

REPORT PRAKTIKUM

“Praktikum Pemrograman Web 1”

Yang diampu oleh Ibu Yati Nurhayati, S.Kom., M.Kom.

MODUL 6



NIM : 20230810088
NAMA : Muhammad Rizal Nurfirdaus
KELAS : TINFC-2023-04

TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS KUNINGAN

Post Test

1. M6-1.html

```
<!DOCTYPE html>
```

```
<html lang = "id">
```

```
<head>
```

```
<meta charset = "UTF-8" />
```

```
<meta name = "viewport" content = "width=device-width", initial-scale =  
1.0>
```

```
<title>JQuery</title>
```

```
<script src = "jquery-3.7.1.js"></script>
```

```
<script>
```

```
$(document).ready(function(){
```

```
    $(".tampil").hide();
```

```
    $(".sembunyi").click(function(){
```

```
        $(".uniku").hide("slow");
```

```
        $(".tampil").show();
```

```
        $(".sembunyi").hide();
```

```
    });
```

```
    $(".tampil").click(function(){
```

```
        $(".uniku").show("slow");
```

```
        $(".tampil").hide();
```

```
        $(".sembunyi").show();
```

```
    });
```

```
});
```

```

</script>

</head>

<body>

    <button class = "sembunyi">Hide</button>

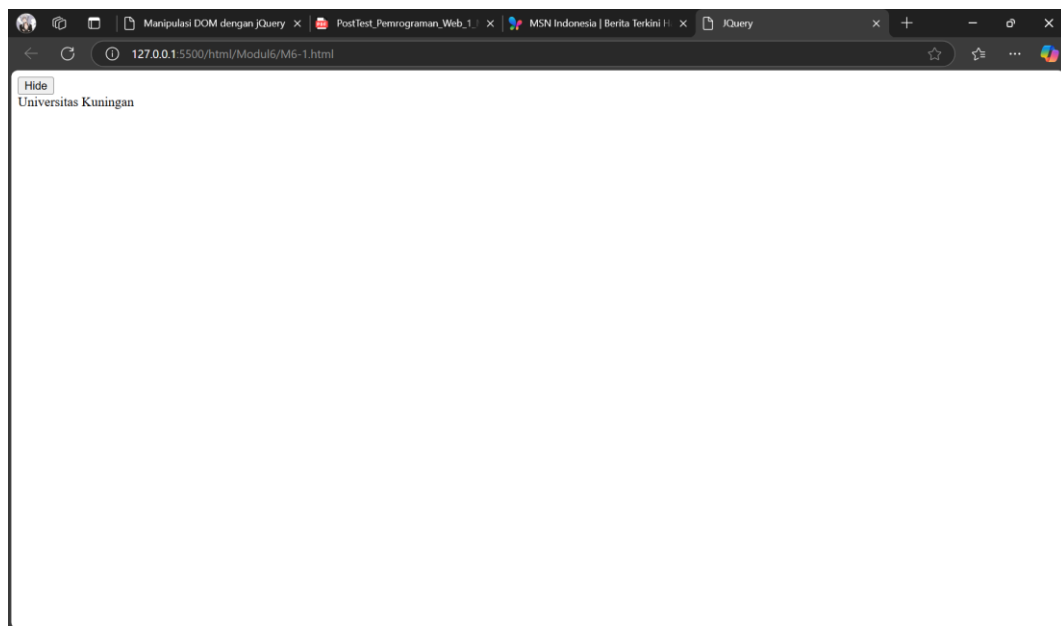
    <button class = "tampil">Show</button>

    <div class = "uniku">Universitas Kuningan</div>

</body>

</html>

```



Analisis :

Kode di atas adalah implementasi sederhana dari manipulasi elemen menggunakan jQuery. Halaman web ini memiliki dua tombol, Hide dan Show, serta sebuah elemen <div> yang menampilkan teks Universitas Kuningan. Saat halaman dimuat, tombol Show disembunyikan menggunakan \$(".tampil").hide();. Ketika tombol Hide diklik, elemen dengan kelas uniku akan disembunyikan secara perlahan (hide("slow")), lalu tombol Hide juga disembunyikan, dan tombol Show akan muncul. Sebaliknya, ketika tombol Show diklik, elemen uniku ditampilkan kembali (show("slow")), tombol Show disembunyikan, dan tombol Hide muncul. Interaksi ini memanfaatkan fungsi hide(), show(), dan click() dari jQuery untuk memberikan pengalaman dinamis pada pengguna.

