

**DESIGN AND IMPLEMENTATION OF A SIMPLE WEBSITE USING HTML  
AND CSS**

**PERANCANGAN DAN IMPLEMENTASI WEBSITE SEDERHANA  
MENGUNAKAN HTML DAN CSS**

**Muh. Arfah Wahlil Pratama<sup>1</sup>, Wahda Mutmainnah<sup>2</sup>, Khusnul Khatimah<sup>3</sup>,  
Yudhika Setiawan<sup>4</sup>, Aril Adila Yusuf<sup>5</sup>, Muh Afriansyah<sup>6</sup>, Andi Nurfadillah<sup>7</sup>**

Universitas Muhammadiyah Kolaka Utara, Lasusua, Indonesia <sup>1,2,3,4,5,6,7</sup>

muharfahwahlilpratama@gmail.com<sup>1</sup>, wahda.mumainnah05@gmail.com<sup>2</sup>,  
nuramirahusril@gmail.com<sup>3</sup>, yudikasetiawan87@gmail.com<sup>4</sup>, elharamlah79@gmail.com<sup>5</sup>,  
ariladilayusuf@gmail.com<sup>6</sup>, muh.afriansa30@gmail.com<sup>7</sup>

**ABSTRACT**

*This study aims to demonstrate the design and implementation of a simple static website using HTML (HyperText Markup Language) and CSS (Cascading Style Sheets) without additional frameworks. In the digital era, the ability to create basic websites has become essential, especially for beginners in web development. This study addresses this need by focusing on content structure using HTML and visual presentation using CSS. The development method employed is the waterfall model, which systematically includes the stages of requirements analysis, design, implementation, and testing. The tools used include Visual Studio Code as the text editor and Google Chrome as the web browser for testing. The results demonstrate the successful development of a simple profile website consisting of a navigation header, main content section, and footer, utilizing HTML5 semantic elements and CSS styling to achieve a responsive and visually appealing layout. The practical implementation emphasizes simplicity and maintainability, with HTML code handling the structural foundation and CSS managing visual presentation, resulting in a functional and easily maintainable website. The study confirms that pure HTML and CSS are sufficient for developing static website projects, providing a solid foundation for beginners. Recommendations for future development include incorporating JavaScript for enhanced interactivity and dynamic functionality. This study provides a practical guide for students and novice developers in understanding fundamental web development concepts.*

**Keywords:** HTML, CSS, Web Design, Static Website, Web Development.

**ABSTRAK**

Penelitian ini bertujuan untuk mendemonstrasikan perancangan dan implementasi website sederhana dan statis menggunakan HTML (HyperText Markup Language) dan CSS (Cascading Style Sheets) tanpa bantuan framework tambahan. Dalam era digital saat ini, kemampuan membuat website dasar menjadi penting, terutama bagi pemula dalam pengembangan web. Penelitian ini menjawab kebutuhan tersebut dengan fokus pada struktur konten melalui HTML dan pengaturan tampilan melalui CSS. Metode pengembangan yang digunakan adalah model waterfall yang mencakup tahapan analisis kebutuhan, perancangan, implementasi, dan pengujian secara sistematis. Perangkat yang digunakan meliputi Visual Studio Code sebagai text editor dan Google Chrome sebagai web browser untuk pengujian. Hasil penelitian menunjukkan keberhasilan pembuatan website profil sederhana yang terdiri dari header navigasi, bagian konten utama, dan footer, menggunakan elemen semantik HTML5 dan styling CSS untuk layout yang responsif dan menarik secara visual. Implementasi praktis menekankan kesederhanaan dan kemudahan pemeliharaan, dengan kode HTML untuk struktur dasar dan CSS untuk presentasi visual, menghasilkan website yang fungsional dan mudah dipelihara. Kesimpulan penelitian menegaskan bahwa HTML dan CSS murni sudah memadai untuk pengembangan proyek website statis, memberikan fondasi yang kuat bagi pemula. Saran untuk pengembangan lanjutan mencakup penambahan JavaScript untuk meningkatkan interaktivitas dan fungsionalitas dinamis. Penelitian ini memberikan panduan praktis bagi mahasiswa dan pengembang pemula dalam memahami konsep dasar pengembangan web.

**Kata Kunci:** HTML, CSS, Perancangan Web, Website Statis, Pengembangan Web.

*This is an open access article distributed under the terms of the Creative Commons  
Attribution 4.0 International License (CC BY 4.0).*

Artikel ini adalah artikel akses terbuka yang didistribusikan di bawah ketentuan  
Lisensi Creative Commons Attribution 4.0 International (CC BY 4.0).



