

LAPORAN PRAKTIKUM

“jQuary”

Dalam rangka memenuhi tugas mata kuliah Praktikum Desain Website

Dosen pengampu: Vearen Dika Sofirudin, S.Pd., M.Ed.



Disusun oleh:

Nama : Rifkhi Ardi Mustaqim

NIM : K3524033

PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER

FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN

UNIVERSITAS SEBELAS MARET

SURAKARTA

2025

Tujuan Praktikum

Tujuan dari praktikum ini adalah untuk:

1. Memahami cara menggunakan JQuery sebagai *framework* JavaScript untuk desain web.
2. Mengerti konsep dan cara kerja dasar JQuery.
3. Mampu mengembangkan dan menerapkan teknik pengolahan konten web yang dinamis menggunakan JQuery dan JQuery UI.

Ringkasan Materi

Praktikum ini berfokus pada JQuery, sebuah *library* JavaScript yang menganut semboyan "*Write less, do more*". JQuery dibuat untuk menyederhanakan berbagai tugas di sisi *client*, seperti manipulasi dokumen HTML, *event handling*, animasi, dan AJAX.

1. Sintaks Dasar dan Selector: Inti dari JQuery adalah penggunaan fungsi Dolar \$ untuk "membungkus" obyek DOM. Penulisan kode JQuery umumnya ditempatkan di dalam fungsi \$(document).ready() untuk memastikan kode dijalankan hanya setelah seluruh halaman selesai dimuat. Seperti CSS, JQuery menggunakan *selector* untuk memilih elemen, seperti \$("#id") untuk ID, \$(".class") untuk kelas, atau \$("element") untuk tag HTML.
2. Perbandingan JQuery vs JavaScript: JQuery menyederhanakan sintaks JavaScript murni secara signifikan.
 - 2.1. JavaScript: document.getElementById("demo").innerHTML = "Hello";
 - 2.2. JQuery: \$("#demo").html("Hello");
3. Manfaat Utama: JQuery mempermudah pengembang dalam:
 - 3.1. Mengakses dan Mengubah Halaman: Mengubah tampilan (CSS) atau isi (HTML) elemen dengan mudah.
 - 3.2. Merespon Interaksi: Melengkapi *event handling* bawaan JavaScript (seperti click, mouseenter, keypress, submit) agar lebih mudah digunakan.
 - 3.3. Animasi: Menyediakan fungsi animasi sederhana (seperti *fading*) yang ringan.
 - 3.4. AJAX: Menyederhanakan proses pengambilan data dari server tanpa perlu *refresh* halaman.
4. JQuery UI: Adalah *library* tambahan yang fokus pada pengembangan *User Interface* (UI) atau *front-end*. Untuk menggunakannya, JQuery utama harus diimpor terlebih dahulu. Modul ini mengenalkan beberapa komponen JQuery UI, yaitu:
 - 4.1. Accordion
 - 4.2. Tabs
 - 4.3. AutoComplete
 - 4.4. Dialog
 - 4.5. Tooltip

Langkah Kerja

Praktikum ini terdiri dari dua latihan mandiri (Bagian E).

1. Latihan 1 (Refactor Highlight Teks ke JQuery)

- **Tugas:** Mengubah Latihan 2 dari praktikum sebelumnya (modul JavaScript) tentang *highlight* kalimat agar menggunakan sintaks JQuery.
- **Implementasi:**
 1. Menyimpan file sebagai latihan1.html.
 2. Mengimpor *library* JQuery (lokal atau CDN).
 3. Fungsi `toggleHighlight()` yang sebelumnya menggunakan `document.querySelectorAll` dan perulangan `forEach` diganti.
 4. Fungsi tersebut diganti dengan *event listener* click JQuery yang menargetkan tombol.
 5. Di dalam *listener* tersebut, logika *toggle* disederhanakan menggunakan *selector* kelas JQuery (misal, `$('.to-highlight')`) dan fungsi `.toggleClass('highlight')`.

