|  |  |
| --- | --- |
| A logo of a building  Description automatically generated | Department of Information Technology - State Polytechnic of Malang  **Jobsheet-6: JQuery**  **Web Design and Programming Courses**  Web Design and Programming Teaching Team  October 2024 |

**Student Identity**

**Name : Lovie Jechonia Tonimba**

**NIM : 244107060101**

**Class : 2G**

**Major : D-IV Business Information System**

**Topic:**

Introduction to basic concepts of jQuery

**Objectives:**

Students are expected to be able to:

1. Understand the basic concepts of jQuery
2. Implementing jQuery

**Introduction**

**Introduction to jQuery**

JQuery was released in 2006 by John Resig. JQuery is a JavaScript library or a collection of "ready-made" JavaScript code/functions. In line with jQuery's slogan, which is "*write less, do more*", jQuery is used to make it easier to compile javaScript code on HTML files. jQuery's syntax is designed in such a way that it makes it easier for programmers to navigate documents, select DOM elements, apply animations, apply *events*, and build AJAX applications. The advantages of using jQuery include:

* Comprehensive documentation and tutorials

The functions provided are well documented along with examples of their use, read on the site http://jquery.com this makes it easier to learn jquery.

* Short and Clear

jQuery prioritizes writing concise and clear code through the sharing of features such aschain-able functions and short function names.

* Addresses cross-browser compatibility issues

JavaScript engines in various browsers are different from each other, so scripts that run in one browser can fail in another. jQuery addresses any of these inconsistencies between browsers and provides an interface that works consistently across all browsers.

* Extensible

jQuery makes framework development very simple. Various new events, elements, and methods can be easily added and reused as plugins.

**Practical Section 1. Preparing to Use JQuery**

There are 2 ways to input jQuery files into HTML: download and access jQuery locally or use a CDN (*Content Delivery Network*).

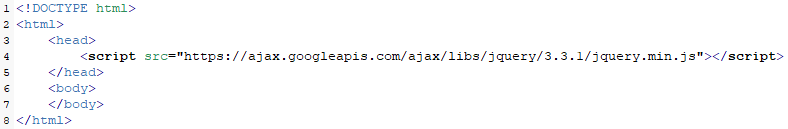
* download and access jQuery locally

In order to use jQuery, users must download the jQuery file and then associate it with the <script> tag. Then follow these steps:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new directory "praktik\_jquery" in the directory dasarWeb that you created in the previous meeting. |
| 2 | Open the official jQuery website on <http://jquery.com/> . then it will appear the main page of jQuery, click the "Download jQuery" button. |
| 3 | Click link “Download the uncompressed, development version of jQuery 3.7.1. Then right clict and choose \“Save as ..”. |
| 4 | Save jquery-3.5.1.js files in the "basicWeb/praktik\_jquery" directory. |
| 5 | To connect the jQuery file with HTML, add a <script tag> by changing the address of the jquery-3.7.1.js to the version we have downloaded i.e. jquery-3.7.1.js. Create a new file and then type in the code like the following example:    The above HTML code must be stored in 1 directory with the jquery-3.7.1.js file |

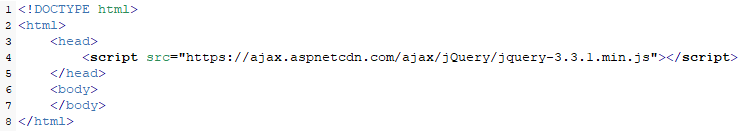
* input jQuery files using CDN (*Content Delivery Network*). CDN is a global repository that can be accessed by anyone. Using a CDN does not require downloading jQuery files, but in order for jQuery to run, it must always be connected to the internet. There are many CDN options, such as Google CDN and Microsoft CDN, how to use them is as follows:

1. Google CDN



Changed with the last version

1. Microsoft CDN



Changed with the last version

In this practicum, you are advised to use jQuery by downloading/accessing jQuery locally. This aims to avoid the possibility of problems in the internet connection.

