

Perbandingan Efektivitas Bootstrap dan Tailwind CSS dalam Pengembangan UI Web Responsif

Miftah Farooq Santoso

Engineering and Informatics Departement, Information Technology Study Program, Bina Sarana Informatika University,
miftah.mfq@bsi.ac.id

Submitted: 20-09-2025, Reviewed: 25-09-2025, Accepted 06-10-2025
<https://doi.org/10.47233/jteksis.v7i4.2260>

Abstract

This study presents a comparative evaluation of Bootstrap 5 and Tailwind CSS as front-end frameworks in modern web interface development. Most prior studies have primarily focused on Bootstrap, while academic investigations on Tailwind remain limited. The method employed involved developing five simple prototypes with both frameworks, followed by analysis based on development time, lines of code, maintainability, design flexibility, and responsiveness. The findings indicate that Bootstrap enables prototypes to be completed on average 20–25% faster (± 30 minutes) than Tailwind (± 40 minutes), owing to the availability of ready-to-use components. Conversely, Tailwind proves more efficient in code structure with approximately 15% fewer additional CSS lines, while also offering superior customization flexibility. The contribution of this research lies in expanding the literature on CSS framework comparisons by providing empirical evidence through direct implementation, an area still rarely addressed in academic studies. Framework selection should therefore be aligned with project goals: Bootstrap is more suitable for rapid implementation, whereas Tailwind is preferable for unique and adaptive design.

Keywords: Bootstrap, Tailwind CSS, front-end framework, responsive design, comparative analysis.

Abstrak

Penelitian ini membahas evaluasi komparatif antara Bootstrap 5 dan Tailwind CSS sebagai framework front-end dalam pengembangan antarmuka web modern. Sebagian besar studi terdahulu lebih menyoroti Bootstrap, sementara kajian akademis tentang Tailwind masih terbatas. Metode yang digunakan adalah pengembangan lima prototipe sederhana dengan kedua framework, kemudian dianalisis berdasarkan waktu pengembangan, jumlah baris kode, maintainability, fleksibilitas desain, dan responsivitas. Hasil penelitian menunjukkan bahwa Bootstrap menghasilkan prototipe rata-rata 20–25% lebih cepat (± 30 menit) dibanding Tailwind (± 40 menit) berkat ketersediaan komponen siap pakai. Sebaliknya, Tailwind lebih efisien dalam struktur kode dengan penghematan sekitar 15% baris CSS tambahan, sekaligus unggul dalam fleksibilitas kustomisasi. Kontribusi penelitian ini adalah memperluas literatur perbandingan framework CSS modern dengan memberikan bukti empiris berbasis implementasi langsung, yang masih jarang dikaji secara akademis. Pemilihan framework sebaiknya disesuaikan dengan tujuan proyek: Bootstrap tepat untuk implementasi cepat, sedangkan Tailwind lebih sesuai untuk desain unik dan adaptif.

Keywords: Bootstrap, Tailwind CSS, framework front-end, desain responsif, komparatif.

This work is licensed under Creative Commons Attribution License 4.0 CC-BY International license



PENDAHULUAN

Perkembangan teknologi web modern menuntut antarmuka pengguna (*User Interface/UI*) yang adaptif, responsif, dan mudah dikelola. *Framework* CSS berperan penting dalam mempercepat pengembangan antarmuka dengan menyediakan komponen siap pakai maupun utilitas yang dapat disesuaikan. Bootstrap telah lama menjadi *framework* dominan karena kemudahan implementasi, dokumentasi yang lengkap, serta konsistensi tampilan [1],[2].

Sejumlah penelitian juga menunjukkan penerapan Bootstrap dalam berbagai konteks, seperti integrasi dengan teknologi *frontend* dan pengembangan *Single Page Application (SPA)* [3]. Hal ini memperkuat posisi Bootstrap sebagai solusi utama dalam literatur akademik maupun praktik industri. Namun, penelitian-penelitian tersebut

umumnya berfokus pada implementasi tunggal Bootstrap, tanpa membandingkan alternatif lain yang menawarkan paradigma berbeda.