2. Latihan 2 (Jam Digital START/STOP)

- **Tugas:** Membuat jam digital yang dapat dimulai (START) dan dihentikan (STOP) menggunakan satu tombol dan disimpan sebagai latihan2.html.
- **Implementasi:**
 1. Membuat layout HTML yang terdiri dari satu elemen `<h1>` (misal, `id="jam"`) untuk menampilkan waktu dan satu `<button>` (misal, `id="start"`).
 2. Di dalam `$(document).ready()`, siapkan variabel untuk menyimpan status jam (misal, `isRunning = false`) dan ID dari `setInterval` (misal, `var clockInterval`).
 3. Memberi *event listener* click pada tombol `$('#start')`.
 4. Fungsi `click` akan memeriksa status `isRunning`:
 - 1.4.1. Jika true (sedang berjalan), panggil `clearInterval(clockInterval)` untuk menghentikan jam dan set `isRunning = false`.
 - 1.4.2. Jika false (berhenti), panggil fungsi `updateClock()`, lalu jalankan `setInterval(updateClock, 1000)` (disimpan ke `clockInterval`), dan set `isRunning = true`.
 5. Buat function `updateClock()` yang mengambil `new Date()`, memformat jam, menit, dan detik (pastikan menggunakan awalan '0' jika angka < 10), lalu tampilkan hasilnya ke `$('#jam')` menggunakan `.text()` atau `.html()`.

Dokumentasi Program dan Hasil Program

Latihan1.html

Kode Program:

```
1 <html></html> > <body> > <script> > ready() callback > hover() callback
2
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Latihan 1 - Highlight Hover</title>
7   <script src="https://code.jquery.com/jquery-3.5.1.slim.js"></script>
8   <style>
9     body { font-family: sans-serif; line-height: 1.6; }
10    p {
11      padding: 10px;
12      border: 1px solid #eee;
13      transition: background-color 0.3s;
14      cursor: pointer;
15    }
16    p:hover { background-color: #ff9999; }
17
18    mark {
19      background-color: yellow;
20      color: red;
21      font-weight: bold;
22    }
23  </style>
24 </head>
25 <body>
26
27  <h1>Latihan 1 - Highlight Klik Met (Hover)</h1>
28
29  <p>The Faculty of Teacher Training and Education, Sebelas Maret University, Surakarta is an Educational Personnel Education Institution (LPTK) which has 24 study programs in 6 majors. Each study program has it: 
30
31  <p>The Informatics and Computer Technology Education (PTIK) study program is planned to take shelter in the management of the Engineering and Vocational Education (PTK) department. This placement revises our pr
32
33  <script type="text/javascript">
34  $(document).ready(function(){
35
36    let phrases = [
37      "Faculty of Teacher Training and Education",
38      "Sebelas Maret University",
39      "Surakarta",
40      "Educational Personnel Education Institution (LPTK)",
41      "24 study programs",
42      "6 majors",
43      "superior, strong and intelligent educational staff",
44      "Informatics and Computer Technology Education (PTIK)",
45      "Engineering and Vocational Education (PTK)",
46      "revises our previous statement",
47      "e-NPBA",
48      "PTIK",
49      "prepared to teach in vocational programs",
50      "vocational department",
51      "Building Engineering Education, and Mechanical Engineering Education",
52      "resource sharing",
53      "mathematics study program"
54    ]
55
56    function escapeRegExp(string) {
57      return string.replace(/[-[\]{}^$+?.,\\]/g, '\\$&');
58    }
59
60    phrases.sort((a, b) => b.length - a.length);
61
62    $('p').each(function(){
63      $(this).data('originalHTML', $(this).html());
64    });
65
66    $('p').hover(
67      function() {
68
69        let newHTML = $(this).data('originalHTML');
70
71        phrases.forEach(function(phrase) {
72          let escapedPhrase = escapeRegExp(phrase);
73          let regex = new RegExp(escapedPhrase, 'g');
74          newHTML = newHTML.replace(regex, '<mark>' + phrase + '</mark>');
75        });
76
77        $(this).html(newHTML);
78      },
79      function() {
80
81        $(this).html($(this).data('originalHTML'));
82      }
83    );
84
85  });
86  </script>
87
88 </body>
89 </html>
```