2. M6-2.html

```
<!DOCTYPE html>

<html lang = "id">

  <head>

    <meta charset = "UTF-8" />

    <meta name = "viewport" content = "width=device-width", initial-scale =
1.0>

    <title>JQuery</title>

    <script src = "jquery-3.7.1.js"></script>

    <script>

      $(document).ready(function(){

        $(".fadeout").click(function(){

          $(".img").fadeOut();

        });

        $(".fadein").click(function(){

          $(".img").fadeIn();

        });

        $(".fadetoogle").click(function(){

          $(".img").fadeToggle();

          $(".img").css("background-color", "orange");

        });

        $(".fadeto").click(function(){

          $(".img").fadeTo("slow", 0.5);

          $(".img").css("background-color", "light-pink");
```

```

        });

    });

</script>

</head>

<body>

    <button class = "fadeout" style = "background-color : #add8e6;">Fade
    Out</button>

    <button class = "fadein" style = "background-color : #00bfff;">Fade
    In</button>

    <button class = "fadetoggle" style = "background-color : #ffa500;">Fade
    Toggle</button>

    <button class = "fadeto" style = "background-color : #ffb6c1;">Fade
    To</button>

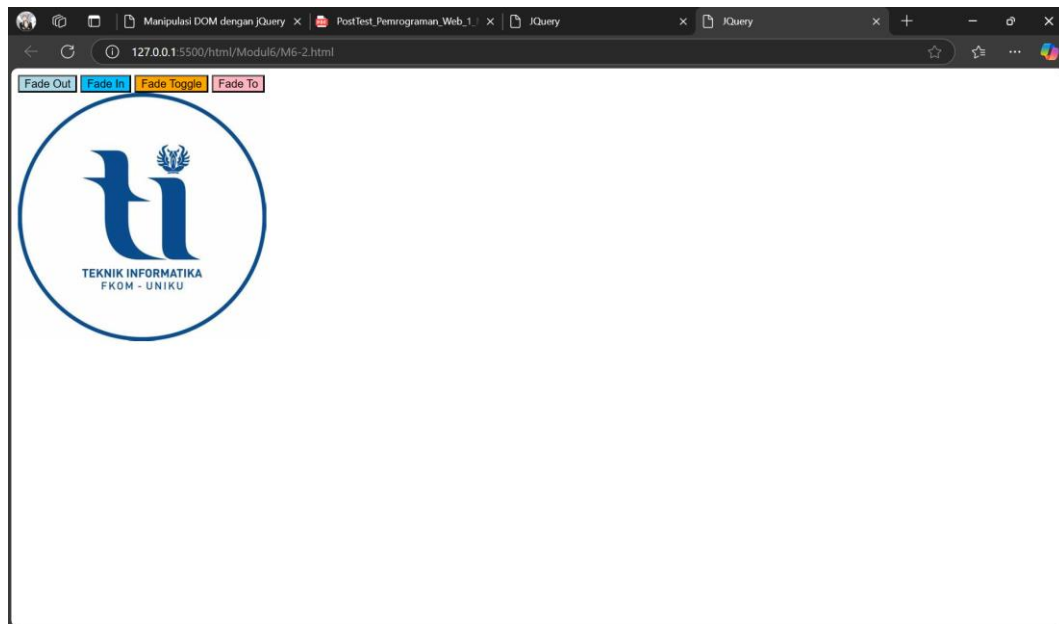
    <br/>

    <img src =
    "https://th.bing.com/th/id/OIP.LWVwJY4gfS7oVBdnVQauAQHaHa?rs=1&pid=I
    mgDetMain" width = "300" height = "300"/>

    </body>

</html>

