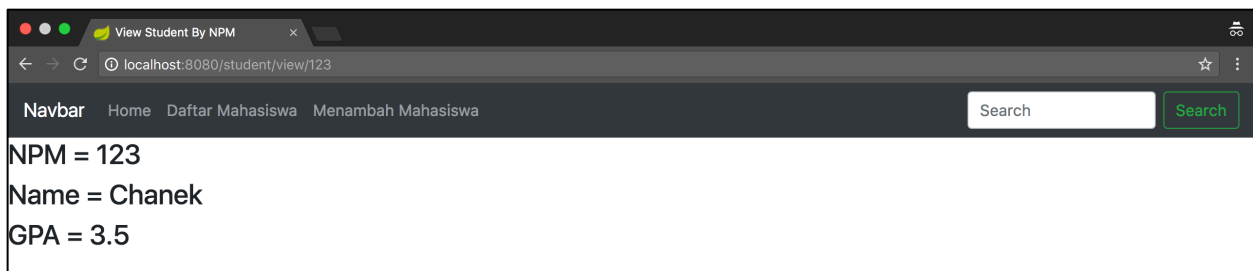


The screenshot shows a web browser window titled 'Add Student' with the URL 'localhost:8080/student/add'. The page has a dark navbar with links 'Home', 'Daftar Mahasiswa', and 'Menambah Mahasiswa'. A search bar is in the top right. The main content area is titled 'Tambah mahasiswa' and contains a form with three input fields: 'NPM', 'Name', and 'GPA'. Below these fields is a 'Save' button.

view.html

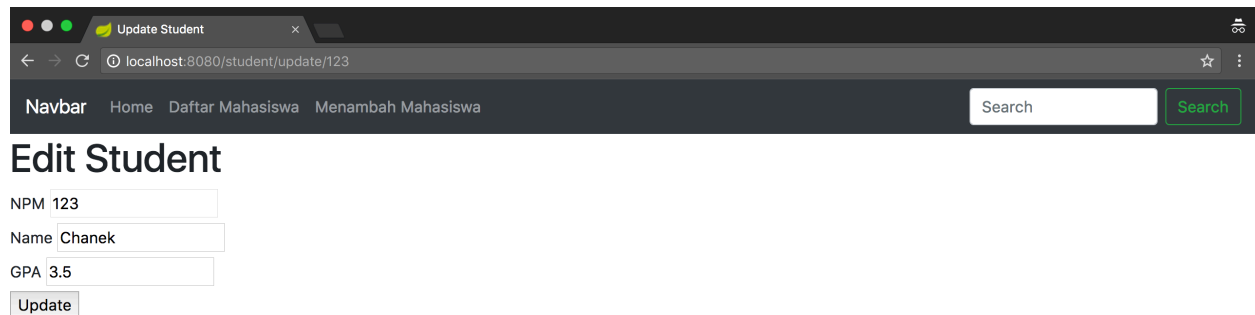


The screenshot shows a web browser window titled 'View Student By NPM' with the URL 'localhost:8080/student/view?npm=123'. The page displays the student information for NPM 123: 'NPM = 123', 'Name = Chanek', and 'GPA = 3.5'.



The screenshot shows a web browser window titled 'View Student By NPM' with the URL 'localhost:8080/student/view/123'. The page displays the student information for NPM 123: 'NPM = 123', 'Name = Chanek', and 'GPA = 3.5'.

form-update.html



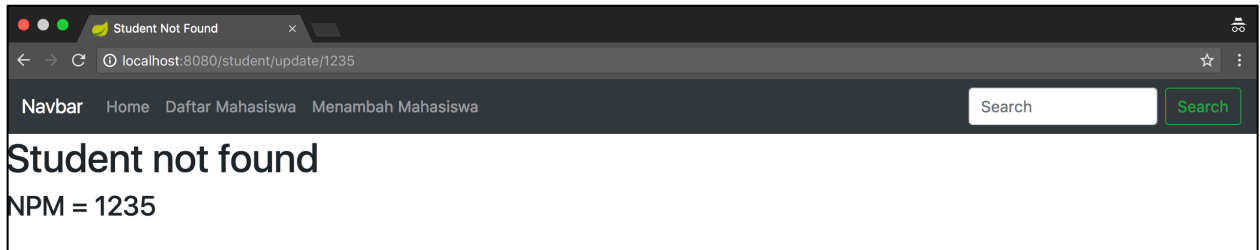
The screenshot shows a web browser window titled 'Update Student' with the URL 'localhost:8080/student/update/123'. The page has a dark navbar with links 'Home', 'Daftar Mahasiswa', and 'Menambah Mahasiswa'. A search bar is in the top right. The main content area is titled 'Edit Student' and contains a form with three input fields: 'NPM' (containing '123'), 'Name' (containing 'Chanek'), and 'GPA' (containing '3.5'). Below these fields is an 'Update' button.



Nama : Winda Dumaria Simanjuntak

NPM : 1706107075

not-found.html



delete.html