Tailwind CSS hadir dengan pendekatan *utility-first* yang menekankan fleksibilitas desain dan efisiensi kode, namun dengan kurva belajar lebih tinggi. Kajian komparatif yang menyoroti keunggulan dan keterbatasan Bootstrap serta Tailwind CSS masih terbatas, padahal keduanya memiliki perbedaan paradigma yang signifikan. Penelitian ini hadir untuk mengisi celah tersebut melalui analisis kritis terhadap efisiensi kode, fleksibilitas desain, responsivitas antarmuka, dan maintainability struktur kode, sehingga dapat menjadi referensi bagi pengembang, pendidik, maupun mahasiswa dalam memilih *framework* CSS sesuai kebutuhan pengembangan web modern.

Melalui studi komparatif ini diharapkan diperoleh pemahaman yang lebih komprehensif mengenai keunggulan dan kelemahan masing-masing framework. Bootstrap diprediksi tetap unggul dalam aspek implementasi awal yang cepat dan konsistensi desain, sementara Tailwind CSS memberikan fleksibilitas lebih tinggi dalam kustomisasi antarmuka. Hasil penelitian ini diharapkan dapat menjadi referensi bagi pengembang, mahasiswa, maupun pendidik dalam memilih *framework CSS* yang sesuai dengan kebutuhan proyek berbasis web responsif.

METODE PENELITIAN

Penelitian ini menggunakan pendekatan kombinasi antara *literature review* dan *prototyping* sederhana. Pendekatan ini dipilih agar analisis perbandingan Bootstrap dan Tailwind CSS tidak hanya bersifat teoritis, tetapi juga didukung oleh implementasi praktis melalui pembuatan contoh antarmuka web responsif.

2.1. Rancangan Penelitian

Langkah penelitian dilakukan secara bertahap sebagai berikut:

- A. Identifikasi Masalah: Merumuskan perbedaan dan kebutuhan dalam pemilihan framework CSS untuk pengembangan antarmuka web modern.
- B. Studi Literatur: Mengkaji penelitian terdahulu terkait penggunaan Bootstrap pada pengembangan web responsif, serta referensi awal terkait munculnya Tailwind CSS.
- C. Perancangan Prototipe: Membuat antarmuka web sederhana menggunakan Bootstrap 5 dan Tailwind CSS dengan fitur dasar seperti komponen *button*, *card*, dan *layout grid*, huruf dan warna.
- D. Pengujian & Analisis: Mengamati hasil implementasi kedua framework berdasarkan aspek efisiensi pengembangan, fleksibilitas desain, serta maintainability kode.
- E. Penyusunan Laporan: Menyajikan hasil perbandingan dalam bentuk tabel, gambar, dan pembahasan komparatif

2.2 Objek dan Ruang Lingkup

Objek penelitian difokuskan pada framework CSS Bootstrap dan Tailwind CSS. Ruang lingkup dibatasi pada implementasi antarmuka web sederhana dan responsif, tanpa mencakup integrasi penuh dengan back-end atau sistem skala besar.

2.3 Bahan dan Alat

Tabel 1. Bahan dan Alat Uji Coba

No	Perangkat Uji Coba			
	Kategori	Nama	Spesifikasi	Keterangan
1	Hardware	Laptop	Intel Core i3, RAM 8 GB, SSD 128 GB	Pengembangan & pengujian prototipe
2	Software	Visual Studio Code	Versi 1.95	Editor kode utama
3	Software	Bootstrap	5.3.3	Framework CSS untuk implementasi prototipe
4	Software	Tailwind	3.4.1	Framework CSS alternatif untuk perbandingan
5	Hardware	Smartphone	Layar 6", Android	Uji responsivitas tampilan
6	Software	Google Chrome/Firefox	129.0	Browser
7	Software	Node.js	22.7.0	Runtime Environment JavaScript

Sumber: Miftah Faruq

2.4 Teknik Pengumpulan Data

Pengumpulan data dilakukan melalui: Studi Literatur: Mengacu pada jurnal, artikel, dan dokumentasi resmi framework. Observasi Prototipe: Melakukan pencatatan waktu pengembangan, tingkat kemudahan kustomisasi, dan hasil tampilan antarmuka pada berbagai ukuran layar.