```



Analisis : Kode di atas adalah implementasi efek animasi fade menggunakan jQuery pada sebuah gambar. Halaman memiliki empat tombol dengan fungsi berbeda: Fade Out menyembunyikan gambar menggunakan `fadeOut()`, Fade In menampilkan gambar kembali dengan `fadeIn()`, Fade Toggle mengatur gambar untuk muncul atau menghilang bergantian menggunakan `fadeToggle()` sambil memberi warna latar belakang orange dengan `css()`, dan Fade To mengatur transparansi gambar ke 50% (`opacity: 0.5`) dengan `fadeTo("slow", 0.5)` sambil mencoba menambahkan warna latar belakang light-pink (meski tidak valid karena nama warna salah). Efek animasi ini memberikan kontrol dinamis atas visibilitas gambar untuk interaksi pengguna.

3. M6-3.html

```
<!DOCTYPE html>
```

```
<html lang = "id">
```

```
<head>
```

```
<meta charset = "UTF-8" />
```

```
<meta name = "viewport" content = "width=device-width", initial-scale = 1.0>
```

```
<title>jQuery</title>
```

```
<script src = "jquery-3.7.1.js"></script>
```

```

<style>

.tab{

    display : none;

}

.tab-button{

    background-color : #f1f1f1;

    cursor : pointer;

    padding : 10px;

    border : 1px solid #ccc;

    display : inline-block;

}

.active{

    background-color : #ccc;

}

</style>

</head>

<body>

<h1>Tugas Web Developer</h1>

<div>

    <div class = "tab-button active" data-tab = "tab1">Tab 1</div>

    <div class = "tab-button" data-tab = "tab2">Tab 2</div>

    <div class = "tab-button" data-tab = "tab3">Tab 3</div>

</div>

```

```
<div id = "tab1" class = "tab" style = "display : block">
```

```
    <h2>Front-End Developer</h2>
```

```
    <p>Mengoptimalkan Fungsi Aplikasi berdasarkan User Interface (UI)  
        dan User Experience (UX).</p>
```

```
    </p>
```

```
</div>
```

```
<div id = "tab2" class = "tab">
```

```
    <h2>Back-End Development</h2>
```

```
    <p>Berjalan di Sisi Server dengan Mengelola Data menjadi  
        Informasi dari Database.
```

```
    </p>
```

```
</div>
```

```
<div id = "tab3" class = "tab">
```

```
    <h2>Full-Stack Development</h2>
```

```
    <p>Memiliki Kemampuan di Kedua Sisi yaitu, Front-End dan Back-End,  
    Sehingga Dapat
```

```
        Menangani Semua Aspek Pengembangan Web.
```

```
    </p>
```

```
</div>
```

```
<script>
```

```
    $(document).ready(function(){
```

```
        $(".tab-button").click(function(){
```



```

        $(".tab").hide();

        $(".tab-button").removeClass("active");

        tabToShow = $(this).data("tab");

        $("#" + tabToShow).show();

        $(this).addClass("active");

    });

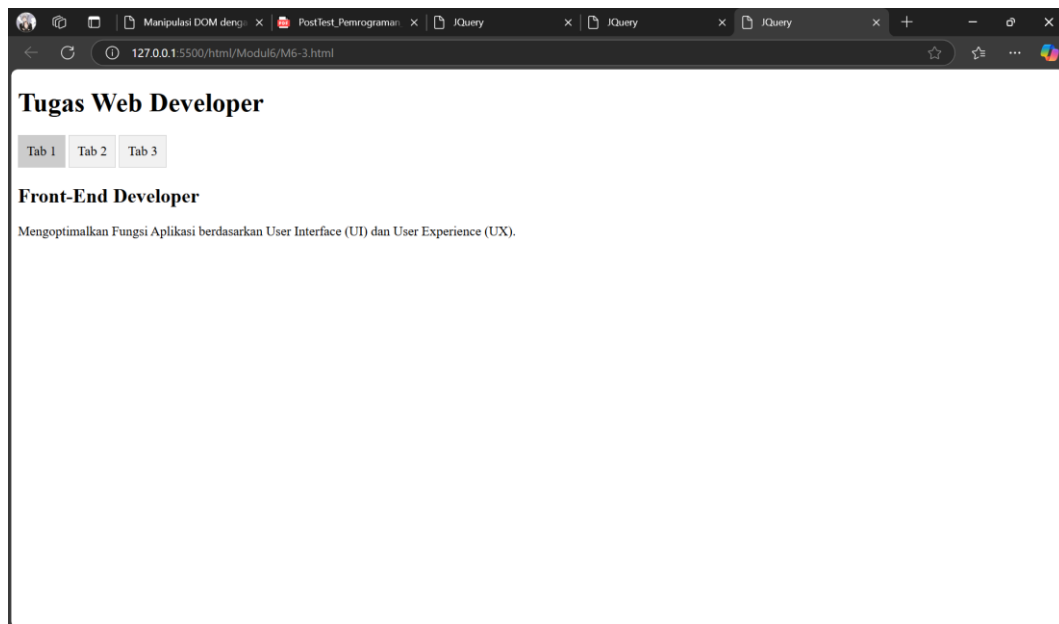
});

</script>

</body>

</html>