Penjelasan kode:

1. Inisialisasi dan Persiapan Data

Di dalam blok `$(document).ready()`, beberapa langkah persiapan krusial dilakukan sebelum event apapun dapat berjalan:

- **let phrases (Definisi Target)** Sebuah array bernama `phrases` didefinisikan untuk menampung semua `string` (frasa) yang akan menjadi target `highlight`.
 - **function escapeRegExp(string) (Fungsi Utilitas)** Sebuah fungsi utilitas `escapeRegExp` dibuat. Fungsi ini sangat penting untuk "meloloskan" (`escapes`) karakter-karakter khusus (seperti `, (,)`) yang memiliki makna dalam *Regular Expression* (RegEx).

Ini memastikan bahwa frasa seperti "P.MIPA" akan dicari sebagai teks literal, bukan sebagai pola *RegExp*.

- **phrases.sort(...)** (**Optimalisasi Pencarian**) Ini adalah langkah optimasi yang vital. *Array* `phrases` diurutkan berdasarkan panjang *string*, dari **yang terpanjang ke yang terpendek**. Pengurutan ini dilakukan untuk mencegah konflik *highlighting* parsial. Sebagai contoh, dengan pengurutan ini, frasa "PTK department" akan ditemukan sebelum frasa "PTK", sehingga *highlighting* berjalan dengan benar.

2. Penyimpanan State (Pencadangan HTML)

Sebelum menerapkan *event hover*, konten asli dari setiap paragraf perlu dicadangkan agar dapat dikembalikan nanti.

- `$('p').each(function(){ ... })` Dilakukan iterasi (perulangan) pada setiap elemen `<p>` (paragraf) di halaman.
- `$(this).data('originalHtml', ...)` Untuk *setiap* paragraf, konten HTML-nya saat ini (`$(this).html()`) diambil dan disimpan ke dalam *cache* data internal jQuery menggunakan metode `.data()`. Ini berfungsi sebagai "cadangan" atau "snapshot" dari keadaan bersih paragraf tersebut.

3. Implementasi Event Hover (Logika Inti)

Logika utama diimplementasikan menggunakan metode `.hover()` dari jQuery, yang diterapkan pada semua elemen `<p>`. Metode ini menerima dua fungsi: satu untuk *mouse-in* (saat kursor masuk) dan satu untuk *mouse-out* (saat kursor keluar).

A. Logika Mouse-In (Fungsi Pertama)

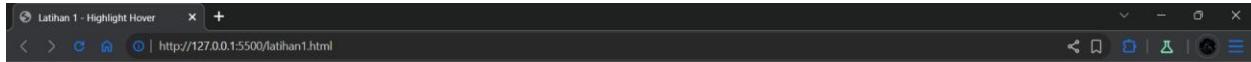
1. **Mengambil Cadangan:** Saat kursor masuk, let `newHtml = $(this).data('originalHtml')` dipanggil. Konten HTML "bersih" yang telah disimpan di `.data()` diambil kembali. Ini penting agar *highlight* tidak menumpuk jika pengguna melakukan *hover* berulang kali.
2. **Iterasi dan Penggantian:** Dilakukan iterasi `forEach` pada *array* `phrases` (yang sudah terurut).
3. Di dalam *loop*, sebuah *RegExp* baru dibuat untuk setiap frasa (yang sudah di-*escape*) dengan *flag* 'g' (global) agar semua kemunculan frasa diganti.
4. Metode `newHtml.replace(...)` dijalankan, mencari frasa target di dalam *string* HTML dan membungkusnya dengan tag `<mark>`.
5. **Pembaruan Tampilan:** Setelah *loop* selesai, `$(this).html(newHtml)` dieksekusi. Konten paragraf diperbarui dengan *string* HTML baru yang kini berisi tag `<mark>`, sehingga *highlight* (yang diatur di CSS) muncul secara visual.