## INTRODUCTION

In today's digital era, websites play an important role as media for communication and information delivery. Websites enable individuals, institutions, and business actors to reach and interact with audiences widely, even at the global level. According to Internet World Stats (2024), global internet users have reached more than 5 billion people, indicating the importance of digital presence across various sectors. Nevertheless, not everyone has adequate technical skills to develop highly complex websites. Therefore, designing a simple website using basic web development languages such as HTML (HyperText Markup Language) and CSS (Cascading Style Sheets) becomes a practical and easy-to-learn alternative.

HTML functions to organize the structure and content of web pages, while CSS is used to manage the appearance and visual layout. Based on Robbins (2018), the combination of HTML and CSS is an important foundation in web development that must be mastered before learning more complex technologies. This approach enables the creation of static websites without requiring server-side programming or additional frameworks, making it highly suitable for beginners and small-scale website development.

This study aims to explain and demonstrate the stages of designing a simple website, such as a group profile page, by focusing on the use of pure HTML and CSS. This approach was chosen because many current online learning resources tend to directly use frameworks such as Bootstrap, which may hinder beginners' understanding of fundamental concepts. Duckett (2011) and MDN Web Docs (2023) emphasize the importance of a deep understanding of HTML and CSS before adopting frameworks. By learning HTML and CSS fundamentally, users are expected to build a strong understanding as a basis for more advanced web development.

In addition, this study is expected to provide a clear overview of the application of basic structure and design in static website development. The results of this study are expected to serve as a learning reference for students or beginners who want to study basic web programming systematically, as well as encourage the improvement of digital skills relevant to current information technology needs.

Based on this background, the research problems in this study are: how is the process of designing and implementing a simple website using HTML and CSS, how is HTML applied in building the structure of a static website, and how is CSS applied in managing the appearance and responsive layout of a simple website. Therefore, the objectives of this study are to describe the stages of designing and implementing a simple website using HTML and CSS, explain the application of HTML as the structural foundation of a static website, and explain the application of CSS in managing visual appearance and responsive website layout.

A website is a collection of interconnected web pages that can be accessed through the internet using a web browser. According to Tanenbaum and Wetherall (2011), a website functions as the main interface in the World Wide Web that facilitates global information exchange. Websites serve as media for information delivery, communication, and interaction between users and website owners.

HTML (HyperText Markup Language) is the standard markup language used to create and structure web pages. Developed by Tim Berners-Lee in 1991, HTML has become the main foundation of modern web development (Berners-Lee, 1999). HTML uses elements and tags to display text, images, links, tables, and other components on website pages. The latest version, HTML5, provides semantic elements such as `<header>`, `<nav>`, `<section>`, and `<footer>`, which improve website structure and accessibility (W3C, 2023).

CSS (Cascading Style Sheets) is a language used to manage the appearance and layout of web pages. Introduced by W3C in 1996, CSS separates content from presentation, enabling developers to control visual aspects of websites efficiently (Meyer, 2006). CSS allows developers to manage colors, fonts, text size, margins, padding, positioning, and responsive layouts, making websites more attractive and consistent across various devices and screen

sizes (Robbins, 2018).

## METHODS

The method used in this study is the Waterfall model (Sommerville, 2016), which is a systematic and sequential software development approach. This model was chosen because it is suitable for projects with clear specifications and does not require complex iterations.

The stages in the Waterfall model include:

1. Requirements Analysis: Identifying the functional requirements of the website, namely displaying a group profile with navigation information, main content, and footer.
2. Design: Designing the page structure using HTML and the visual design using CSS, including layout, color scheme, and typography.
3. Implementation: Writing HTML code for content structure and CSS code for visual styling according to the prepared design.
4. Testing: Testing the website's appearance and functionality on various web browsers to ensure compatibility and display consistency.

The software used in this study includes:

1. Windows 11 operating system.
2. Visual Studio Code version 1.85 as the text editor.
3. Google Chrome version 120 and Mozilla Firefox version 121 as web browsers for compatibility testing.

The hardware used was a laptop with an Intel Core i5 10th generation processor, 8 GB RAM, and 256 GB SSD storage.