```



Analisi : Kode di atas adalah implementasi sistem tab navigation menggunakan jQuery. Halaman ini memiliki tiga tombol tab yang mengontrol tampilan konten: Tab 1, Tab 2, dan Tab 3, yang masing-masing menampilkan deskripsi tentang tugas seorang Front-End, Back-End, dan Full-Stack Developer. Secara default, konten Tab 1 ditampilkan, sedangkan lainnya disembunyikan (display: none). Saat pengguna mengklik sebuah tab, fungsi jQuery dijalankan untuk menyembunyikan semua elemen tab (`$(".tab").hide()`), menampilkan tab yang

relevan dengan mengambil data atribut data-tab, dan menambahkan kelas active untuk memberikan efek visual pada tombol tab yang dipilih.

4. M6-4.html

```
<!DOCTYPE html>

<html lang = "id">

  <head>

    <meta charset = "UTF-8" />

    <meta name = "viewport" content = "width=device-width", initial-scale = 1.0>

    <title>Pie Chart dan Bar Chart</title>

    <script src = "jquery-3.7.1.js"></script>

    <script src = "chart.js"></script>

  </head>

  <body>

    <h1>Data Tingkat Pendidikan Masyarakat (dalam %)</h1>

    <h2>Tahun 2024</h2>

    <canvas id = "myPieChart"></canvas>

    <script>

      $(document).ready(function(){

        pieData = {

