# Tutorial

**Initialization**

1. Buatlah sebuah folder dengan format NIM\_Nama\_Week06!
2. Di dalam folder ini buatlah sebuah file main.css. Isi dari file main.css dapat dilihat seperti pada gambar di bawah ini.

# Prepare Catalog CSS

**body**{

padding:20px;

}

.card-container{ width:100%; padding-top: 80px; padding-left:12px;

padding-right:12px;

}

.card {

padding:24px; width:20%; float:left;

box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2); transition: 0.3s;

border-radius: 5px; */\* 5px rounded corners \*/*

height:450px;

}

.card **img**{

width:100%; height:auto;

}

.card:hover {

box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2);

}

.container {

padding: 2px 16px;

}

**input**[**type**=**checkbox**]{ width: 24px; height: 24px; position: relative; top: 6px;

padding: 0px; margin: 0px; overflow: visible;

}

**input**[**type**=**text**]:focus { width: 40%;

}

**input**[**type**=**text**] { width: 130px;

box-sizing: border-box; border: 2px solid #ccc; border-radius: 4px; font-size: 16px; background-color: white;

background-image: url('images/search.png'); background-size: 21px 21px;

background-position: 10px 10px; background-repeat: no-repeat; padding: 12px 20px 12px 40px;

-webkit-transition: width 0.4s ease-in-out; transition: width 0.4s ease-in-out;

}

**label**{

font-family: Verdana; font-size: 16px;

}

**Initialization**

1. Setelah CSS di atas selesai dituliskan, buatlah sebuah file index.html. Isi dari file index.html dapat dilihat pada gambar di bawah ini.

<!DOCTYPE html>

**<html>**

**<head>**

**<title>**Javascript Functional Programming**</title>**

**<link** rel="stylesheet" type="text/css" href="main.css"**>**

**<script** type="text/javascript" src="script.js"**></script>**

**</head>**

**<body** onload="initialize()"**>**

**<div>**

**<input** type="text" name="txtSearch" id="txtSearch" placeholder="Search..."**>**

**</div>**

**<div>**

**<input** type="checkbox" name="cboDiscount" id="cboDiscount"**>**

**<label>**Apply Member Discount**</label>**

**</div>**

**<div** id="card-container" class="card-container"**>**

**</div>**

**</body>**

**</html>**

# Just Javascript Things.

1. Buatlah sebuah file bernama script.js dan isi dari file ini dapat dilihat seperti pada gambar di bawah ini.

function initialize(){

*//mengambil item dari localStorage*

let products = localStorage.getItem("products") let discountFlag = false;

*//jika item tidak tersedia*

if(products === null){

*//fetch data menggunakan fungsi fetchData()*

fetchData()

*//mengambil ulang item*

products = localStorage.getItem("products")

}

*//mengubah item menjadi JavaScript Object*

products = JSON.parse(products)

*//menampilkan produk dalam bentuk card menggunakan fungsi renderAllProduct*

renderAllProduct(products)

}

1. Isi dari fungsi fetchData dapat dilihat seperti pada gambar di bawah ini.

function fetchData(){

let req = new XMLHttpRequest()

req.onreadystatechange = function(){ if(req.readyState != 4 || req.status != 200)

return;

let data = JSON.parse(req.responseText)

*//menyimpan data dengan access key 'products' di localStorage*

localStorage.setItem("products",req.responseText)

}

req.onerror = function() { alert(req.responseText)

}

req.open('GET','productdata.json',true) req.send()

}

# Functional Programming (Map & forEach)

1. Isi dari fungsi renderAllProduct dapat dilihat seperti pada gambar di bawah ini.

function renderAllProduct(products,discountFlag){ products = JSON.parse(JSON.stringify(products)) if(discountFlag)

products.map(product => { product.price \*= 0.75 return product

})

document.getElementById('card-container').innerHTML = '' products.forEach( product => renderProduct(product))

}

1. Seperti yang dapat dilihat pada potongan kode di atas terdapat penggunaan fungsi map() dan juga forEach(). Kedua fungsi tersebut sama-sama berfungsi untuk melakukan iterasi terhadap sebuah array. Perbedaan dari kedua fungsi adalah fungsi map() melakukan transformasi terhadap setiap element array menggunakan nilai kembalian yang telah ditentukan, sedangkan fungsi forEach tidak memiliki nilai kembalian.
2. Pada contoh kasus di atas fungsi map digunakan untuk mentransformasikan list product menjadi list product dengan nilai harga \* 0.75, sedangkan fungsi forEach digunakan untuk menjalankan fungsi renderProduct(). Isi dari fungsi renderProduct dapat dilihat pada gambar di bawah ini.

function renderProduct(product){

const cardContainer = document.getElementById('card-container') let card = document.createElement('div')

const cardClass = document.createAttribute('class') cardClass.value = 'card' card.setAttributeNode(cardClass)

let img = document.createElement('img') img.src = '/images/'+product.image

card.appendChild(img)

let container = document.createElement('div')

const containerClass = document.createAttribute('class') containerClass.value = 'container' container.setAttributeNode(containerClass)

let itemName = document.createElement('h4') itemName.appendChild(document.createTextNode(product.name)) container.appendChild(itemName)

let itemPrice = document.createElement('p') itemPrice.appendChild(document.createTextNode(product.price)) container.appendChild(itemPrice)

let itemSeller = document.createElement('p')

let seller = product.seller.name+"("+product.seller.location+")" itemSeller.appendChild(document.createTextNode(seller))

container.appendChild(itemSeller) card.appendChild(container)

cardContainer.appendChild(card)

