

Laporan Teknologi Web

Geometry Calculator



By :

Putu Ananda Adi Savitri

2215101057

Fakultas Teknik dan Kejuruan

Jurusan Teknik Informatika

Prodi Ilmu Komputer

Universitas Pendidikan Ganesha

Singaraja

2023

## CHAPTER I

### INTRODUCTION

#### A. Basic Theory

Geometry (from Ancient Greek γεωμετρία (geōmetría) "measurement of the earth"; from γῆ (gê) "earth, earth" and μέτρον (metron) "a measurement") is a branch of mathematics related to the properties of space such as distance, shape, size and relative position of shapes. Geometry, along with arithmetic, is one of the oldest branches of mathematics. A mathematician who works in the field of geometry is called a geometer. Until the 19th century, geometry was almost exclusively Euclidean geometry, which included the concepts of points, lines, planes, distances, angles, surfaces, and curves as fundamental concepts.

Originally developed to model the physical world, geometry has applications in almost every science as well as art, architecture, and other graphics-related activities. Geometry also has applications in fields that seem unrelated to mathematics. For example, methods of algebraic geometry were the basis for Wiles's proof of Fermat's Last Theorem, a problem formulated in fundamental arithmetic terms and which remained unsolved for centuries.

In this report, the objectives to be achieved include:

- a. Understand about basic web.
- b. Understand about Geometry system.

## CHAPTER II

### MAIN CONTENT

HTML (Hypertext Markup Language), CSS (Cascading Style Sheets) and JavaScript are the basic technologies used in web development to create and design websites with interactive features, including Each type serves a different purpose. HTML is the backbone of web content and is used to structure the content of a website. It identifies elements and their relationships on a web page. CSS is used to control the presentation and style of HTML elements. It determines how web content is displayed, including layout, fonts, colors, and spacing. JavaScript is a programming language used to add interactivity and behavior to web pages. It allows you to create dynamic content, respond to user actions, and manipulate your website's Document Object Model (DOM). Here In the context of a geometric calculator, HTML provides the structure and layout of the user interface, allowing users to enter values and view results, CSS can be applied to make the calculator to be visually appealing and user-friendly, and JavaScript JavaScript does the calculations. based on user input, such as calculating the volume of a cube, cube, sphere, etc.

#### a) *HTML*

The provided code is an HTML document that defines the structure and content of a web page for a geometry calculator. Here I'll break down the code step by step:

1. `<!DOCTYPE html>`: This declaration specifies the document type and version of HTML being used, which is HTML5 in this case.
2. `<html lang="en">`: The `<html>` element is the root element of an HTML document. The `lang="en"` attribute indicates that the primary language of the document is English.
3. `<head>`: The `<head>` section of the HTML document contains metadata and links to external resources.
  - i. `<meta charset="UTF-8">`: This specifies the character encoding for the document as UTF-8, which is a widely used character encoding for handling various characters and symbols.
  - ii. `<meta name="viewport" content="width=device-width, initial-scale=1.0">`: This meta tag is commonly used for responsive web design. It sets the viewport width to the device's width and sets the initial zoom level to 1.0.
  - iii. `<title>Kalkulator Volume Bangun Ruang</title>`: This sets the title of the web page, which appears in the browser's title bar or tab.

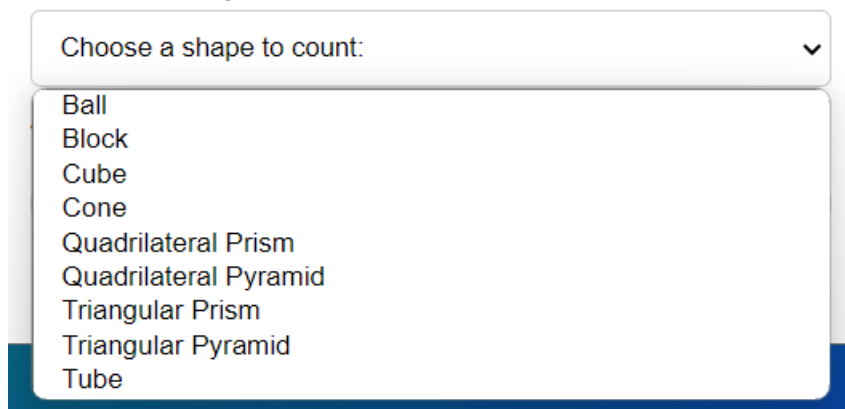
A screenshot of a browser window's title bar. It is dark gray with a light gray circular icon on the left, followed by the text 'Kalkulator Volume Bangun Ruang' in white, and a small 'X' icon on the right to close the window.
  - iv. `<link rel="stylesheet" href="style.css">`: This links an external CSS stylesheet named "style.css" to the HTML document. This stylesheet likely contains styles (e.g., for layout, fonts, colors) to be applied to the page's elements.