The website design stages in this study follow the Waterfall model described previously. In detail, these stages include:

1. Requirements Analysis: Identifying the elements required in the group profile website, namely a header with navigation menu, home section with photo and group description, about section, contact section with email and WhatsApp information, and footer with copyright information.
2. HTML Structure Design: Designing the HTML document structure using HTML5 semantic elements such as <nav>, <section>, and <footer>. The structure was designed to facilitate navigation and code maintenance.
3. Visual Design Planning: Designing the website appearance using CSS, including the selection of color schemes, namely blue for the background and dark gray for the navbar, typography using Arial and Helvetica fonts, and responsive layout using flexbox and positioning techniques.
4. Code Implementation: Writing HTML code for content structure and CSS code for visual styling according to the design.
5. Testing and Validation: Conducting functional testing to ensure that all website elements are displayed correctly, as well as cross-browser compatibility testing on Google Chrome and Mozilla Firefox. Testing also includes HTML and CSS code validation using the W3C Validator to ensure compliance with web standards

## RESULTS AND DISCUSSION

The website designed in this study is a simple static website in the form of a student group profile page. The website display design was made simple with a navigation header, main content, and footer structure. The navigation header was designed with a fixed position

at the top of the page to make menu access easier. The selected color scheme is blue (#5ba0e4) for the main background and dark gray (#2c3e50) for the navbar, creating good contrast and readability. Each section was designed to be easily understood by users by applying simple yet effective design principles.

HTML implementation was carried out using HTML5 semantic elements such as <html>, <head>, <body>, <nav>, <section>, and <footer> to improve website structure and accessibility. HTML was used to display text, images, and the structure of the website page in an organized manner. The following explains each part of the HTML implementation:

- a. DOCTYPE Declaration and Meta Tags: Using <!DOCTYPE html> for HTML5 and meta charset UTF-8 to support Unicode characters.
- b. Navigation Bar: Using the <nav> element with <ul> and <li> lists for a well-structured navigation menu.
- c. Home Section: Displaying a profile photo using <img> and a group description within the <p> element.
- d. About and Contact Sections: Presenting additional information in separate sections for better content organization.
- e. Footer: Contains copyright information using the <footer> element as the closing section of the page.

The following is an example of HTML code implementation:

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <title>Profil Kelompok 4</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <!-- NAVBAR -->
  <nav>
    <ul>
      <li><a href="#beranda">Beranda</a></li>
      <li><a href="#tentang">Tentang</a></li>
      <li><a href="#kontak">Kontak</a></li>
    </ul>
  </nav>

  <!-- BERANDA -->
  <section id="beranda" class="container">
    
    <h1>Selamat Datang di Profil Kelompok 4</h1>
    <p>Kelompok 4 adalah mahasiswa yang tertarik dengan pengembangan web, khususnya HTML dan CSS untuk membangun website sederhana dan menarik.</p>
  </section>

  <!-- TENTANG -->
  <section id="tentang" class="container">
    <h2>Tentang Kelompok Kami</h2>
    <p>Kelompok kami merupakan mahasiswa yang sedang mempelajari pemrograman web dasar seperti HTML dan CSS untuk membuat website statis.</p>
```

```
</section>

<!-- KONTAK -->
<section id="kontak" class="container">
  <h2>Kontak</h2>
  <p>Email: yudikasetiawan87@gmail.com, ...</p>
  <p>WhatsApp: 0822-5929-9548, ...</p>
</section>

<!-- FOOTER -->
<footer>
  <p>© 2025 Profil Kelompok 4. Semua hak dilindungi.</p>
</footer>
</body>
</html>
```

CSS implementation was used to manage background color, typography, spacing, and page layout. CSS was applied separately through an external file named style.css, following the principle of separation of concerns to make code maintenance easier.

The following explains the CSS implementation:

- a. Reset and Base Styling: Setting the body margin to 0 to remove the browser's default margin, and setting the font-family to Arial/Helvetica with background-color #5ba0e4.
- b. Smooth Scrolling: Applying scroll-behavior: smooth to the html element for a smooth scrolling effect during navigation.
- c. Fixed Navbar: Using position: fixed for the navbar so that it remains visible while scrolling, with background-color #2c3e50 and z-index 1000.
- d. Flexbox Layout: Applying display: flex to the navigation menu for responsive alignment.
- e. Responsive Sections: Each section uses min-height: 100vh for a full viewport height display and proportional padding.

The following is an example of CSS code implementation:

```
/* RESET */
body {
  margin: 0;
  font-family: Arial, Helvetica, sans-serif;
  background-color: #5ba0e4;
  color: #333;
}

/* SMOOTH SCROLL */
html {
  scroll-behavior: smooth;
}

/* NAVBAR */
nav {
  background-color: #2c3e50;
  position: fixed;
  top: 0;
  width: 100%;
```