}

# Functional Programming (Filter)

1. Agar fungsi search dan applyMemberDiscount pada website berjalan dengan baik, tambahkan potongan kode berikut pada fungsi **initialize()**

let displayProducts = products

let txtSearch = document.getElementById('txtSearch')

txtSearch.addEventListener('keyup',function(){ let searchWord = this.value.toLowerCase() displayProducts = products.filter(product =>

product.name.toLowerCase().includes(searchWord)

)

renderAllProduct(displayProducts,discountFlag)

})

let cboDiscount = document.getElementById('cboDiscount') cboDiscount.addEventListener('change',function(){

if(this.checked){

discountFlag = true

}else{

discountFlag = false

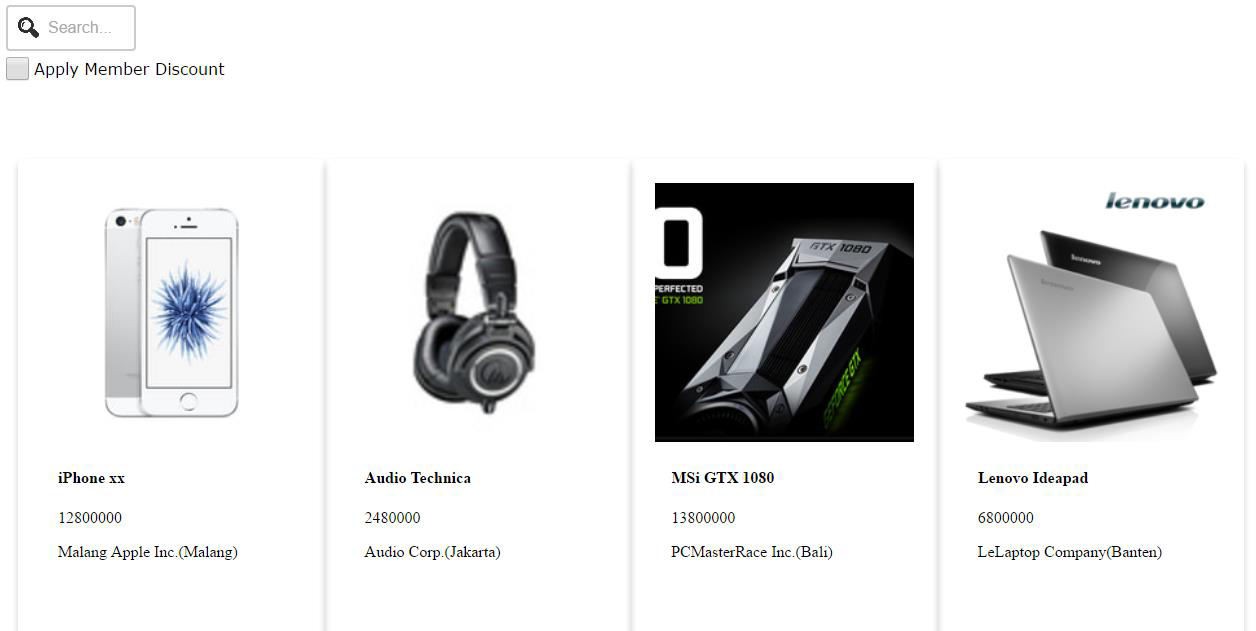
}

renderAllProduct(displayProducts,discountFlag)

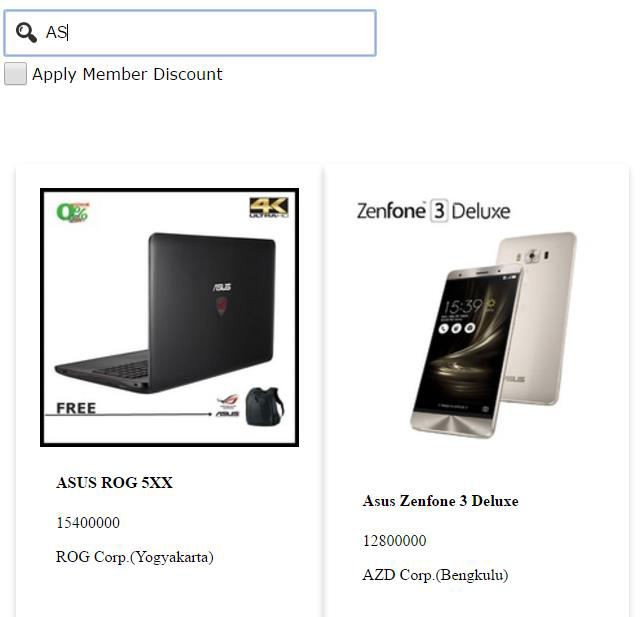
})

# Results

1. Hasil dari potongan kode yang telah anda tuliskan dapat dilihat seperti pada gambar berikut.



1. Hasil dari potongan kode saat fitur search digunakan



1. Hasil dari potongan kode saat Apply Member Discount ditekan (harga akan berkurang)