4. `<body>`: The `<body>` section contains the visible content of the web page.
- `<div class="container">`: This is a container `div` that wraps all the page content. It's often used for layout and styling purposes.
  - `<div class="Volume">`: Inside the container, there's a `div` with the class "Volume," which appears to be a section or block of content related to volume calculations for geometric shapes.
  - `<h1>Geometry calculator</h1>`: This is a heading displaying "Geometry calculator."

## Geometry calculator

- `<div class="form-container">`: This `div` with the class "form-container" likely contains the form and input elements for selecting a geometric shape and calculating its volume.
- `<label for="shape">Choose a shape:</label>`: This label is associated with a `select` element using the `for` attribute. It tells the user what the `select` element is for.
- `<select id="shape">`: This is a `select` element with the id "shape." It provides options for choosing different geometric shapes, such as cubes, blocks, spheres, etc.
- `<option value="" disabled selected hidden>Choose a shape to count:</option>`: This is the default option, which is disabled, selected, and hidden. It serves as a placeholder instructing the user to choose a shape.
- Various `<option>` elements represent different shapes that can be selected.

### Choose a shape:



- `<button onclick="calculateVolume()">Calculate</button>`: This is a button element with the label "Calculate." When clicked, it triggers the JavaScript function `calculateVolume()`, which should be defined in the linked "script.js" file.

Calculate Volume

- x. `<div id="dimension-inputs"></div>`: This empty `div` may be used to display input fields specific to the selected shape. These input fields would likely appear dynamically based on the user's choice. Here I use 'Cone' as example.

**Choose a shape:**

**Radius (cm):**

**Height (cm):**

- xi. `<label for="answer">The Answer: </label>`: Similar to the previous label, this one is associated with an element with the id "answer."
- xii. `<div id="result"></div>`: This empty `div` with the id "result" is likely where the calculated volume will be displayed. The content will be inserted into this `div` dynamically via JavaScript.

**Choose a shape:**

**Radius (cm):**

**Height (cm):**

**Calculate Volume**

**The Answer:**

The Volume is: 209.44 cm<sup>3</sup>

5. `<script src="script.js"></script>`: This line includes an external JavaScript file named "script.js." This JavaScript file likely contains the code responsible for calculating the volume of selected geometric shapes and updating the "result" `div` with the result.

## b) CSS

The provided CSS code defines styles for various elements within a web page. Here I'll break down the code step by step:

1. ``body``:
  - i. ``font-family``: This property sets the font family for text within the entire web page. It specifies multiple font choices in case the user's system doesn't have the first font listed. The fonts listed are in order of preference.
  - ii. ``background-color``: Sets the background color of the entire page to a light grayish color (`#f4f4f4`).
  - iii. ``box-sizing``: It ensures that when you set the width or height of an element, it includes padding and borders in the calculation, making it easier to control the layout.
  - iv. ``margin`` and ``padding``: These properties set the margin and padding of the ``body`` element to 0, removing any default spacing around the page content.
2. ``container``:
  - i. ``width``: Makes the container span the entire width of its parent element.
  - ii. ``min-height``: Sets the minimum height of the container to 100% of the viewport height (`100vh``), ensuring it at least covers the entire viewport's height.
  - iii. ``background``: Sets a linear gradient background for the container, going from a greenish color (`#00a138`) to a deep blue color (`#1200ff`).
  - iv. ``padding``: Adds 10 pixels of padding around the container.
3. ``Volume``:
  - i. ``max-width``: Limits the maximum width of the ``Volume`` element to 400 pixels.
  - ii. ``margin``: Centers the ``Volume`` element horizontally within its parent.
  - iii. ``background-color``: Sets the background color of the ``Volume`` element to white.
  - iv. ``border``: Adds a 1-pixel solid border around the ``Volume`` element with a dark gray color (`#6b6b6b`).
  - v. ``border-radius``: Rounds the corners of the ``Volume`` element with a 10-pixel border radius.
  - vi. ``padding``: Adds 30 pixels of padding inside the ``Volume`` element.
  - vii. ``text-align``: Centers the text content within the ``Volume`` element.
4. ``h1``:
  - i. ``font-size``: Sets the font size of all `<h1>` elements to 23 pixels.
  - ii. ``color``: Sets the text color of all `<h1>` elements to a dark red color (`#a50000`).
5. ``form-container``:
  - i. ``margin-bottom``: Adds 20 pixels of margin at the bottom of elements with the ``form-container`` class.
  - ii. ``text-align``: Aligns text within elements with the ``form-container`` class to the left.