          labels : ["SD", "SMP", "SMA"],

          datasets : [

            {

              data : [50, 30, 20],
```

```
        backgroundColor : ["#ff6384", "#36a2eb", "#ffce56"]
    }
]
}

ctxPie = document.getElementById("myPieChart").getContext("2d");

myPieChart = new Chart(ctxPie, {

    type : "pie",

    data : pieData,

});

});

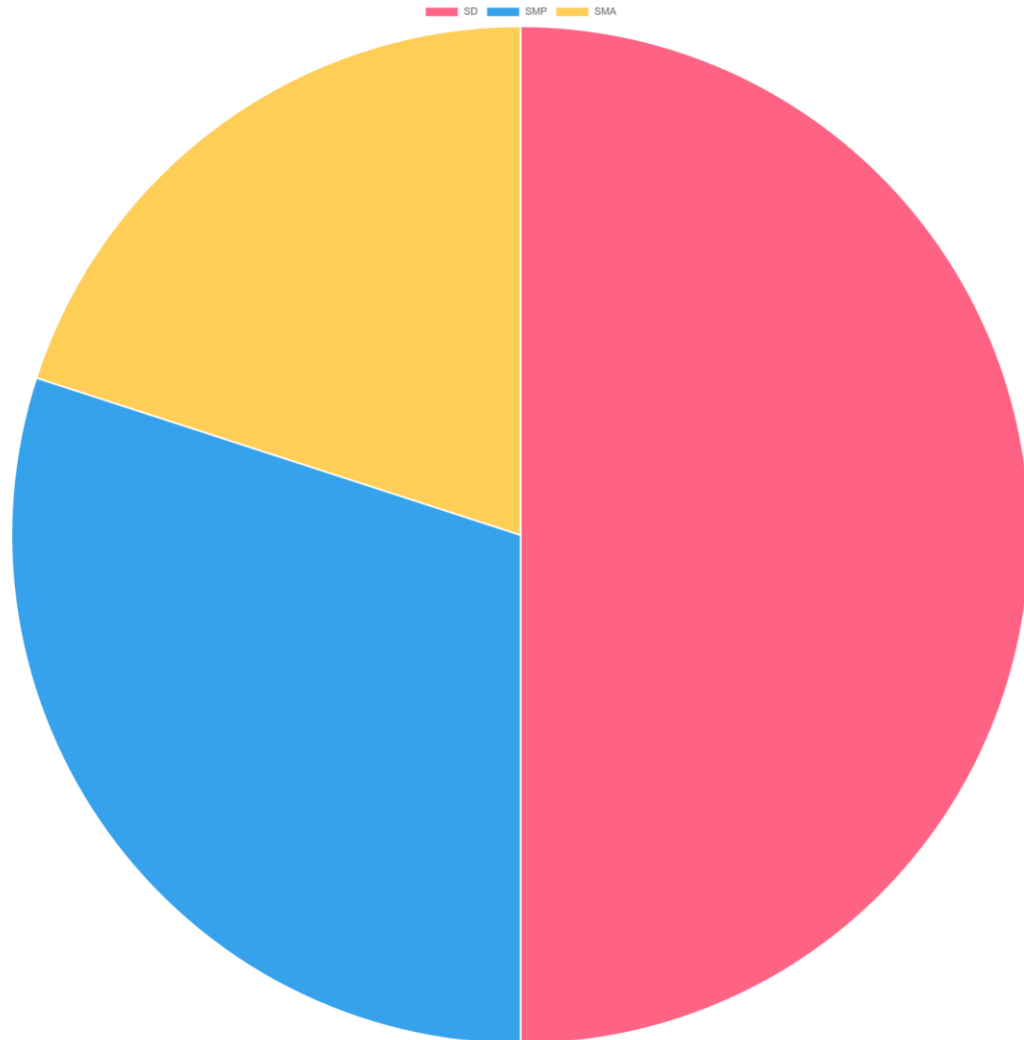
</script>

</body>

</html>
```

Data Tingkat Pendidikan Masyarakat (dalam %)

Tahun 2024



Analisis : Kode di atas adalah implementasi pembuatan Pie Chart menggunakan Chart.js untuk menampilkan data tingkat pendidikan masyarakat pada tahun 2024. Grafik ini menggambarkan persentase tingkat pendidikan, yaitu SD (50%), SMP (30%), dan SMA (20%), dengan masing-masing bagian pie diberi warna berbeda: merah muda (#ff6384), biru (#36a2eb), dan kuning (#ffce56). Elemen `<canvas>` digunakan sebagai wadah grafik, dan grafik dirender dengan mengambil konteks 2D melalui `getContext("2d")`. Data dan konfigurasi chart didefinisikan dalam objek `pieData`, yang mencakup label dan dataset dengan nilai dan warna. Fungsi jQuery memastikan grafik dibuat saat dokumen selesai dimuat, memberikan visualisasi data yang interaktif dan informatif.