2.5 Teknik Analisis

Data dianalisis dengan metode komparatif. Hasil implementasi Bootstrap dan Tailwind dibandingkan berdasarkan kriteria berikut: Efisiensi pengembangan (waktu koding, ketersediaan komponen siap pakai), Fleksibilitas desain (kemudahan kustomisasi tampilan). *Maintainability* kode (struktur dan keterbacaan kode CSS/HTML).

Dengan metode ini, penelitian diharapkan mampu memberikan gambaran praktis mengenai kelebihan dan kekurangan masing-masing framework CSS, serta rekomendasi penggunaannya dalam pengembangan antarmuka web modern.

2.6 Studi Literatur

A. HTML (*HyperText Markup Language*)

HTML adalah bahasa markup standar yang digunakan untuk membangun struktur

halaman web dengan menyusun elemen seperti teks, gambar, tabel, dan tautan[4]. Menurut[5], HTML berfungsi sebagai kerangka dasar (*skeleton*) dari aplikasi berbasis web yang nantinya diperindah dengan CSS dan interaktif melalui JavaScript. Sementara[6] menekankan bahwa perkembangan HTML5 memberikan dukungan elemen semantik yang lebih kaya untuk pengembangan web modern.

B. CSS (Cascading Style Sheets)

CSS adalah bahasa gaya (*style sheet language*) yang digunakan untuk mengatur tampilan halaman web, termasuk warna, font, margin, dan layout[7]. Penelitian[8] menjelaskan bahwa CSS mendukung pemisahan antara konten dan desain sehingga memudahkan maintainability kode. Sementara[9] menambahkan bahwa penggunaan CSS memungkinkan pengembangan web responsif dengan memanfaatkan *media query*.

C. Bootstrap CSS

Bootstrap CSS adalah framework CSS berbasis komponen yang menyediakan elemen siap pakai untuk membangun antarmuka web responsif dengan cepat[10]. Menurut[5], keunggulan Bootstrap terletak pada konsistensi desain, dokumentasi yang lengkap, serta ketersediaan grid system yang mempermudah pengaturan layout. Penelitian[11] menunjukkan bahwa Bootstrap banyak digunakan dalam proyek pendidikan maupun aplikasi bisnis karena stabilitasnya.

D. Tailwind CSS

Tailwind CSS merupakan framework berbasis utility-first yang memungkinkan pengembang mengatur desain langsung melalui kelas utilitas pada elemen HTML[12]. Menurut[13], pendekatan ini memberikan fleksibilitas tinggi dalam membuat antarmuka unik tanpa terikat gaya bawaan. Sementara[14] menekankan bahwa Tailwind lebih efisien dalam membangun prototipe modern dan mudah diintegrasikan dengan kerangka kerja JavaScript.

E. Visual Studio Code

Visual Studio Code (VS Code) adalah editor kode sumber yang ringan namun memiliki fitur canggih seperti integrasi Git, debugging, dan dukungan ekstensi[15]. Penelitian[16] menyebutkan bahwa VS Code menjadi salah satu editor terpopuler untuk pengembangan web karena gratis

dan multiplatform. Menurut[17], VS Code juga mendukung berbagai plugin yang mempermudah integrasi dengan framework CSS maupun JavaScript.

F. Browser Google Chrome

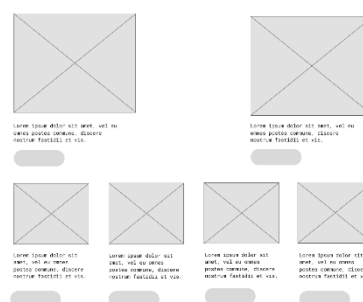
Browser merupakan perangkat lunak utama untuk mengakses dan menampilkan konten web. Menurut[18], browser modern seperti Chrome dan Firefox menyediakan developer tools yang membantu pengembang melakukan debugging dan optimasi. Penelitian[19] menunjukkan bahwa pengujian multi-browser penting untuk menjamin konsistensi tampilan dan performa aplikasi.

