

LAPORAN PRAKTIKUM PEMROGRAMAN WEB

PRAKTIKUM 4 - JQUERY



Disusun oleh:

Nama : Alfath Roziq Widhayaka

Nim : L0122012

Kelas : A

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA

UNIVERSITAS SEBELAS MARET

2024

A. Penjelasan Source Code

Berikut merupakan source code yang telah saya buat pada praktikum sebelumnya yaitu Website portofolio pribadi yang menggunakan **HTML**, **CSS**, dan **Javascript**. Pada penugasan praktikum kali ini saya akan mengganti/mengubah penggunaan **Javascript** sebelumnya menjadi **JQuery** pada file **script.js** dengan **CDN JQuery**.

▪ Modifikasi Jam Digital JQuery sesuai dengan Local Date

Source code :

```
1 $(document).ready(function() {
2     // Waktu
3     function updateClock() {
4         var now = new Date();
5         var dname = now.getDay(),
6         mo = now.getMonth(),
7         dnum = now.getDate(),
8         yr = now.getFullYear(),
9         hou = now.getHours(),
10        min = now.getMinutes(),
11        sec = now.getSeconds(),
12        pe = "AM";
13
14        if (hou == 0) {
15            hou = 12;
16        }
17        if (hou > 12) {
18            hou = hou - 12;
19            pe = "PM";
20        }
21
22        Number.prototype.pad = function(digits) {
23            for (var n = this.toString(); n.length < digits; n = 0 + n);
24            return n;
25        }
26
27        var months = ["Januari", "Februari", "Maret", "April", "Mei", "Juni", "Juli", "Agustus", "Septemb",
28        var week = ["Minggu", "Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu"];
29        var ids = ["dayname", "month", "daynum", "year", "hour", "minutes", "seconds", "period"];
30        var values = [week[dname], months[mo], dnum.pad(2), yr, hou.pad(2), min.pad(2), sec.pad(2), pe];
31        for (var i = 0; i < ids.length; i++)
32            $("#" + ids[i]).text(values[i]);
33    }
34
35    function initClock() {
36        updateClock();
37        setInterval(updateClock, 1000);
38    }
39
40    initClock();
```

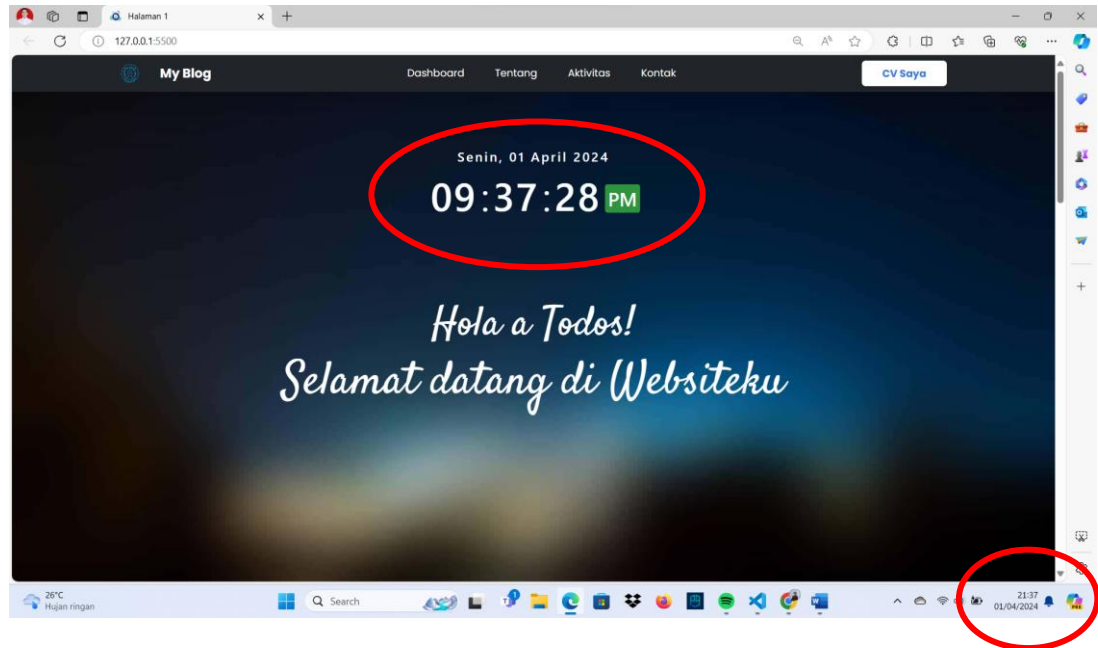
Gambar 1 sc JQuery jam digital local date

Source code JQuery di atas adalah script JQuery library untuk mengubah tampilan waktu pada halaman web. Pada bagian akhir, **initClock()** dipanggil untuk memulai perhitungan waktu saat halaman dimuat pertama kali.