5. M6-5.html

```

<!DOCTYPE html>

<html lang = "id">

  <head>

    <meta charset = "UTF-8" />

    <meta name = "viewport" content = "width=device-width", initial-scale =
1.0>

    <title>Akses Kamera dengan JQuery</title>

    <script src = "jquery-3.7.1.js"></script>

  </head>

  <body>

    <h1>Menampilkan Video dari Kamera</h1>

    <button id = "start">Mulai Kamera</button>

    <button id = "stop">Hentikan Kamera</button>

    <script>

      $(document).ready(function(){

        const video = document.createElement("video");

        let mediaStream;

        $("#start").click(function(){

          navigator.mediaDevices

            .getUserMedia({ video: true })

            .then(function(stream){

              mediaStream = stream;

              video.srcObject = stream;

```

```
    })

    .catch(function(err){

        console.error("Gagal mengakses Kamera: " + err);

    });

});

$("#stop").click(function(){

    if(mediaStream){

        mediaStream.getTracks().forEach((track) => track.stop());

        video.srcObject = null;

    }

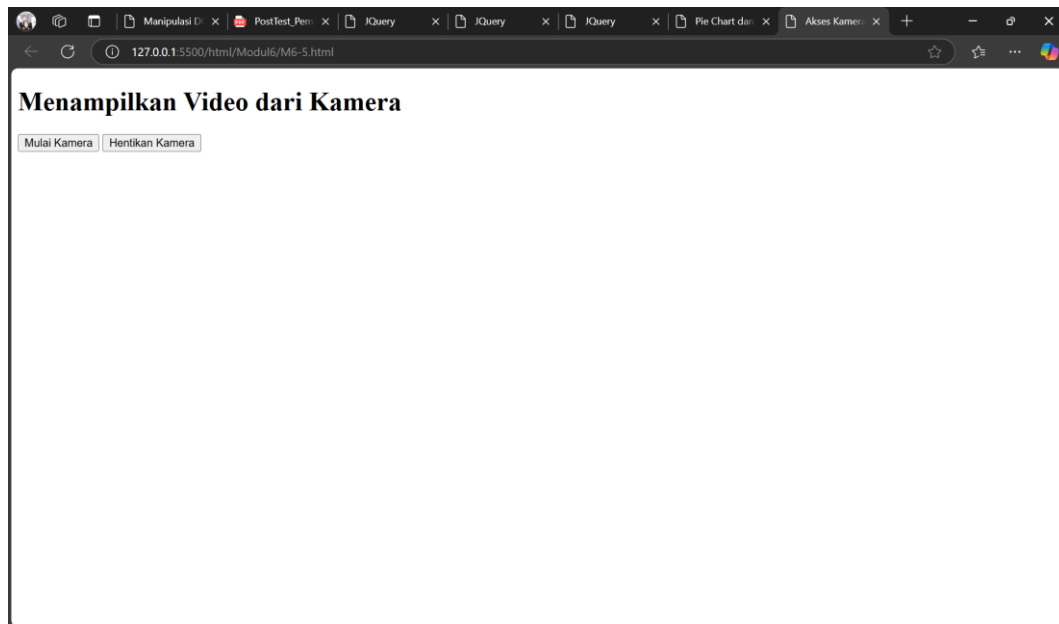
});

});

</script>

</body>

</html>
```



Analisis : Kode di atas adalah implementasi sederhana untuk mengakses kamera perangkat menggunakan Web API dan jQuery. Halaman memiliki dua tombol: Mulai Kamera dan Hentikan Kamera. Ketika tombol Mulai Kamera diklik, fungsi `navigator.mediaDevices.getUserMedia()` dipanggil dengan opsi `{ video: true }` untuk meminta akses kamera, lalu aliran media ditetapkan ke elemen video melalui `video.srcObject`. Jika terjadi kesalahan, pesan error akan ditampilkan di konsol. Tombol Hentikan Kamera menghentikan semua aliran kamera dengan memanggil `getTracks()` pada objek media dan menjalankan `stop()` untuk masing-masing track.