G. Node.js

Node.js adalah runtime environment berbasis JavaScript yang memungkinkan pengembang menjalankan kode JavaScript di sisi server maupun pengelolaan alat bantu pengembangan[20]. Menurut[21], Node.js mendukung ekosistem package manager (npm) yang banyak dipakai untuk instalasi framework modern seperti Tailwind CSS. Sementara[22] menegaskan bahwa Node.js mendukung proses otomatisasi build proyek yang mempermudah efisiensi workflow.

2.7 Perancangan Desain & Prototipe

Prototipe dalam penelitian ini dirancang dengan model sederhana berupa artikel berita yang terdiri atas tiga elemen utama: gambar, teks deskripsi, dan tombol aksi “Read More/Selengkapnya”. Dapat dilihat pada gambar 1.



Gambar 1. Rancangan Layout

Desain ini dipilih karena mewakili pola umum antarmuka web informatif, sederhana, dan mudah diimplementasikan pada kedua framework.

A. Instalasi Bootstrap CSS

Instalasi Bootstrap dilakukan melalui dua pendekatan, yaitu menggunakan CDN (*Content Delivery Network*) dan melalui *package manager* (NPM). Pada penelitian ini digunakan metode CDN agar

implementasi lebih sederhana dan mudah direplikasi. Setelah integrasi CSS dan JavaScript Bootstrap berhasil, langkah selanjutnya adalah membangun struktur halaman dengan memanfaatkan sistem grid, navbar, dan card untuk menampilkan konten.

Bootstrap dapat digunakan dengan menambahkan link CDN berikut pada bagian `<head>` dokumen HTML:

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
```

B. Struktur Grid Bootstrap

Bootstrap menyediakan sistem grid berbasis *flexbox* yang memungkinkan pengembang menyusun tata letak responsif dengan mudah. Struktur dasarnya terdiri dari tiga komponen utama, yaitu *container*, *row*, dan *column*. *container* berfungsi sebagai pembungkus utama, *row* untuk membagi baris, sedangkan *col*- digunakan untuk membagi kolom dengan ukuran yang fleksibel.

```
<div class="container">
  <div class="row">
    <div class="col-md-6">Kolom 1</div>
    <div class="col-md-6">Kolom 2</div>
  </div>
</div>
```

Potongan kode di atas memperlihatkan penggunaan kelas `col-md-6` yang membagi area menjadi dua kolom dengan lebar sama pada perangkat medium ($\geq 768\text{px}$) ke atas. Pada perangkat berlayar kecil, kolom akan otomatis ditampilkan secara vertikal sehingga tetap responsif.

C. Implementasi Layout Artikel

Setelah memahami struktur grid dasar, tahap selanjutnya adalah penerapan nyata dalam bentuk layout artikel. Pada prototipe ini, setiap artikel terdiri dari gambar (kiri), teks deskripsi (kanan), serta sebuah tombol aksi "Selengkapnya".

```
<div class="container">
  <div class="row">
    <div class="col-md-4">
      
    </div>
    <div class="col-md-8 d-flex flex-column">
      <h5>Judul Artikel</h5>
```