- **updateClock()** digunakan untuk memperbarui tampilan waktu pada halaman web. Itu dijalankan untuk mengambil waktu saat ini menggunakan objek Date JavaScript. Kemudian, waktu tersebut diformat sesuai dengan format yang diinginkan dan disimpan dalam variabel dname (nama hari), mo (bulan), dnum (tanggal), yr (tahun), hou (jam), min (menit), sec (detik), dan pe (AM/PM). Selanjutnya, waktu ditampilkan dengan cara memodifikasi teks pada elemen HTML yang memiliki id sesuai dengan waktu tersebut.

- **initClock()** digunakan untuk memanggil fungsi **updateClock()** sekali saat halaman dimuat pertama kali menggunakan **initClock()**, dan kemudian diatur untuk dipanggil kembali setiap detik menggunakan **setInterval(updateClock, 1000)**.

Hasil :



Gambar 2 hasil jam digital local date

Gambar 2 merupakan hasil jam digital memakai JQuery yang menggunakan local date pada laptop. Terlihat jam nya memiliki waktu yang sama perbedaan nya hanya di Website menggunakan format jam AM PM bukan 24 jam.

- **Modifikasi Animasi Foto JQuery Ketika Cursor Berada diatas Foto**

Source code :

```

1 // Animasi Gambar Tentang
2 $("#gambarheader").mouseenter(function() {
3     $(this).addClass("zoom");
4 });
5
6 $("#gambarheader").mouseleave(function() {
7     $(this).removeClass("zoom");
8 });
9 });

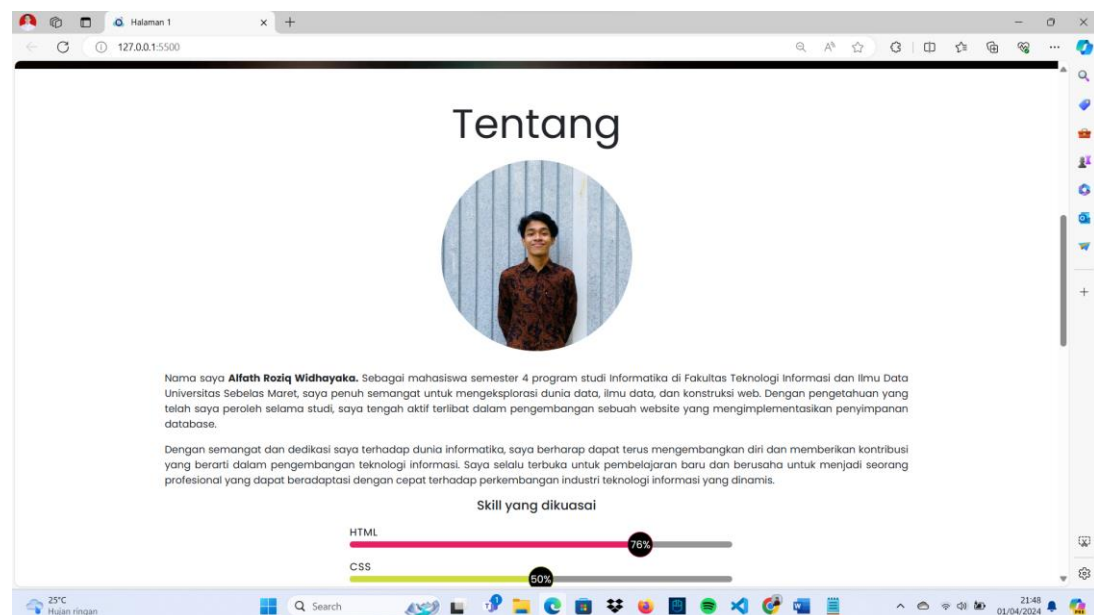
```