6. M6-6.html

```
<!DOCTYPE html>

<html lang = "id">

  <head>

    <meta charset = "UTF-8" />

    <meta name = "viewport" content = "width=device-width, initial-scale = 1.0">

    <title>Manipulasi DOM dengan JQuery</title>

    <script src = "jquery-3.7.1.js"></script>

  </head>
```

```

<body>

  <h1>Manipulasi DOM dengan JQuery</h1>

  <button id = "tambah">Tambah Data</button>

  <button id = "ubah">Ubah Data Pertama</button>

  <button id = "hapus">Hapus Data Terakhir</button>

  <div id = "output">

    <div class = "item">Data 1</div>

    <div class = "item">Data 2</div>

    <div class = "item">Data 3</div>

  </div>

  <script>

    $(document).ready(function(){

      $("#tambah").on("click", function(){

        $("#output").append('<div class = "item">Data Baru</div>');

      });

      $("#ubah").on("click", function(){

        $("#output .item:first").text("Data Pertama Diubah");

      });

      $("#hapus").on("click", function(){

        $("#output .item:last").remove();

      });

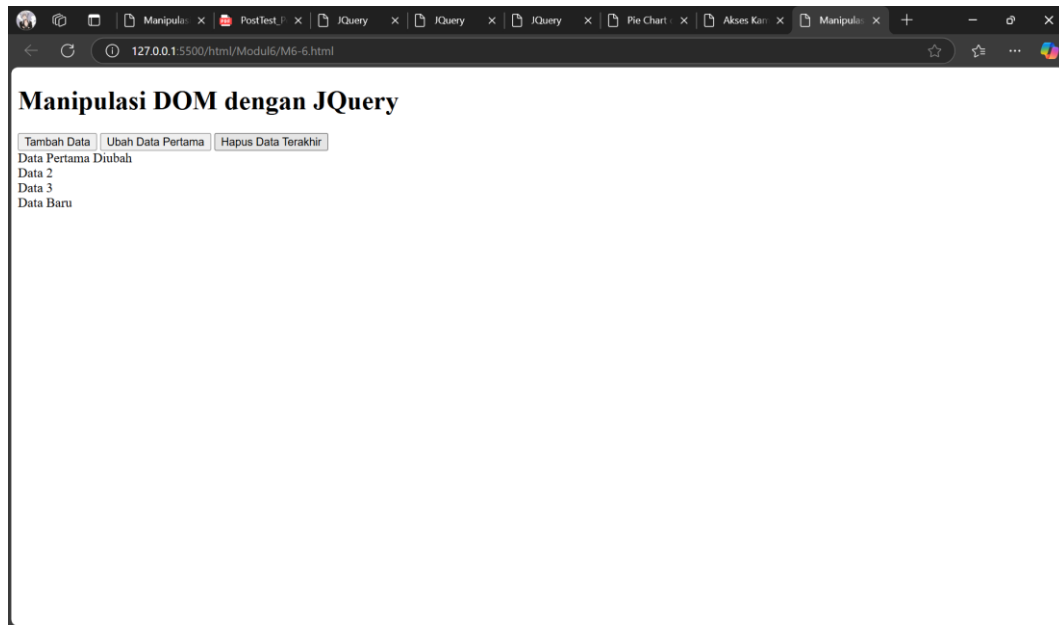
    });

  </script>

```


</body>

</html>



Analisis : Kode di atas adalah implementasi manipulasi DOM menggunakan jQuery untuk menambah, mengubah, dan menghapus elemen secara dinamis. Halaman memiliki tiga tombol: Tambah Data, Ubah Data Pertama, dan Hapus Data Terakhir. Tombol Tambah Data menambahkan elemen <div> baru dengan teks "Data Baru" ke akhir daftar menggunakan metode append(). Tombol Ubah Data Pertama mengganti teks elemen pertama dalam daftar menjadi "Data Pertama Diubah" menggunakan text() dan selektor :first. Tombol Hapus Data Terakhir menghapus elemen terakhir dalam daftar menggunakan remove() dan selektor :last. Dengan interaksi ini, pengguna dapat memodifikasi konten daftar secara dinamis, memberikan pengalaman interaktif yang fleksibel.