**Practical Section 2. Document Ready Function**

Once you understand how to use jQuery. Before executing the next code, jQuery will make sure that all the desired elements or elements are already displayed on the web page, the function to be used is the document ready() function. Here is the basic syntax of the document ready() function:

|  |
| --- |
| $(document).ready(function()(  jQuery code line  }); |

To understand the function of document ready() perform the following practicum steps:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new HTML file named ready.html in the dasarWeb/praktik\_jquery directory, type the following program code:    Changed with the last version |
| 2 | Save the file, then open a browser and run it by typing localhost/dasarWeb/praktik\_jquery/ready.html |
| 3 | After the program is run, click the "Click Me!" button, then observe the display on the browser.  When the page loads in the browser, the button element with the value “Click Me!” will be displayed. Once the document is fully ready, the jQuery code inside the <script> will wait for a click event on that button. When the user clicks the button with the ID button1, a browser-native alert box will appear on the screen, displaying the message “This is Week 4 Practicum”. After the user presses the OK button on the alert, the dialog box will disappear, and the page will return to its original state. |
| 4 | Next, modify the file by removing the program code on lines 6 and 10, so that the program code looks like this:    Changed with the last version |
| 5 | Save the file. Then open a browser and run the program code by typing  localhost/dasarWeb/praktik\_jquery/ready.html |
| 6 | After the program is run, click the "Click Me!" button, then observe the display on the browser.  Nothing will happen when the button is clicked.  This code will cause an error in the browser because jQuery and the button event handler are defined within the <head> element, but the event handler (lines 6-10) is not placed within the $(document).ready() function. Because the <script> tag is in the <head>, JavaScript will try to find and attach the .click() function to the element with the ID button1 ($(“#button1”)) before the <body> element (where the button itself is located) is finished loading and exists in the Document Object Model (DOM). As a result, the button element is not found, the event handler is never attached, and when the button is clicked, no action is associated with it, so the alert message will not appear. |
| 7 | From your observations, what do you understand from the document ready() function? Record the results of your observations and explanations of your understanding below (Question No. 1)  The $(document).ready() function is like a “Wait Until Everything Is Ready” command for JavaScript code (especially jQuery). By using $(document).ready(), it tell the script to delay its execution until the browser has finished reading and building the entire page structure. This ensures that when the script finally runs, the targeted elements are definitely there and ready to interact. |

**Practical Section 3. Selector**

It was previously explained that jQuery makes it easy for us to simplify JavaScript code. In other words, jQuery allows for much shorter writing of JavaScript program code. For example, if there is an HTML element like the following:

|  |
| --- |
| <button id="button"> Click Me! </button> |

in JavaScript to find HTML elements use the getElementById() method, so that to access the <button> element that has id="button" is:

|  |
| --- |
| var x = document.getElemenById("button"); |

with jQuery selector the program code writing becomes shorter, to look for HTML elements with id="button" is:

|  |
| --- |
| var x = $(“#button”); |

The jQuery selector functions to select/retrieve HTML elements to be manipulated. Here are some ways the selector selects/retrieves HTML elements:

* selector tag

Use the Tag Selector by directly mentioning the name of the element tag, such as paragraph <p>, image <img>, header <h1>, and so on.

* id

Using the id selector is to include a hashtag sign (#) before the element's name

* class

Using a selector class is to include a period (.) before the name of the element.

The jQuery selector syntax is usually created to select HTML elements and perform actions on the selected elements. Here is the basic syntax of the jQuery selector:

|  |
| --- |
| $(selector).action() |

* dollar sign ($), to define jQuery
* (selector), to indicate the selected element
* action(), is the jQuery action that will be performed on the selected element