```
        z-index: 1000;
    }

    nav ul {
        list-style: none;
        display: flex;
        justify-content: center;
        margin: 0;
        padding: 15px 0;
    }

    nav ul li {
        margin: 0 20px;
    }

    nav ul li a {
        color: rgb(237, 95, 95);
        text-decoration: none;
        font-weight: bold;
    }

    nav ul li a:hover {
        text-decoration: underline;
    }

    /* SECTION */
    section {
        min-height: 100vh;
        padding: 100px 20px 40px;
        text-align: center;
    }

    /* PROFILE PHOTO */
    .profil {
        border-radius: 50%;
        width: 200px;
        height: 200px;
        object-fit: cover;
    }
```

System testing in this study was conducted to validate that the developed website met the specifications and functioned properly.

The testing methods used include:

1. Functional Testing: Verifying that all website elements, including header, navigation, content, and footer, are displayed correctly and that the navigation menu functions as expected.
2. Compatibility Testing or Cross-Browser Testing: Testing the website on Google Chrome version 120 and Mozilla Firefox version 121 to ensure display and functionality consistency across browsers.
3. Responsiveness Testing: Testing the website display on various screen sizes, namely desktop 1920×1080, laptop 1366×768, and tablet 768×1024, to ensure that the layout

remains optimal.

4. Code Validation: Using W3C HTML Validator and W3C CSS Validator to ensure that the HTML and CSS code comply with web standards.
5. Performance Testing: Measuring page loading time using Google Lighthouse to ensure optimal website performance.

Each testing process was carried out with systematic documentation by recording test results and making improvements if errors or inconsistencies were found. The testing results were then analyzed to determine whether the website had met the design objectives or still required improvements in structure or visual appearance.

System accuracy testing was conducted to measure the accuracy of the website in displaying page structure and content according to the design.

The testing was carried out using testing scenarios that included:

1. Header and Navigation Display Testing: Verifying that the navbar is displayed in a fixed position, has a background color of #2c3e50, and navigation links function with smooth scrolling.
2. Section Content Testing: Ensuring that the profile photo is displayed with a 50% border-radius, home, about, and contact texts are displayed with center alignment, and spacing is proportional.
3. Footer Testing: Verifying that the footer appears at the bottom of the page with correct copyright information.
4. Responsiveness Testing: Ensuring that the layout remains optimal at 1920×1080 desktop, 1366×768 laptop, and 768×1024 tablet screen sizes.

System accuracy was calculated based on the percentage of successful website display against all testing scenarios performed, using the following formula:

$$\text{Accuracy} = (\text{Number of Successful Scenarios} / \text{Total Number of Testing Scenarios}) \times 100\%$$

Based on the testing results, from 20 testing scenarios conducted, including functional testing, compatibility testing, responsiveness testing, and code validation, all scenarios (20/20) were successfully completed. The page structure, text content, profile image, and supporting information were displayed without errors. The testing results showed that the website display accuracy reached 100% across all testing scenarios. W3C validation showed no errors in the HTML and CSS code, and Google Lighthouse testing produced a performance score of 95/100, accessibility score of 100/100, and best practices score of 100/100. This indicates that the simple website designed using HTML and CSS had fulfilled the design objectives and was able to display information accurately and consistently.

Based on the research results, the simple website designed using HTML and CSS was able to display information in a structured and easy-to-understand manner. The website was designed as a static website with a client-side rendering architecture, where all content is rendered directly by the browser without requiring server processing. The structure consisting of navigation header, main content sections such as home, about, and contact, and footer enables profile information to be presented systematically and neatly without requiring complex data processing.

The use of HTML as a markup language proved effective in building the basic website structure. The application of HTML5 semantic elements such as <nav>, <section>, and <footer> improved website accessibility and Search Engine Optimization (SEO), in accordance with W3C recommendations (2023). The HTML elements used were able to organize website content logically so that it could be displayed properly by web browsers and understood easily by screen readers for users with disabilities.

Meanwhile, CSS played an important role in managing the visual appearance of the website. The implementation of CSS with an external file followed the principle of separation of concerns, separating structure (HTML) from presentation (CSS), making code maintenance and scalability easier (Duckett, 2011). The techniques applied included:

- a. Flexbox for responsive navigation layout.
- b. Fixed positioning for a navbar that remains visible during scrolling.
- c. Smooth scrolling for better user experience.
- d. Hover effects for visual interactivity.
- e. Responsive design using percentage-based width and viewport units (vh).