B. Logika Mouse-Out (Fungsi Kedua)

Fungsi ini jauh lebih sederhana dan efisien:

1. **Mengembalikan Cadangan:** Saat kursor keluar, `$(this).html($(this).data('originalHtml'))` dieksekusi.
2. Logika ini mengabaikan HTML yang sedang ditampilkan (yang berisi `<mark>`) dan langsung mengembalikan konten paragraf ke *snapshot "bersih"* yang telah disimpan di `.data()`. Ini secara efektif menghapus semua *highlight* dan mengembalikan paragmat ke keadaan semula.

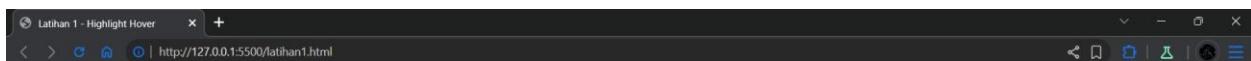
Screenshot hasil:



Latihan 1 - Highlight Kalimat (Hover)

The Faculty of Teacher Training and Education, Sebelas Maret University, Surakarta is an Educational Personnel Education Institution (LPTK) which has 24 study programs in 6 majors. Each study program has its own characteristics in producing superior, strong and intelligent educational staff.

The Informatics and Computer Technology Education (PTIK) study program is planned to take shelter in the management of the Engineering and Vocational Education (PTK) department. This placement revises our previous statement, where the PTIK study program is under the P.MIPA department. PTIK is a study program that will produce graduates who are prioritized to teach in vocational programs. So, it would be better if PTIK was under the PTK department which oversees vocational programs such as Building Engineering Education, and Mechanical Engineering Education. To meet the needs of lecturers in the field of information technology that cannot be met by the PTK department, PTIK will conduct resource sharing with the mathematics study program.



Latihan 1 - Highlight Kalimat (Hover)

The Faculty of Teacher Training and Education, Sebelas Maret University, Surakarta is an Educational Personnel Education Institution (LPTK) which has 24 study programs in 6 majors. Each study program has its own characteristics in producing superior, strong and intelligent educational staff.

The **Informatics and Computer Technology Education (PTIK)** study program is planned to take shelter in the management of the **Engineering and Vocational Education (PTK)** department. This placement **revises our previous statement**, where the **PTIK** study program is under the **P.MIPA** department. **PTIK** is a study program that will produce graduates who are **prioritized to teach in vocational programs**. So, it would be better if **PTIK** was under the **PTK department** which oversees vocational programs such as **Building Engineering Education, and Mechanical Engineering Education**. To meet the needs of lecturers in the field of information technology that cannot be met by the **PTK department**, **PTIK** will conduct **resource sharing** with the **mathematics study program**.

Latihan2.html

Kode Program:

```

42 <script>
43     $(document).ready(function(){
44
45         var clockInterval;
46         var isRunning = false;
47
48         var namaHari = ['Minggu', 'Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat', 'Sabtu'];
49         var namaBulan = ['Januari', 'Februari', 'Maret', 'April', 'Mei', 'Juni',
50                         'Juli', 'Agustus', 'September', 'Oktober', 'November', 'Desember'];
51
52
53         function padZero(num) {
54             return (num < 10) ? '0' + num : num;
55         }
56
57         function updateClock() {
58             var now = new Date();
59
60             var h = padZero(now.getHours());
61             var m = padZero(now.getMinutes());
62             var s = padZero(now.getSeconds());
63             var formattedTime = h + ':' + m + ':' + s;
64
65             var dayName = namaHari[now.getDay()];
66             var dayDate = now.getDate();
67             var monthName = namaBulan[now.getMonth()];
68             var year = now.getFullYear();
69             var formattedDate = dayName + ', ' + dayDate + ' ' + monthName + ' ' + year;
70
71             $('#tanggal').text(formattedDate);
72             $('#jam').text(formattedTime);
73         }
74
75         $('#start').click(function(){
76             if (isRunning) {
77                 clearInterval(clockInterval);
78                 isRunning = false;
79             } else {
80                 updateClock();
81                 clockInterval = setInterval(updateClock, 1000);
82                 isRunning = true;
83             }
84         });
85
86         updateClock();
87     });
88 </script>
89 </body>
90 </html>

```

Penjelasan Kode:

1. Inisialisasi dan Persiapan Variabel

Fungsi utama dari skrip ini dibungkus dalam `$(document).ready()`. Hal ini dilakukan untuk memastikan bahwa semua elemen DOM telah dimuat (*di-render*) sepenuhnya oleh *browser* sebelum skrip jjQuery mencoba untuk memanipulasinya.