To understand the use of tag, id, and class selectors follow the steps of the practicum below:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new HTML file named selector.html in the dasarWeb/praktik\_jquery directory, type the following program code into the file.    Changed with the last version |
| 2 | Save the file, then open a browser and run the program code by typing localhost/dasarWeb/praktik\_jquery/selector.html |
| 3 | After the program is run, click the "Click Me!" button, then observe the display on the browser. |
| 4 | Record and explain your observations after running the program code above. (Question No. 2)  When this code is loaded in the browser, the user will initially see a series of visible text including the Chapter Title, Sub Chapter Title, three distinct paragraphs, and one additional paragraph followed by a button labeled "Klik Saya!". Upon clicking the button, the embedded jQuery code will execute immediately. This code is instructed to hide all elements targeted, Sub Chapter Title, the first paragraph, the second paragraph, and the third paragraph will all disappear from view, while the button itself and the final paragraph ("Ini paragraf yang akan tetap ditampilkan") will remain visible. |
| 5 | In the program code in step number 1, there are some jQuery selector writings. Write down and explain what jQuery Selector is in your code. (Question No. 3)  jQuery Selectors are the fundamental tools used to locate and choose specific HTML elements on a webpage so they can be manipulated. They operate much like CSS selectors, all encased within the basic syntax of $(""). In the script, it use five distinct selectors, the Element Selector $("h2") to hide all H2 headings, the Class Selector $(".subjudul") to target any element with that specific class, the ID Selector $("#paragraf") to select the unique element with that ID, the Compound Selector $("#paragraf.dua") for a highly specific match requiring both that ID and class, and finally, the Descendant Selector $("div p.paragraf") to select only paragraph elements with the class 'paragraf' that are nested inside a <div> tag. These selectors tell jQuery precisely which content pieces to hide when the user clicks the button. |

**Practical Section 4. Events**

*An event* is something that the user can do to an HTML element. Examples *of events* are *click*, *double click*, *mouseover* (the mouse cursor is on top of the element), *mouseout* (the mouse cursor is out of the top of the element), and so on. The basic syntax of jQuery events is as follows:

|  |
| --- |
| $("jquery\_selector").jquery\_event(function() {  ... event out ... }); |

* dollar sign ($), to define jQuery
* ("jquery\_selector"), to indicate the selected element
* jquery\_event(function(), is an event that will be performed on the selected element

To understand the use of jQuery events follow these steps:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new HTML file named event.html in the dasarWeb/praktik\_jquery directory, then type the following program code:      Changed with the last version |
| 2 | Save the file, then open a browser and run the program code by typing localhost/dasarWeb/praktik\_jquery/event.html |
| 3 | Once the program is running, do the following:   * Place the mouse cursor over the element      * Move the mouse cursor out the element      * one-click on the element      * Double-click on the element     Then observe the changes that occur |
| 4 | Record and describe the changes that occurred based on your observations (Question No. 4)  Upon loading in the browser, the user will see a large box containing the text "Praktikum Minggu ke-4 JQuery" styled with a pink background and centered alignment. This element is set up with four distinct jQuery event handlers that trigger visual changes upon interaction, a single click will permanently change the text color to white, moving the mouse over the box will change the background color to silver, moving the mouse out of the box will change the background color to blue, and finally, a double-click will permanently add a solid 3px black border around the box, resulting in a dynamic element that changes color, text appearance, and border style based on user interaction. |

**Practical Section 5. Hide and Show Effect**

JQuery provides "ready-to-use" functions that can be used to give a variety of interesting visual effects to elements. The basic syntax of jQuery effects is:

|  |
| --- |
| $("jquery\_selector").jquery\_effect({parameter}, {value}) |

* dollar sign ($), to define jQuery
* ("jquery\_selector"), to indicate the selected element
* jquery\_effect({parameter}, {value}), is the effect that will be applied to the element. Effects can have certain parameters and values to set the effect to be given.

There are various effects that jQuery provides, including hide() to hide elements and show() to show elements. There are also fadeIn(), fadeOut(), and fadeTo() effects that are used to show or fade elements slowly, making the transitions look smoother. To understand jQuery Effect, follow these steps:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new HTML file named hideshow.html in the dasarWeb/praktik\_jquery directory, then type the following program code:    Changed with the last version |
| 2 | In the same directory, create a new file named style.css then type the following program code: |
| 3 | Save the file, then open a browser and run the program code by typing localhost/dasarWeb/praktik\_jquery/hideshow.html |
| 4 | After the program clicks the buttons in order, then observe the changes that occur.  **Sembunyikan**    **Tampilkan**    **Fade out**    **Fade in**    **Fade to** |
| 5 | Record and describe the changes that occurred based on your observations (Question No. 5)  When the buttons are clicked, they trigger different jQuery animations on these squares. The "Sembunyikan" button will immediately make all three squares disappear, while the "Tampilkan" button will instantly bring them back. The "Fade Out" button will make them disappear with varying speeds, Red instantly, Green slowly, and Blue over 3 seconds. The "Fade In" button reverses this, making them reappear at the same varying speeds. Finally, the "Fade To" button changes the opacity of the squares to 15%, 40%, and 70% respectively, making them semi-transparent instead of fully hidden or shown. |