Gambar 3 sc JQuery animasi gambar profil di tentang

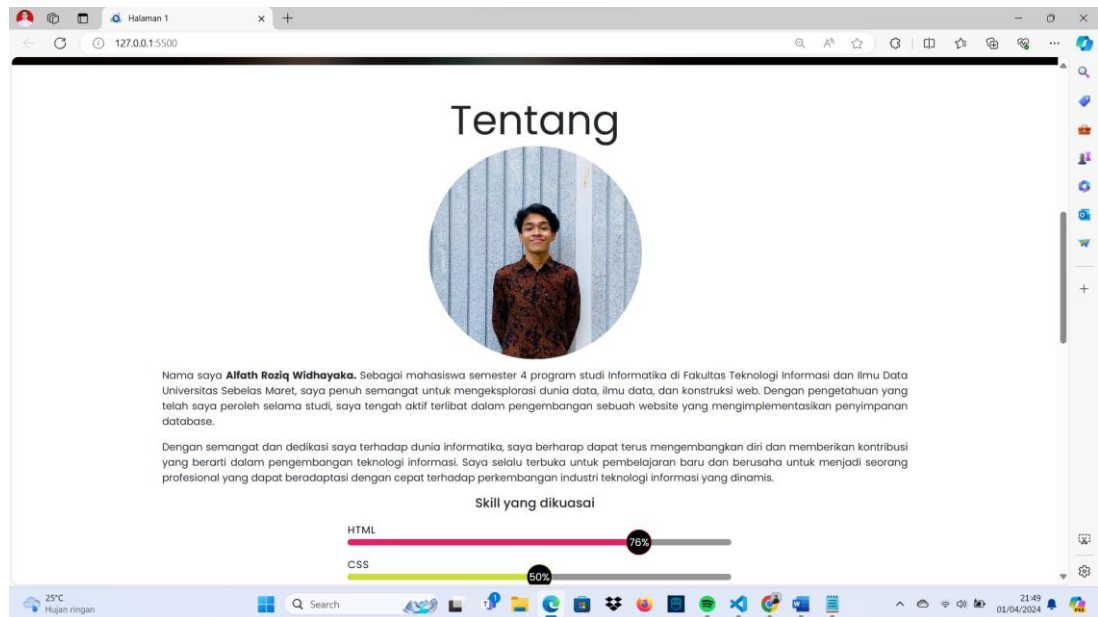
Source code di atas menggunakan JQuery untuk membuat animasi pada gambar saat pengguna mengarahkan kursor mouse ke gambar (mouseenter) dan saat kursor mouse meninggalkan gambar (mouseleave). saat pengguna membawa kursor mouse ke gambar, gambar akan di-zoom secara visual. Ketika kursor mouse meninggalkan gambar, animasi zoom akan dihentikan.

- `$("#gambarheader").mouseenter(function() {...});` merupakan fungsi callback yang didefinisikan akan dijalankan. Di dalamnya, elemen tersebut ditambahkan "zoom" menggunakan `$(this).addClass("zoom")`. Ini berfungsi untuk memicu animasi zoom pada gambar.
- `$("#gambarheader").mouseleave(function() {...});` merupakan penggunaan saat mouse keluar, fungsi callback yang didefinisikan akan dijalankan. Di dalamnya, kelas CSS "zoom" dihapus dari elemen tersebut menggunakan `$(this).removeClass("zoom")`. Ini akan menghentikan animasi zoom pada gambar.

Hasil :



Gambar 4 sebelum hasil animasi gambar



Gambar 5 sesudah hasil animasi gambar

Gambar 4 merupakan gambar sebelum kursor berada di dalam gambar, terlihat gambar tersebut normal seperti biasa. Gambar 5 merupakan gambar dari hasil ketika kursor berada di dalam gambar tersebut, terlihat gambar tersebut menjadi lebih besar yang memiliki aspek scale 1.11 lebih besar dari gambar normalnya.

- **Modifikasi Animasi Card Aktivitas JQuery Ketika Kursor Berada diatas Foto**

Source code :

```

1 // Animasi Features Card
2 $(document).ready(function () {
3     $(".features .card").mouseenter(function () {
4         $(this).css({
5             transform: "scale(1.05)",
6             transition: "transform 0.3s ease"
7         });
8     }).mouseleave(function () {
9         $(this).css("transform", "scale(1)");
10    });
11 });

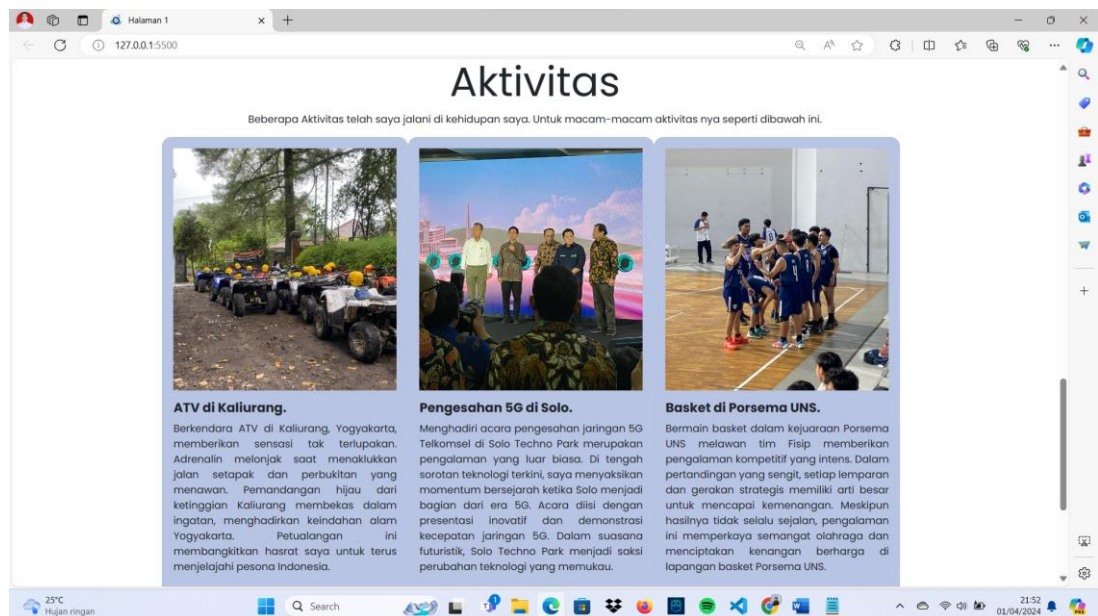
```