Di dalam blok *ready* tersebut, beberapa variabel penting dideklarasikan:

- **clockInterval:** Variabel ini disiapkan (dideklarasikan tanpa nilai) untuk menampung ID unik yang akan dikembalikan oleh fungsi `setInterval`. Penyimpanan ID ini krusial agar interval tersebut dapat dihentikan (dibatalkan) di kemudian waktu menggunakan `clearInterval`.

- **isRunning**: Variabel ini diinisialisasi sebagai false. Variabel ini berfungsi sebagai *flag* (penanda status) boolean untuk mengelola logika *toggle* pada tombol START/STOP.
- **namaHari dan namaBulan**: Dua array ini dibuat untuk berfungsi sebagai *lookup table* (pemetaan). Tujuannya adalah untuk menerjemahkan nilai indeks numerik yang dikembalikan oleh metode `.getDay()` (0-6) dan `.getMonth()` (0-11) dari object Date menjadi string nama hari dan bulan dalam Bahasa Indonesia.

2. Fungsi Utilitas dan Fungsi Inti

Terdapat dua fungsi utama yang membangun logika jam ini:

Fungsi padZero(num) Ini adalah sebuah fungsi utilitas (pembantu) yang dibuat untuk tujuan *formatting*.

- Fungsi ini menerima satu parameter numerik (num).
- Dengan menggunakan **operator ternary**, fungsi ini memeriksa apakah angka tersebut kurang dari 10.
- Jika ya, fungsi akan mengembalikan string dengan awalan '0' (misalnya, 9 menjadi "09"). Jika tidak, angka tersebut dikembalikan seperti sedia kala.

Fungsi updateClock() Ini adalah fungsi "mesin" utama yang bertanggung jawab untuk mengambil waktu saat ini dan menampilkannya ke halaman.

1. **Pengambilan Waktu**: var now = new Date() digunakan untuk membuat object Date baru yang berisi waktu dan tanggal sistem saat ini.
2. **Format Waktu**: Metode `.getHours()`, `.getMinutes()`, dan `.getSeconds()` dipanggil dari object now. Setiap nilai tersebut dilewatkan ke fungsi padZero untuk memastikan format dua digit.
3. **Format Tanggal**: Metode `.getDay()`, `.getDate()`, `.getMonth()`, dan `.getFullYear()` digunakan. Nilai indeks dari `.getDay()` dan `.getMonth()` digunakan untuk mengambil string yang sesuai dari array namaHari dan namaBulan.
4. **Manipulasi DOM**: Nilai string yang sudah diformat (untuk tanggal dan waktu) kemudian disuntikkan ke dalam elemen HTML menggunakan selector ID jQuery `($("#tanggal")` dan `($("#jam")`) dan metode `.text()`.

3. Logika Interaktivitas (Event Handler)

Logika interaktif diimplementasikan menggunakan *event handler* click pada elemen dengan ID #start.

- `$("#start").click(function(){ ... })`; Kode di dalam fungsi ini hanya akan dieksekusi ketika tombol "START/STOP" diklik.
- **Logika Toggle**:
 - Sebuah kondisional if (`isRunning`) digunakan untuk memeriksa status jam saat ini.