**Practical Section 6. Slide Effect**

The slide effect that jQuery provides is used to remove or display elements as if they were opening/closing something. The basic syntax used is the same as the syntax for adding hide() or show() effects, but uses a different jQuery effect function.

There are 3 types of slide effects, namely slideUp, slideDown, and slideToogle. To understand the effect of slides, follow these practicum steps:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new HTML file named slide1.html in the dasarWeb/praktik\_jquery directory, then type the following program code: |
| 2 | In the same directory, create a new file named styleSlide.css and type the following program code: |
| 3 | Save the file, then open a browser and run the program code by typing localhost/dasarWeb/praktik\_jquery/slide1.html |
| 4 | Once the program is running, click on the "Click for Slide Effect" panel, then observe the changes that occur |
| 5 | Record and explain the changes that occurred based on your observations (Question No. 6)  When this page loads, users will see two boxes with the same style, with turquoise background and a light gray border, the first box labeled Click for Slide Effect (#flip) will appear thin, while the second box labeled “Good Morning” (#box2) will appear much larger What happens is that when the user clicks on the “Click for Slide Effect” box (#flip), the large div containing the text “Good Morning” (#kotak2) will slowly disappear upwards using the slide animation effect (slideUp(“slow”)). |
| 6 | Create a new HTML file file named slide2.html, copy the program code in step 1 and modify it by changing a few lines of program code, so that the program code looks like this:    Changed with the last version |
| 7 | Save the file, then open a browser and run the program code by typing localhost/dasarWeb/praktik\_jquery/slide2.html |
| 8 | Once the program is running, click on the "Click for Slide Effect" panel, then observe the changes that occur. |
| 9 | Record and describe the changes that occur based on your observations (Question No. 7)  When the page loads in the browser, only the box labeled “Click for Slide Effect” (#flip) will be visible, while the larger box labeled “Good Morning” (#kotak2) is initially hidden due to the attribute style=“display:none;”. The change occurs when the user clicks on the visible box (#flip), at that moment, the jQuery code will execute the .slideDown(“slow”) function, causing the “Good Morning” box (#kotak2) to appear slowly with a slide animation from top to bottom. |
| 10 | Create a new HTML file named slide3.html, copy the program code in step 1 and modify it by changing a few lines of program code, so that the program code looks like this:    Changed with the last version |
| 11 | Save the file, then open a browser and run the program code by typing localhost/dasarWeb/praktik\_jquery/slide3.html |
| 12 | Once the program is running, click on the "Click for Slide Effect" panel, then observe the changes that occur. |
| 13  ­­ | Record and describe the changes that occurred based on your observations (Question No. 8)  When the page loads in the browser, both boxes will be visible since there is no code explicitly hiding the #kotak2 element by default. The smaller box, **"**Klik untuk Efek Slide" (#flip), acts as the trigger. The observable change that occurs repeatedly is that when the user clicks #flip, the jQuery function .slideToggle("slow") executes, causing the larger box, "Selamat Pagi" (#kotak2), to slowly hide with a sliding-up animation if it is currently visible, and then slowly appear with a sliding-down animation if it is currently hidden. This ensures that every subsequent click alternately reveals and conceals the large box. |

**Practical Section 7. Animation**

Animation effects are used to move an element. The animate() method can be used to create animation effects. The jQuery animate() syntax is as follows:

|  |
| --- |
| $("jquery\_selector").animate({parameter},{value}) |

* dollar sign ($), to define jQuery
* ("jquery\_selector"), to indicate the selected element
* Method animate({parameter},{value}), has parameters and values. Parameters define the properties of the element to be animated, e.g. "top", "left". Value defines the value of a parameter, for example "fast", "slow", or it can be in milliseconds (5000 = 5 seconds).