```
<p>Deskripsi singkat artikel yang
menjelaskan isi konten.</p>
<div class="mt-auto">
  <a href="#" class="btn btn-primary
btn-sm">Selengkapnya</a>
</div>
</div>
</div>
```

Desain tersebut menunjukkan bagaimana grid Bootstrap memudahkan penyusunan elemen visual. Gambar ditempatkan di kolom pertama dengan ukuran sepertiga lebar layar (`col-md-4`), sedangkan teks dan tombol aksi diletakkan di kolom kedua (`col-md-8`). Penggunaan kelas `d-flex flex-column` serta `mt-auto` membuat tombol selalu berada di bagian bawah kolom, menjaga konsistensi antarmuka.

Untuk pengembangan lebih lanjut, variasi layout dapat dibuat, misalnya: Dua kolom artikel populer pada bagian atas halaman dan empat kolom artikel terbaru pada bagian bawah halaman. Dengan demikian, implementasi grid Bootstrap tidak hanya mendukung fleksibilitas desain, tetapi juga mempercepat proses pembuatan antarmuka web yang adaptif.

D. Instalasi Tailwind

Tailwind CSS dapat diinstal melalui berbagai cara, mulai dari menggunakan CDN hingga konfigurasi penuh menggunakan Node.js. Pada penelitian ini digunakan pendekatan instalasi berbasis Node Package Manager (NPM) karena lebih fleksibel dalam pengaturan dan memungkinkan penggunaan build tool untuk mengoptimalkan ukuran file CSS. Langkah awal adalah melakukan inisialisasi proyek dengan perintah berikut pada terminal:

```
npm init -y
npm install -D tailwindcss
npx tailwindcss init
```

Perintah di atas akan membuat file konfigurasi `tailwind.config.js` yang dapat digunakan untuk menyesuaikan skema warna, tipografi, dan *utility classes* sesuai kebutuhan.

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

Potongan kode ini memastikan bahwa gaya dasar (*base*), komponen bawaan, serta *utility classes* dapat digunakan langsung dalam file HTML. Setelah itu, proses kompilasi dilakukan dengan menjalankan perintah:

```
npx tailwindcss -i ./src/input.css -
o ./dist/output.css --watch
```

Hasil kompilasi menghasilkan file output.css yang kemudian dipanggil dalam dokumen HTML melalui tag <link>. Dengan pendekatan ini, pengembangan antarmuka dapat dilakukan secara modular sekaligus menjaga efisiensi ukuran file.

Alternatif lain dapat menggunakan metode CDN dengan cara sebagai berikut:

```
<script
src="https://cdn.tailwindcss.com"></
script>
```

Instalasi menggunakan CDN memiliki keunggulan berupa proses yang cepat dan praktis, sehingga sesuai untuk kebutuhan prototipe

E. Struktur Grid Tailwind CSS

Tailwind tidak punya component grid bawaan seperti Bootstrap, melainkan memakai *utility classes*. Beberapa yang paling sering dipakai: container yang berfungsi untuk membungkus konten dengan lebar maksimal responsif, *grid* yang berfungsi untuk mengaktifkan *grid layout*, *grid-cols-{n}* untuk menentukan jumlah kolom (misalnya *grid-cols-2* = 2 kolom), *sm: md: lg: xl:* untuk breakpoints untuk responsif, *gap-{n}* untuk pengaturan jarak antar kolom/baris.

```
<div class="container mx-auto p-4">
  <div class="grid grid-cols-1 md:grid-
cols-2 gap-4">
    <div class="bg-blue-200 p-4">Kolom
1</div>
    <div class="bg-green-200 p-4">Kolom
2</div>
  </div>
</div>
```

Potongan kode diatas berfungsi untuk mengatur layout agar memiliki 2 kolom dan responsif. Pada mobile hanya 1 kolom (*grid-cols-1*), mulai layar medium (*md*) jadi 2 kolom, *gap-4* memberi jarak antar kolom/baris. Struktur ini adalah dasar untuk semua layout di Tailwind.

F. Implementasi Layout Artikel dengan Tailwind

Struktur kode yang akan diterapkan untuk membangun prototipe artikel: Artikel Populer (2 kolom) yang menggunakan *md:grid-cols-2* dan Artikel Terbaru (4 kolom) yang menggunakan *sm:grid-cols-2 lg:grid-cols-4*.