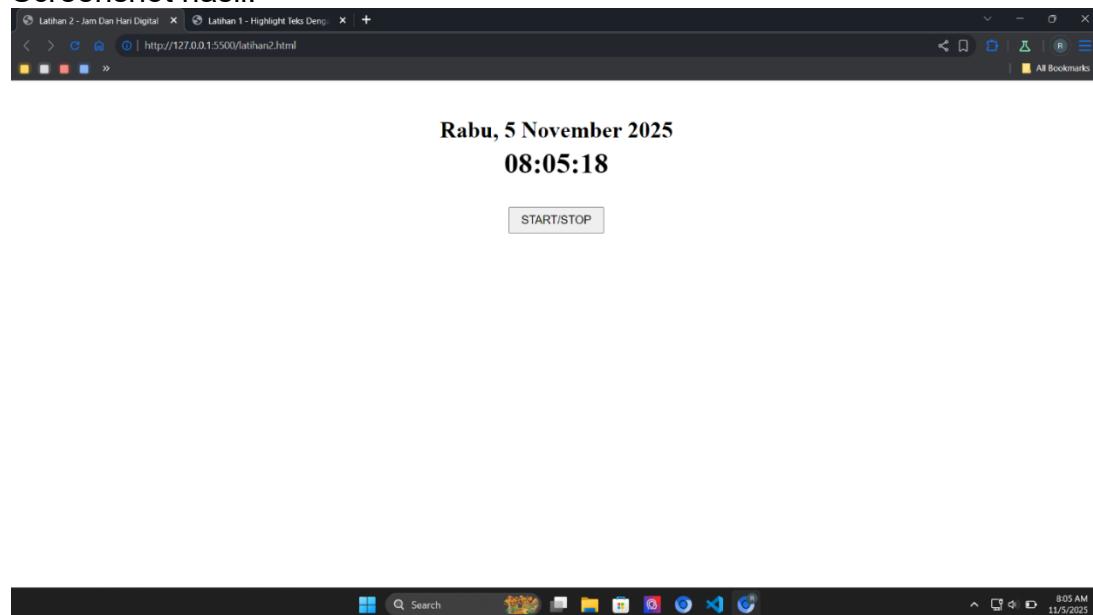
- **Jika true (Jam sedang berjalan):** Perintah clearInterval(clockInterval) dieksekusi untuk menghentikan pemanggilan updateClock yang berulang. Flag isRunning kemudian diatur kembali ke false.
- **Jika false (Jam sedang berhenti):**
 1. updateClock() dipanggil satu kali secara manual. Ini untuk memastikan tampilan jam diperbarui secara instan saat tombol diklik, tanpa menunggu interval 1 detik pertama.
 2. setInterval(updateClock, 1000) dieksekusi. Perintah ini memulai pemanggilan fungsi updateClock secara periodik setiap 1000 milidetik (1 detik).
 3. ID yang dikembalikan oleh setInterval disimpan dalam variabel clockInterval.
 4. Flag isRunning diatur ke true.

4. Inisialisasi Tampilan Awal

Di baris terakhir skrip (namun masih di dalam \$(document).ready()), terdapat satu panggilan fungsi updateClock() di luar *event handler*.

Panggilan ini bertujuan untuk **menginisialisasi tampilan jam dan tanggal segera** setelah halaman dimuat. Tanpa baris ini, elemen <h2> dan <h1> akan menampilkan teks *placeholder* bawaan HTML hingga tombol "START" pertama kali diklik.

Screenshot hasil:



Kesimpulan

Berdasarkan praktikum yang telah dilakukan, dapat ditarik kesimpulan:

Latihan 1: Refactor (Highlight Teks)

Tugas ini adalah tentang **migrasi dan efisiensi sintaks**.

Berhasil mengubah kode JavaScript murni (vanilla JS) menjadi sintaks JQuery. Ini membuktikan prinsip "Write less, do more". Mengganti beberapa baris kode DOM JavaScript (seperti `document.querySelectorAll` dan *looping* `forEach`) dengan satu perintah JQuery yang lebih ringkas, yaitu `$('.to-highlight').toggleClass('highlight')`.

Latihan 2: Membangun (Jam Digital)

Tugas ini adalah tentang **membangun komponen interaktif dan mengelola state**.

Membangun jam digital dari awal. Latihan ini mempraktikkan *event handling* JQuery (`.click()`) untuk mengelola tombol START/STOP. Dalam tugas juga sukses mengelola *state* (status `isRunning`) dan fungsi *timer* (`setInterval` dan `clearInterval`) untuk menjalankan atau menghentikan jam. Ini menunjukkan cara JQuery mengelola *event* dan memanipulasi DOM secara *real-time* (menggunakan `.text()`).