**Method Animate()**

To understand the use of the animate() method in jQuery, follow these steps:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new HTML file named animate.html in the dasarWeb/praktik\_jquery directory, then type the following program code:    Changed with the last version |
| 2 | In the same directory, create a new file named styleAnimate.css and then type the following program code: |
| 3 | Save the file, then open a browser and run the program code by typing localhost/dasarWeb/praktik\_jquery/animate.html |
| 4 | Once the program is running, click the "Start Animation" button, then observe the changes that occur. |
| 5 | Record and describe the changes that occurred based on your observations (Question No. 9)  This final combined code will initially display a button labeled "Start Animation" followed by a by purple box positioned below the button. The box is styled with a solid border and critically has position: relative; set in the CSS, which is necessary for jQuery's animation property to work. The observable change is triggered when the user clicks the "Start Animation" button, which executes the jQuery .animate({left: 300}) function, causing the purple box to smoothly slide to the right from its starting position. Since the animation is only set to move to the right, clicking the button multiple times will not extend the movement beyond this final position. |

**Method Chaining**

Chaining inside jQuery makes it possible to assign multiple animation methods into a single element using a single *statement*. Follow these steps to understand method chaining:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new HTML file named chaining.html in the dasarWeb/praktik\_jquery directory, and then type the following HTML code:    Changed with the last version |
| 2 | Save the file, then open a browser and run the program code by typing localhost/dasarWeb/praktik\_jquery/chaining.html |
| 3 | Once the program is running, click the "Start Animation" button, then observe the changes that occur. |
| 4 | Record and explain the changes that occur based on your observations (Question No. 10) Upon loading the page, the user will see a "Start Animation" button followed by the visible element targeted by the class .box. When the user clicks the button, the jQuery code executes a sequence of five distinct animation steps on the box, leveraging the chaining and queueing features of the .animate() function, which ensures each animation runs only after the previous one is fully completed. Specifically, the box will first expand horizontally to width, then expand vertically to height, followed by sliding to the right (due to marginLeft), then its border thickness will increase too, and finally its opacity will reduce to (half transparent). |

**Practical Section 8. JQuery DOM Manipulation**

JQuery provides methods for efficiently manipulating the DOM (*Document Object Model*). Here are some of the methods used to manipulate the DOM:

* text(), sets or returns thetext of the selected element
* html(), set or *return* the content of the selected element while retaining the HTML tags present in the element
* val(), set or *return* the value of the form

**Fetching (GET) Content from HTML Elements**

Practicum steps:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new HTML named get.html in the dasarWeb/praktik\_jquery directory, then type the following program code:    Changed with the last version |
| 2 | Save the file, then open a browser and run the program code by typing localhost/dasarWeb/praktik\_jquery/get.html |
| 3 | Once the program is running, click the buttons in order, then observe the changes.  **Get text**    **Get HTML**    **Get value** |
| 4 | Record your observations and explain the differences between the get text(), html(), and val() methods (Question No. 11)  These three jQuery methods are used to retrieve content from selected HTML elements, but they differ based on the content type and the element being targeted. The .text() method retrieves the plain, rendered text content, ignoring all HTML tags within the element, as demonstrated by the "Get Text" button only alerting "This is the first paragraph." Conversely, the .html() method retrieves the full content, including any raw HTML tags, which is why the "Get HTML" button alerts "This is <b> another </b> paragraph." Finally, the .val() method is specifically designed for form elements (like <input>) and retrieves the user-entered value currently inside that field, as demonstrated by the "Get Value" button alerting whatever text the user has typed. |