7. M6-7.html

<!DOCTYPE html>

<html lang = "id">

<head>

<meta charset = "UTF-8" />

<meta name = "viewport" content = "width=device-width, initial-scale = 1.0">

```
<title>Penerapan Event Handling dengan JQuery</title>

<script src = "jquery-3.7.1.js"></script>

<style>

    body {

        font-family: Arial, sans-serif;

        margin : 20px;

    }

    .box {

        width : 100px;

        height : 100px;

        background-color : lightblue;

        display : inline-block;

        text-align : center;

        line-height : 100px;

        font-weight : bold;

        cursor : pointer;

        transition : background-color 0.3s;

    }

</style>

</head>

<body>

    <h1>Penerapan Event Handling dengan JQuery</h1>

    <div class = "box" id = "box1">Box 1</div>
```

```
<div class = "box" id = "box2">Box 2</div>

<div class = "box" id = "box3">Box 3</div>

<script>

    $(document).ready(function(){

        $(".box").on("click", function(){

            $(this).toggleClass("active");

            alert('Anda mengklik ${$(this).text()}');

        });

        $(".box").on("mouseenter", function(){

            $(this).css("background-color", "lightgreen");

        });

        $(".box").on("mouseleave", function(){

            $(this).css("background-color", "lightblue");

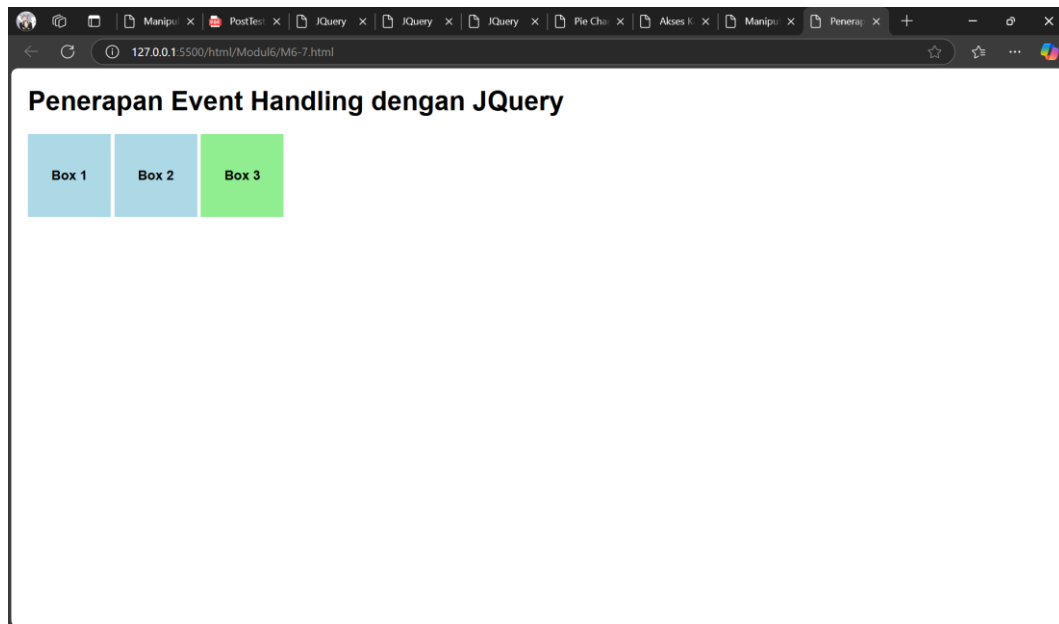
        });

    });

</script>

</body>

</html>
```



Analisis : Kode di atas menunjukkan penerapan event handling menggunakan jQuery untuk menangani berbagai interaksi pengguna dengan elemen-elemen berbentuk kotak (div dengan class box). Setiap kotak memiliki tiga event utama: klik, hover masuk (mouseenter), dan hover keluar (mouseleave). Saat kotak diklik, kelas active ditambahkan atau dihapus dengan toggleClass(), dan sebuah alert menampilkan teks kotak yang diklik. Saat mouse memasuki kotak, warna latar berubah menjadi hijau muda menggunakan css(), dan saat mouse keluar, warna kembali ke biru muda. Interaksi ini memberikan efek visual yang dinamis dan responsif, meningkatkan pengalaman pengguna secara intuitif dengan animasi sederhana pada warna latar.