Gambar 6 sc JQuery animasi card aktivitas

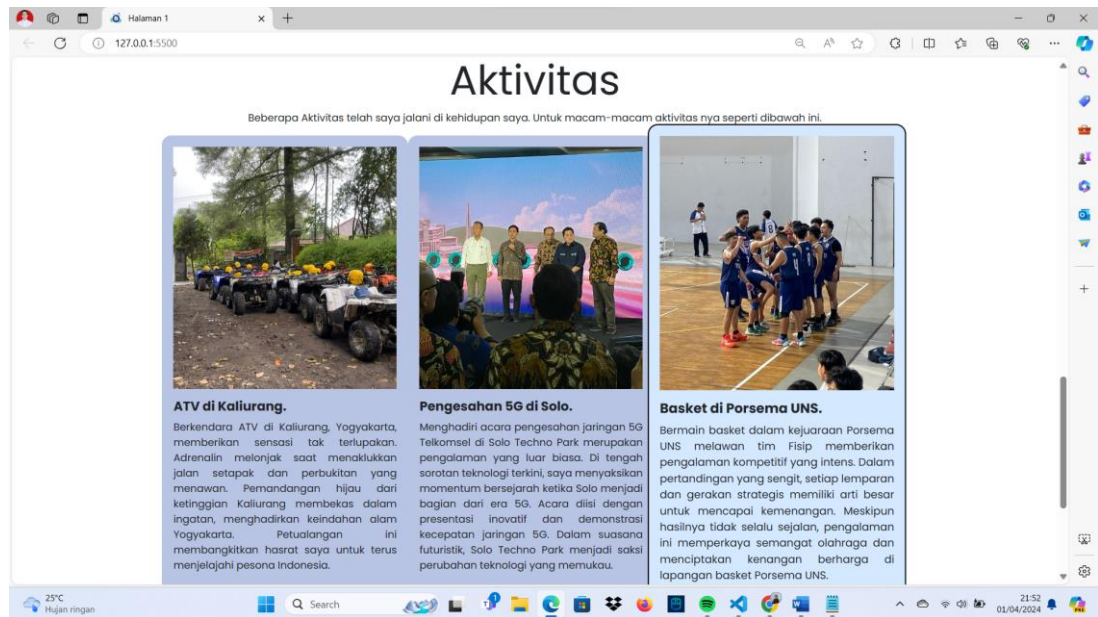
Kode JQuery di atas mengatur animasi untuk kartu fitur aktivitas (features card) ketika pengguna membawa kursor mouse ke kartu, kartu akan membesar sedikit secara visual. Ketika pengguna mengeluarkan kursor mouse dari kartu, kartu akan kembali ke ukuran aslinya.

- `$(".features .card").mouseenter(function () {...}).mouseleave(function () {...});` merupakan fungsi saat mouse masuk ke kartu fitur, fungsi callback pertama dijalankan. Ketika mouse meninggalkan kartu fitur, fungsi callback kedua dijalankan. Saat mouse masuk ke kartu fitur, fungsi callback pertama mengubah "transform" dari kartu fitur tersebut menggunakan metode `$(this).css({...})`. Transform diubah menjadi `scale(1.05)` yang akan membuat kartu fitur tersebut membesar 5% dari ukuran aslinya.
- Saat mouse meninggalkan kartu fitur, fungsi callback kedua mengubah kembali "transform" dari kartu fitur tersebut menjadi skala aslinya dengan menggunakan `$(this).css("transform", "scale(1)")`.

Hasil :



Gambar 7 sebelum hasil animasi card aktivitas



Gambar 8 sesudah hasil animasi card aktivitas

Gambar 7 merupakan gambar sebelum kursor berada di atas kartu fitur aktivitas, terlihat kartu tersebut normal seperti biasa. Gambar 8 merupakan gambar sesudah ketika kursor berada diatas salah satu kartu fitur aktivitas, terlihat salah satu kartu menjadi lebih besar dari sebelumnya. Semua kartu dapat diperbesar jika mouse berada diatas masing-masing kartu.