The testing results showed that the website could be accessed and displayed properly through Google Chrome and Mozilla Firefox web browsers without display errors. Cross-browser compatibility testing showed good display consistency across platforms, and responsiveness testing proved that the layout remained optimal at various screen sizes. All content, such as profile photos, brief descriptions, and supporting information, could be displayed according to the initial design.

This indicates that the use of pure HTML and CSS is sufficient to build a simple static website. The results of this study are in line with Duckett (2011), who stated that HTML and CSS are the main foundations of web development, as well as MDN Web Docs (2023), which emphasizes the importance of fundamental understanding before adopting additional frameworks or libraries. Robbins (2018) also emphasized that mastery of pure HTML and CSS provides full control over design and improves understanding of basic web development concepts.

Although the designed website still has limitations, such as the absence of dynamic interactivity, including form validation or complex animation, and no database connection for data processing, this approach is appropriate for learning purposes in basic Web Programming courses and for static profile websites. Further development can be carried out by adding JavaScript to improve website interactivity, implementing CSS frameworks such as Bootstrap to accelerate development, or integrating a backend with technologies such as PHP, Node.js, or Python for more complex functionality.

## CONCLUSION

Based on the results of designing, implementing, and testing a simple website using HTML and CSS, it can be concluded that the website was successfully built with a clear, systematic, and easy-to-understand structure. The application of HTML5 semantic elements such as `<nav>`, `<section>`, and `<footer>` was able to improve page structure, accessibility, and Search Engine Optimization (SEO). In its implementation, HTML was used to organize the structure and content of the page, while CSS played a role in managing the visual appearance, including layout, typography, and color schemes, making the website more attractive and responsive.

The resulting website is static and capable of displaying complete group profile information, including navigation menus, profile photos, group descriptions, contact information, and a footer containing copyright information. All website components could be displayed properly through web browsers according to the prepared design. The testing results showed that the website ran without display errors on various browsers, such as Google Chrome and Mozilla Firefox, with an information presentation accuracy level of 100% across all testing scenarios. In addition, code validation showed no errors in HTML or CSS, and performance testing produced excellent scores in performance, accessibility, and best practices. These findings indicate that the use of pure HTML and CSS is sufficient to build a simple static website as a learning medium for basic web programming.

The design of this website is also highly suitable for beginner students because it does not

require frameworks, server-side programming languages, or databases. Mastery of basic HTML and CSS concepts provides a strong foundation for more complex web development in the future, including JavaScript integration, front-end frameworks, and backend technologies. Therefore, the design and implementation of a simple website using HTML and CSS can serve as an effective learning medium for understanding the fundamentals of web development.

Based on the research results, future website development can be directed toward adding JavaScript to enhance interactivity, such as form validation, animation, and dynamic content loading. In addition, more advanced responsive design techniques using CSS Grid and media queries should be applied so that the website can adapt optimally to various device sizes. Future development may also include backend and database integration so that the website can support dynamic features such as Create, Read, Update, and Delete (CRUD). To increase website visibility in search engines, the implementation of more comprehensive Search Engine Optimization (SEO) strategies is also recommended. Furthermore, usability testing involving real users needs to be conducted to obtain more objective feedback and improve the quality of user experience.

### **ACKNOWLEDGMENT**

The authors would like to express their gratitude to all parties who supported the implementation of this research.

### **REFERENCES**

- Berners-Lee, T. (1999). *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web*. HarperCollins.
- Duckett, J. (2011). *HTML and CSS: Design and Build Websites*. John Wiley & Sons.
- Internet World Stats. (2024). *World Internet Users Statistics and 2024 World Population Stats*. Retrieved from <https://www.internetworldstats.com/stats.htm>
- MDN Web Docs. (2023). *HTML: HyperText Markup Language*. Mozilla Foundation. <https://developer.mozilla.org/en-US/docs/Web/HTML>
- MDN Web Docs. (2023). *CSS: Cascading Style Sheets*. Mozilla Foundation. <https://developer.mozilla.org/en-US/docs/Web/CSS>
- Meyer, E. A. (2006). *CSS: The Definitive Guide (3rd ed.)*. O'Reilly Media.
- Robbins, J. N. (2018). *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics (5th ed.)*. O'Reilly Media.
- Sommerville, I. (2016). *Software Engineering (10th ed.)*. Pearson Education.
- Tanenbaum, A. S., & Wetherall, D. J. (2011). *Computer Networks (5th ed.)*. Pearson Education.
- W3C. (2023). *HTML5: A vocabulary and associated APIs for HTML and XHTML*. World Wide Web Consortium. <https://www.w3.org/TR/html5/>
- W3Schools. (2023). *HTML Tutorial*. <https://www.w3schools.com/html/>
- W3Schools. (2023). *CSS Tutorial*. <https://www.w3schools.com/css/>