```
<!-- Struktur Grid -->
<div class="grid gap-6 md:grid-cols-2">
  <!-- Card Artikel -->
```

```
<div class="bg-white rounded-xl shadow-
md flex flex-col">
  
  <div class="p-4 flex flex-col flex-
grow">
    <h3 class="text-xl font-
semibold">Judul Artikel</h3>
    <p class="text-gray-600">Deskripsi
singkat...</p>
    <a href="#" class="mt-auto px-4 py-2
bg-blue-600 text-white rounded-
lg">Selengkapnya</a>
  </div>
</div>
</div>
```

Potongan kode di atas memperlihatkan penggunaan grid Tailwind dengan *md:grid-cols-2* untuk membagi layout menjadi dua kolom pada layar medium ke atas, dan satu kolom pada layar kecil. Komponen artikel ditempatkan dalam card yang terdiri atas gambar, judul, deskripsi, dan tombol aksi. Dengan prinsip *utility-first*, setiap elemen diatur menggunakan kelas seperti *bg-white*, *rounded-xl*, *shadow-md*, *p-4*, hingga *mt-auto* untuk memastikan tata letak tetap rapi dan responsif.

Selanjutnya, Tailwind dihubungkan dengan file CSS utama untuk memungkinkan penerapan utilitas pada komponen antarmuka. Untuk memperkuat analisis komparatif, penelitian ini mengembangkan lima variasi prototipe yang merepresentasikan kondisi berbeda dalam pengembangan antarmuka web, yaitu sebagai berikut:

1. Artikel Tunggul

Struktur dasar terdiri atas satu gambar di sisi kiri dan teks deskripsi di sisi kanan, dengan tombol aksi "Read More" yang ditempatkan di bawah deskripsi. Tujuan variasi ini adalah mengevaluasi penggunaan komponen dasar dan sistem grid sederhana.

2. Artikel Berganda (Grid)

Menampilkan beberapa artikel dalam satu baris dengan memanfaatkan sistem grid. Pada Bootstrap digunakan kelas *.row* dan *.col*, sedangkan pada Tailwind digunakan utilitas *grid-cols* atau *flex*. Variasi ini bertujuan untuk menguji konsistensi layout saat menampilkan banyak konten secara bersamaan.

3. Artikel dengan Variasi Tombol

Tombol aksi diberi variasi gaya, meliputi ukuran (*small*, *medium*, *large*), warna (*primary*, *secondary*), serta efek hover. Fokus variasi ini adalah membandingkan fleksibilitas masing-masing framework dalam memberikan opsi kustomisasi tombol.

4. Artikel Responsif
Prototipe diuji pada berbagai perangkat, meliputi desktop, tablet, dan smartphone. Tujuan variasi ini adalah menilai bagaimana setiap framework mengatur breakpoint serta mendukung desain adaptif pada berbagai ukuran layar.
5. Artikel dengan Elemen Tambahan
Menambahkan detail informasi seperti label kategori atau tanggal publikasi. Variasi ini dimaksudkan untuk menilai kemudahan integrasi elemen tambahan dalam struktur layout.

Seluruh variasi prototipe diimplementasikan menggunakan Bootstrap 5 dan Tailwind CSS secara terpisah dengan struktur konten dan komponen yang sama. Pendekatan ini memastikan bahwa perbandingan dilakukan secara adil (*apple to apple*), sehingga analisis dapat difokuskan pada perbedaan sintaks, efisiensi pengembangan, fleksibilitas desain, serta struktur kode yang dihasilkan.

Pengujian dilakukan dengan membuat prototipe antarmuka sederhana menggunakan Bootstrap 5 dan Tailwind CSS. Pada masing-masing implementasi, waktu pengerjaan dicatat menggunakan stopwatch sejak halaman HTML kosong hingga prototipe selesai. Selain itu, *maintainability* dinilai melalui dua indikator: (1) jumlah baris CSS tambahan yang ditulis di luar *library* utama, dan (2) keterbacaan kode HTML, yang diamati dari banyaknya kelas utilitas yang digunakan serta kerapian struktur kode. Indikator pengukuran dirangkum pada Tabel 2.