**Modifying (SET) Content from HTML Elements**

Practicum steps:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new HTML file named set.html in the dasarWeb/praktik\_jquery directory, then type the following program code:    Changed with the last version |
| 2 | Save the file, then open a browser and run the program code by typing localhost/dasarWeb/praktik\_jquery/set.html |
| 3 | Once the program is running, click the buttons in order, then observe the program code as it runs.  **Get text**    **Get HTML**    **Get value** |
| 4 | Record your observations and explain the differences between the set text(), html(), and val() methods (Question No. 12)  The set methods .text(), .html(), and .val() are used to change the content of elements when their respective buttons are clicked, and their behavior differs significantly based on how they handle HTML tags. The .text("<b>Hello World</b>") method, triggered by button 1, will treat the input as plain text and replace the content of #test1 with the literal string " ⟨b⟩Hello World⟨/b⟩", displaying the raw tags without formatting. The .html("<b>Hello World!</b>") method, triggered by button 2, will interpret the input as HTML markup, replacing the content of #test2 with bold, formatted text "Hello World!". Finally, the .val("Polinema") method, triggered by button 3 is used specifically to set the input value of form elements, causing the text field (#test3) to be pre-filled with the string "Polinema". |

**Adding and Removing Content from HTML Elements**

JQuery provides methods for adding and removing HTML elements. Some of the methods that can be used are append() and remove(). Here are the practicum steps to understand these methods:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a file named styleTable.css in the dasarWeb/praktik\_jquery directory, then type the following code: |
| 2 | Create a new HTML file named addremove.html in the dasarWeb/praktik\_jquery directory, then type the following program code:    Changed with the last version |
| 3 | Save the file, then open a browser and run the program code by typing localhost/dasarWeb/praktik\_jquery/addremove.html |
| 4 | Once the program is running, do the following:   * Fill out the form with your name and email address * Click "Add Row"      * Mark the row you added      * Click "delete row"     Then observe the changes |
| 5 | Record and explain your observations (Question No. 13)  This jQuery script creates a dynamic table management interface that allows users to add and delete rows without reloading the page. When the "Add Row" button is clicked, the code reads the current Name and Email values from the input fields (#name and #email), constructs a new table row (<tr>) containing a checkbox, the name, and the email, and appends this new row to the table's body (<table> <tbody>) using the .append() method. Conversely, when the "Delete Row" button is clicked, the script iterates through all checkboxes named 'record' inside the table body; if a checkbox is found to be checked (.is(':checked')), it uses .parents("tr").remove() to quickly find the parent table row and permanently delete it. |

**Practical Section 9. JQuery CSS Manipulation**

JQuery also has the ability to manipulate CSS. There are several functions, namely:

* addClass(), adds one or more classes to the selected element
* removeClass(), removes one or more classes into the selected element
* css(), set and return the style attribute

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a file named styleCSS.css in the dasarWeb/praktik\_jquery directory, then type the following code: |
| 2 | Create a new HTML file named cssmanipulation.html in the dasarWeb/praktik\_jquery directory, then type the following program code:    Changed with the last version |
| 3 | Save the file, then open a browser and run the program code by typing localhost/dasarWeb/praktik\_jquery/cssmanipulation.html |
| 4 | Once the program is running, click the buttons in order, then observe the program code as it runs.  **Add classes to elements**    **Remove classes to elements**    **Set classes to elements** |
| 5 | Record and explain your observations (Question No. 14)  When the "Add classes to elements" button (with class .btn1) is clicked, it adds the class blue to all <h1>, <h2>, <p>, and <div> elements, making their text color blue (as defined in the CSS). It also specifically adds the class important to the <div> element, making its text bold and extra-large. When the "Remove classes to elements" button (with class .btn2) is clicked, it removes the blue class from all h1, h2, p, div elements, returning their text color to the default (usually black), but the <div> retains the important class. Finally, when the "Set classes to elements" button (with class .btn3) is clicked, it uses the jQuery .css() method to directly apply an inline style, setting the background-color to yellow and the font-size to 100% for all h1, h2, p, div elements. |

**Practical Section 10. Create a Slide Show**

In this practicum, a slideshow is made to display photos. Slideshows are created without using UI plug-ins, but use several jQuery functions including: appendTo(), fadeIn(), fadeOut(), delay()using method chaining which allows to pass multiple animation methods into a single element using a single statement.