6. ``label``:
  - i. ``display``: Makes ``<label>`` elements block-level, so they start on a new line and have some spacing around them.
  - ii. ``font-weight``: Sets the font weight of ``<label>`` elements to bold.
  - iii. ``margin-bottom``: Adds 5 pixels of margin at the bottom of ``<label>`` elements.
7. ``select`` and ``input``:
  - i. ``width``: Sets the width of ``<select>`` and ``<input>`` elements to 100% of their parent container.
  - ii. ``padding``, ``border``, ``border-radius``, and ``margin-bottom``: Define consistent styling for both ``<select>`` and ``<input>`` elements, including padding, border, border-radius, and margin at the bottom.
8. ``#dimension-inputs``:
  - i. ``display``: Makes elements with the ID ``dimension-inputs`` block-level.
  - ii. ``font-weight``: Sets the font weight of elements with this ID to bold.
  - iii. ``margin-bottom`` and ``margin-right``: Adds margins at the bottom and right of elements with this ID.
9. ``button``:
  - i. ``padding``: Sets the padding of ``<button>`` elements to 10 pixels vertically and 20 pixels horizontally.
  - ii. ``background-color``: Sets the background color of ``<button>`` elements to a deep blue color (#301798).
  - iii. ``color``: Sets the text color of ``<button>`` elements to white.
  - iv. ``border``, ``border-radius``, and ``cursor``: Define border styling, border radius, and cursor behavior for ``<button>`` elements.
  - v. ``transition``: Specifies a smooth transition for the background color when hovering over the button.
  - vi. ``margin-bottom`` and ``margin-top``: Adds margins at the bottom and top of ``<button>`` elements.
10. ``button:hover``: This is a CSS pseudo-class that defines styles for a button when hovered over. It changes the background color to a darker blue color (#112167).
11. ``#result``:
  - i. ``max-width``: Limits the maximum width of the element with the ID ``result`` to 400 pixels.
  - ii. ``margin``: Centers the ``#result`` element horizontally within its parent.
  - iii. ``background``: Sets the background color of the ``#result`` element to a light gray (#d9d9d9).

- iv. ``border-radius``: Rounds the corners of the ``#result`` element with a 20-pixel border radius.
- v. ``padding``: Adds padding to the ``#result`` element for spacing.
- vi. ``text-align``: Centers the text content within the ``#result`` element.
- vii. ``font-weight``: Sets the font weight of text within the ``#result`` element to 15 pixels (which is unusual; font weight is typically set with words like "bold").

### c) *JavaScript*

The provided JavaScript code is responsible for handling the logic of a geometry calculator web application. It manages user interactions, dynamically generates input fields based on the selected shape, and calculates the volume of various geometric shapes. Here I'll break down the code step by step:

1. ``calculateVolume()`` Function:
  - i. This function is triggered when the user clicks the "Calculate" button on the web page.
  - ii. It retrieves several elements by their IDs:
  - iii. ``shapeSelect``: The ``select`` element with the ID "shape," which allows the user to choose a geometric shape.
  - iv. ``dimensionInputs``: A container element with the ID "dimension-inputs" where input fields for shape dimensions will be dynamically generated.
  - v. ``resultElement``: An element with the ID "result" where the calculated volume will be displayed.
  - vi. It clears the content of the ``dimensionInputs`` container using ``dimensionInputs.innerHTML = ""``;
  - vii. It then retrieves the value of the selected shape from the ``shapeSelect`` element using ``shapeSelect.value``.
  - viii. Depending on the selected shape, it dynamically generates and inserts input fields into the ``dimensionInputs`` container. For each shape, it checks which shape is selected and inserts the appropriate HTML input fields. For example, if "kubus" (cube) is selected, it inserts an input field for side length.
  - ix. Finally, it creates a "Calculate Volume" button using ``document.createElement("button")``. When this button is clicked, it calls the ``calculate(selectedShape)`` function to calculate the volume of the selected shape and displays the result in the ``resultElement``.
2. ``calculate(selectedShape)`` Function:
  - i. This function takes the ``selectedShape`` as an argument and calculates the volume based on the selected shape.
  - ii. It uses a series of conditional statements (if-else if) to check which shape is selected and calculate the volume accordingly.



- iii. For each shape, it retrieves the input values (e.g., side length, radius, height) from the corresponding input fields and performs the volume calculation.
  - iv. The calculated volume is returned as a result.
3. Event Listener for DOMContentLoaded: This event listener waits for the DOM (Document Object Model) to be fully loaded before it executes. This ensures that the JavaScript code won't run until the HTML document is ready.
  4. Event Listener for "change" on the Shape Selector: It listens for changes in the selected shape using `addEventListener("change", calculateVolume)`. When the user selects a different shape from the dropdown, it triggers the `calculateVolume()` function to update the input fields accordingly.

## CHAPTER III

### CLOSING

#### A. Conclusion

In summary, this HTML code defines a web page for a geometry calculator. Users can choose a geometric shape, enter dimensions if necessary, and then click the "Calculate" button to get the volume calculation displayed on the page. The appearance and functionality are likely further defined and controlled by the linked CSS and JavaScript files ("style.css" and "script.js"). The CSS code defines the styles and layout for elements within a web page, including fonts, colors, backgrounds, spacing, and alignment. These styles are applied to create a visually appealing and responsive user interface for a geometry calculator. And the JavaScript code provides the functionality for a geometry calculator web application. It dynamically generates input fields based on the selected shape, calculates the volume for various geometric shapes, and displays the result when the user clicks the "Calculate" button. It also updates the input fields when the user changes the selected shape from the dropdown.