Tabel 2. Indikator Pengukuran

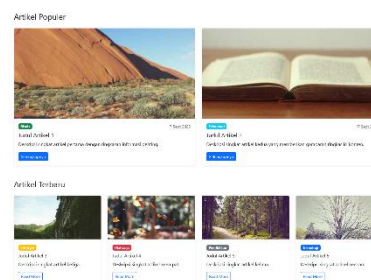
No	Indikator Pengukuran		
	Aspek	Indikator	Teknik Pengukuran
1	Efisiensi Waktu	Lama pengerjaan prototipe (Bootstrap vs Tailwind)	Stopwatch (menit)
2	Efisiensi Kode	Jumlah baris HTML & CSS tambahan	Perhitungan baris
3	<i>Maintainability</i>	(a) Jumlah CSS tambahan; (b) keterbacaan kode HTML	Observasi manual
4	Responsivitas UI	Kemampuan prototipe menyesuaikan berbagai ukuran layar	Uji coba perangkat

Sumber: Miftah Farooq

HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil implementasi dan analisis komparatif antara Bootstrap 5 dan Tailwind CSS berdasarkan lima variasi prototipe yang telah

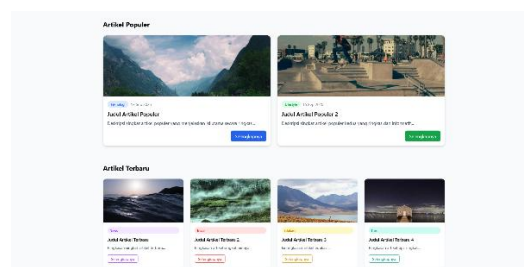
dikembangkan, yaitu artikel tunggal, artikel berganda, artikel dengan variasi tombol, artikel responsif, serta artikel dengan elemen tambahan.



Sumber: Miftah Farooq

Gambar 2. Tampilan Layout dengan Bootstrap

Dari hasil implementasi, ditemukan bahwa efisiensi pengembangan relatif lebih tinggi pada Bootstrap karena ketersediaan komponen siap pakai, seperti card, button, dan grid system, yang dapat langsung digunakan tanpa penulisan kelas tambahan. Sebaliknya, Tailwind CSS menuntut penulisan kelas utilitas lebih panjang, tetapi memberikan fleksibilitas lebih tinggi dalam menyesuaikan desain sesuai kebutuhan.



Sumber: Miftah Farooq

Gambar 2. Tampilan Layout dengan Tailwind

Pada aspek fleksibilitas desain, Tailwind terbukti lebih unggul karena pendekatan berbasis utilitas memungkinkan pengembang mengatur margin, padding, warna, dan tipografi secara presisi tanpa harus menambahkan CSS eksternal. Bootstrap memang menyediakan opsi kustomisasi melalui variabel dan utility classes, tetapi cakupannya lebih terbatas jika dibandingkan dengan Tailwind.

Tabel 3. Perbandingan Bootstrap 5 dan Tailwind CSS

Perbandingan Bootstrap 5 dan Tailwind CSS			
No	Aspek	Bootstrap 5	Tailwind
1	Pendekatan Desain	Berbasis component siap pakai (card, button, grid).	Berbasis utility-first, setiap gaya diatur melalui kelas utilitas.
2	Efisiensi Kode	HTML lebih ringkas karena banyak komponen dibungkus dalam kelas semantik.	HTML lebih panjang karena menumpuk banyak kelas utilitas.
3	Fleksibilitas	Terbatas pada opsi bawaan; kustomisasi lebih sulit tanpa override CSS.	Sangat fleksibel; setiap detail desain bisa diatur langsung dari kelas.
4	Responsivitas	Menggunakan breakpoint bawaan (sm, md, lg, xl).	Breakpoint fleksibel, dapat disesuaikan di konfigurasi tailwind.config.
5	Ketersediaan Komponen	Banyak komponen standar siap pakai (nav, modal, form, carousel).	Tidak ada komponen bawaan; pengembang harus menyusun dari utilitas dasar.
6	Kecepatan Pengembangan	Cepat untuk layout standar karena komponen sudah tersedia.	Cepat untuk desain kustom; lebih lambat jika butuh banyak variasi manual.
7	Keterbacaan Kode	Lebih bersih dan mudah dipahami untuk pemula.	Kelas utilitas menumpuk, tapi lebih jelas menunjukkan fungsi tiap gaya.
