

Tutorial

Initialization

1. Buatlah sebuah folder dengan format NIM_Nama_Week05!
2. Di dalam folder ini buatlah sebuah file index.html, bukalah file index.html menggunakan text editor/IDE kesayangan anda!
3. Copy file main.css dari pertemuan minggu lalu ke folder minggu ini.

Initialization

4. Tuliskan potongan kode pada Gambar 2.1 ke dalam file index.html pada text editor anda

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Javascript</title>
    <script src="main.js" charset="utf-8"></script>
    <link rel="stylesheet" href="main.css" media="screen"
title="no title" charset="utf-8">
    <script src="main.js"></script>
  </head>
  <body onload="onDocumentFinish()">
    <div style="max-width:1200px;">
      <h1>User</h1>
      <table id="tableUser">
        <thead>
          <tr>
            <th>Nama</th>
            <th>Email</th>
            <th>Address</th>
            <th>Website</th>
            <th>Company</th>
            <th>Action</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td style="text-align:center;" colspan="6">No
Data</td>
          </tr>
        </tbody>
      </table>
    </div>
  </body>
</html>
```

Gambar 2.1 Isi dari file index.html

Javascript AJAX

5. Untuk melakukan percobaan request ke sebuah server, kita akan mencoba untuk memperoleh data dari url <https://jsonplaceholder.typicode.com/users> menggunakan HTTP method GET. Tuliskan potongan kode pada Gambar 2.2 ke main.js.

```
function loadUserData() {  
    // membuat sebuah objek untuk melakukan request ke HTTP server  
    // secara asynchronous  
    let request = new XMLHttpRequest()  
    let url = 'https://jsonplaceholder.typicode.com/users'  
    request.open('GET', url, true)  
  
    //handler saat request berhasil di-load  
    request.onload = function() {  
        if (request.status >= 200 && request.status < 400) {  
            let data = JSON.parse(request.responseText)  
            generateUserTable(data);  
        } else {  
            alert('Page Not Found!')  
        }  
    }  
    //handler jika koneksi ke server terganggu  
    request.onerror = function() {  
        alert('Request Failed! Check your internet connection!')  
    }  
    //menjalankan request  
    request.send()  
}  
  
function onDocumentFinish() {  
    loadUserData()  
}
```

Gambar 2.2 Potongan Kode User Data Request

Javascript DOM Manipulation

6. Fungsi generateUserTable di dalam loadUserData digunakan untuk menambahkan row pada table.

```

function generateUserTable(data){
  let idx = 0
  let tbody = document.getElementById('tableUser').querySelector('tbody')
  tbody.innerHTML = ''
  for(idx = 0;idx<data.length;idx++){
    //create row
    let singleRow = document.createElement('tr')

    // create multiple columns
    let colName = document.createElement('td')
    colName.appendChild(document.createTextNode(data[idx].name))
    let colEmail = document.createElement('td')
    colEmail.appendChild(document.createTextNode(data[idx].email))
    let colAddress = document.createElement('td')
    let address = data[idx].address
    colAddress.appendChild(document.createTextNode(address.street+ ' ' +address.suite+ ' ' +address.city))
    let colWebsite = document.createElement('td')
    colWebsite.appendChild(document.createTextNode(data[idx].website))
    let colCompany = document.createElement('td')
    colCompany.appendChild(document.createTextNode(data[idx].company.name))

    //menambahkan column ke row
    singleRow.appendChild(colName)
    singleRow.appendChild(colEmail)
    singleRow.appendChild(colAddress)
    singleRow.appendChild(colWebsite)
    singleRow.appendChild(colCompany)

    //create detail button
    let colAlbum = document.createElement('td')
    let btnAlbums = document.createElement('button')
    btnAlbums.appendChild(document.createTextNode('Albums'))
    //create detail button
    let userId = document.createAttribute('user-id')
    userId.value = data[idx].id
    let userName = document.createAttribute('user-name')
    userName.value = data[idx].name
    //memberikan attribute pada btn, shg nanti valuenya dapat diambil
    btnAlbums.setAttributeNode(userName)
    btnAlbums.setAttributeNode(userId)
    //bind loadAlbum function to event click
    btnAlbums.addEventListener('click',loadAlbum)

    colAlbum.appendChild(btnAlbums)

    singleRow.appendChild(colAlbum)

    tbody.appendChild(singleRow)
  }
}

```

Gambar 2.3 Potongan Kode GenerateUserTable.

Exercise

1. Tambahkan potongan kode di bawah ini pada index.html tepat di bawah div dengan ukuran 1200px!

```
<div style="max-width:600px;">
  <h1 id="albumTitle">Albums</h1>
  <table id="tableAlbum">
    <thead>
      <tr>
        <th>Nama</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td style="text-align:center;" colspan="2">No Data</td>
      </tr>
    </tbody>
  </table>
</div>
<div id="photos" style="text-align:center">
</div>
```

2. Impelementasikan fungsi loadAlbum() pada fungsi generateUserTable() untuk melakukan AJAX Request ke url [https://jsonplaceholder.typicode.com/albums?userId=\(userId:Number\)](https://jsonplaceholder.typicode.com/albums?userId=(userId:Number)). Pada saat button ditekan, data dari server akan digunakan untuk men-*generate* album pada table dengan id tableAlbum. (Hint: userId dapat diambil menggunakan fungsi getAttribute() pada button)

Challenge

3. Tambahkan sebuah th pada thead > tr dengan text "Details".
4. Tambahkan button pada tiap row pada saat fungsi loadAlbum() dijalankan!
5. Bind event loadPhotos() pada tiap button tersebut yang akan melakukan request ke url [https://jsonplaceholder.typicode.com/photos?albumId=\(albumId:Number\)](https://jsonplaceholder.typicode.com/photos?albumId=(albumId:Number)). Dan mengisi div dengan id "photos" dengan tag img yang attribute src ny diperoleh dari url tersebut.
6. Contoh dari soal challenge dapat dilihat seperti pada Gambar di bawah ini.

Albums - Leanne Graham

Nama	Details
quidem molestiae enim	<div>Details</div>
sunt qui excepturi placeat culpa	<div>Details</div>
omnis laborum odio	<div>Details</div>
non esse culpa molestiae omnis sed optio	<div>Details</div>
eaque aut omnis a	<div>Details</div>
natus impedit quibusdam illo est	<div>Details</div>
quibusdam autem aliquid et et quia	<div>Details</div>
qui fuga est a eum	<div>Details</div>
saepe unde necessitatibus rem	<div>Details</div>
distinctio laborum qui	<div>Details</div>

Pada saat button Details ditekan, request akan dilakukan dan div photos akan berisikan gambar seperti pada Gambar di bawah ini.