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new folder named img inside dasarWeb/praktik\_jquery |
| 2 | Put three .jpg formatted images, rename them to "gambar1.jpg, gambar2.jpg, and gambar3.jpg", then place them in the dasarWeb/praktik\_jquery/img directory |
| 3 | Create a new CSS file named styleSlideShow.css then type the following code: |
| 4 | Create a new HTML file named slideshow.html in the dasarWeb/praktik\_jquery directory, then type the following program code:    Changed with the last version |
| 5 | Save the file, then open a browser and run the program code by typing localhost/dasarWeb/praktik\_jquery/slideshow.html |
| 6 | Observe the program code as it runs. |
| 7 | Record and explain your observations (Question No. 15)  Upon page load, all images and titles within the by #slider container are initially hidden. The showNextImage function is then called immediately and repeats every it increments a counter (i), selects the next image and title (#sliderImage1 and #title1), moves them to the end of the #slider element's children using .appendTo(), makes them smoothly fade in over, pauses visibility for using .delay(), and then makes them smoothly fade out over. Since the counter resets from 3 to 0, this process creates a seamless looping slideshow of the three images, with each title overlaid at the bottom of the slide using CSS positioning. |

**Practical Section 11. JQuery UI Plugin**

jQuery UI (*user interface*) is the most popular plugin among jQuery programmers. Because to the point that it is made into one manual with the jQuery manual on the official website. jQuery UI was created by Paul Bakaus who worked with Stefan Petre to create sophisticated effects and components of jQuery's complementary libraries, such as accordion and datepicker.

Follow these steps to perform the jQuery UI installation:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Download jQuery UI in <http://jqueryui.com/download/> choose version (stable) |
| 2 | The downloaded file is still a .zip file. to use it, extract the jQuery file (jquery-ui-1.12.1.zip) in the dasarWeb/praktik\_jquery/jquery-ui-1.14.0 |
| 3 | To connect the jQuery UI Plugin file with HTML, add the <script> tag with the address jquery-ui-1.14.0/jquery-ui.js. Create a new HTML file and then type in the code like the following example:    Changed with the last version |

**Datepicker**

Datepicker functions to retrieve dates from the calendar system on the computer, making it easier for us to choose dates, because they are displayed in their entirety with an attractive appearance. To use the datepicker, follow these steps:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new HTML file named date.html in the dasarWeb/praktik\_jquery directory, and type the following code:    Changed with the last version |
| 2 | Save the file, then open a browser and run the program code by typing localhost/dasarWeb/praktik\_jquery/date.html |
| 3 | Record and explain your observations (Question No. 15)  This simple HTML and JavaScript code utilizes the jQuery UI Datepicker widget to enhance a standard text input field, turning it into an interactive calendar selector. Upon the page loading, the jQuery code executes the function $("#date\_ex").datepicker(), which attaches the datepicker functionality to the input field with the ID date\_ex. The result is that when a user clicks on the "Selected Date" text input box, a calendar pop-up will appear, allowing the user to easily select a date by clicking on it, which then populates the text box with the chosen value. |

**Accordion**

Accordion functions to group content in separate panels. Web visitors can open and close the desired panels. Follow these steps:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a new HTML file named accordion.html in the dasarWeb/praktik\_jquery and type the following code:    Changed with the last version |
| 2 | Save the file, then open a browser and run the program code by typing localhost/dasarWeb/praktik\_jquery/accordion.html  **Header 1**    **Header 2**    **Header 3** |
| 3 | Record and explain your observations (Question No. 16)  This program is using jQuery UI Accordion to create collapsible sections on a webpage. When the code runs, it loads the jQuery and jQuery UI libraries, then initializes an accordion on the <div> with the ID jQuery\_accordion. Inside the accordion, there are three sections, the first shows a simple paragraph, the second contains text and an image, and the third includes text with an ordered list. Each section is hidden or shown when its header (like "header 1", "header 2", "header 3") is clicked, giving the page an interactive and organized layout. |

**Introduction to AJAX**

**AJAX** stands for *Asynchronous Javascript and Xml*. AJAX is not a programming language, but it is a technique for creating better, faster and more interactive web applications. With AJAX, JavaScript can load data from the server into a web browser without reloading the entire page. What AJAX does is use the JavaScript-based XMLHttpRequest object to send and receive Description to and from a web server.

**XMLHttpRequest**

XMLHttpRequest is a JavaScript object. Here is the code used to create the XMLHttpRequest object:

|  |
| --- |
| <script language=”javascript” type=”text/javascript”>  var xmlHttp = new XMLHttpRequest();  </script> |

To get and send data from/to a database or file on a server using traditional javascript, you need to create an HTML Form. And the user has to click the "submit" button to send/get Description, wait for a response from the server, then a new page in the form of results will be loaded. Since the server always gives a new page every time the user presses the submit button, a simple web application will run slowly and will be less user-friendly.

With Ajax, javascript will communicate directly with the server through the javascript object that is the XMLHttpRequest. With the XMLHttpRequest object, a web page can make a request to, and get a response from, the web server without reloading the entire page. The user will stay on the same page. Even the user will not know if there is data sent and received from the server, because javascript performs data transactions behind the scenes. Requests are sent *asynchronously*, which means that JavaScript code (and the user) doesn't wait on the server to respond. So that users can continue to enter data and use the application. The following image shows a comparison of a traditional web application and a web application that uses AJAX.

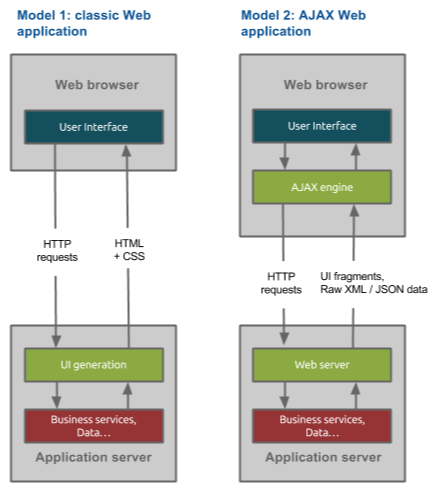


Figure 1. Comparison of traditional and AJAX web applications

**Practical Section 12. JQuery AJAX**

Different browsers implement AJAX differently. If you adopt the JavaSricpt way, different code is required for different browsers. For this reason, JQuery provides a solution to simplify the implementation of AJAX so that it can be used across browsers.

**JQuery Load()**

One of the functions of jQuery is load(). The load() method is used to load data from the server and place the returned HTML to the selected element. The basic syntax of the load() method is:

|  |
| --- |
| $(selector).load(URL, data, complete); |

* URL, is the address or name of the file on the server to be retrieved
* Data, optional, is a key-value pair that will be sent to the server
* Complete, optional, is a function that is executed when data is retrieved.

To understand the load() function, follow these steps:

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Put a .jpg formatted image, rename it to "gambar.jpg", then place the image in the dasarWeb/praktik\_jquery/img |
| 2 | Create a new HTML file named test-content.html then type the following code: |
| 4 | Create a new HTML file named home.html and then type the following code:  Changed with the last version |
| 5 | Save the file, then open a browser and run the program code by typing localhost/dasarWeb/praktik\_jquery/home.html |
| 6 | Once the program is running, click the "Click Me!" button, then observe the program code run. |
| 7 | Record and explain your observations (Question No. 17)  This jQuery AJAX is designed to load and replace content from an external file upon a button click using the .load() method. When the home.html page loads, the user initially sees the header "Klik tombol dibawah untuk menampilkan konten" inside the div with ID hint, followed by the "Klik Saya!" button. When the user clicks the button, the jQuery script attempts to fetch the content of test-content.html using the specific relative path "test-content.html". If the AJAX request is successful, the entire content of test-content.html (which includes the <h1>, the paragraph "Tutorial Fungsi Load()", and the image) will completely replace the original content inside the <div id="hint"> element. |

**Github link :** [**https://github.com/Lovie-Tonimba/semester3-PemrogramanWeb.git**](https://github.com/Lovie-Tonimba/semester3-PemrogramanWeb.git)

**Reference:**

1. Duckett, John. 2014. Javascript & JQuery: Interactive Front-end Web Development. John Wiley & Sons, Inc: Indiana, USA.
2. Chaffer, J & Swedberg, K. 2013. Learning jQuery 4th Edition: Better Interaction, Design, and, Web Development with Simple JavaScript Techniques. Birmingham: Packt Publishing Ltd.