8	Kustomisasi Desain	Terbatas pada tema bawaan, perlu SCSS/CSS tambahan untuk detail khusus.	Sangat mudah kustomisasi langsung dari kelas atau konfigurasi tema.
9	Ukuran File CSS	Lebih ringan karena hanya memuat kelas bawaan.	Bisa sangat besar (± 3 MB via CDN), tapi optimal jika pakai purge CSS.
10	Kesesuaian Proyek	Cocok untuk proyek cepat dengan tampilan umum/standar.	Cocok untuk proyek yang butuh desain unik, konsisten, dan adaptif.

Sumber: Miftah Farooq

Sementara itu, dalam hal struktur dan keterbacaan kode, Bootstrap menghasilkan kode HTML yang lebih ringkas karena banyak fungsi sudah dibungkus dalam komponen. Akan tetapi, kondisi ini terkadang membuat desain lebih kaku jika diperlukan penyesuaian detail. Tailwind menghasilkan HTML yang lebih panjang akibat banyaknya kelas utilitas, tetapi setiap baris kode menunjukkan fungsi yang jelas, sehingga memudahkan pengembang dalam melakukan pemeliharaan.

Dari sisi responsivitas, kedua framework mampu menampilkan layout dengan baik pada berbagai perangkat. Bootstrap memanfaatkan sistem breakpoint yang sudah ditentukan sebelumnya (sm, md, lg, xl), sedangkan Tailwind menawarkan breakpoint yang lebih fleksibel dan dapat disesuaikan melalui konfigurasi. Pengujian pada prototipe artikel menunjukkan bahwa keduanya dapat menampilkan konten secara konsisten pada desktop, tablet, maupun smartphone, meskipun Tailwind memberikan keleluasaan lebih besar dalam menyesuaikan ukuran.

Akhirnya, pada integrasi elemen tambahan seperti label kategori dan tanggal publikasi, Bootstrap mempermudah integrasi melalui komponen badge atau text-muted yang sudah tersedia, sedangkan Tailwind mengharuskan penulisan manual menggunakan kelas utilitas. Walaupun demikian, Tailwind tetap memberikan keleluasaan penuh dalam menentukan gaya visual dari elemen tambahan tersebut.

Secara keseluruhan, hasil penelitian menunjukkan bahwa Bootstrap lebih unggul dari sisi efisiensi waktu pengembangan berkat ketersediaan

komponen bawaan, sedangkan Tailwind unggul dalam fleksibilitas desain dan kustomisasi tampilan. Oleh karena itu, pemilihan framework sebaiknya mempertimbangkan kebutuhan proyek: Bootstrap cocok untuk pengembangan cepat dengan tampilan standar, sedangkan Tailwind lebih tepat digunakan untuk proyek yang membutuhkan desain unik dan konsistensi gaya visual yang tinggi.

Berdasarkan Tabel 3, terlihat bahwa Bootstrap 5 dan Tailwind CSS memiliki pendekatan yang berbeda dalam pengembangan antarmuka. Bootstrap lebih menekankan pada ketersediaan komponen siap pakai sehingga mampu mempercepat proses pembuatan prototipe dengan tampilan standar. Hal ini menjadikan struktur kode lebih ringkas dan mudah dipahami, terutama bagi pengembang pemula. Sebaliknya, Tailwind CSS mengedepankan pendekatan utility-first yang menawarkan fleksibilitas tinggi dalam mengatur detail desain. Walaupun menghasilkan HTML yang lebih panjang, struktur kelas utilitas yang digunakan memperlihatkan fungsi yang jelas dari setiap gaya, sehingga memudahkan pemeliharaan dalam jangka panjang.

Dari sisi responsivitas, keduanya mampu menampilkan konten secara konsisten pada berbagai perangkat, meskipun Tailwind lebih unggul karena mendukung penyesuaian breakpoint yang fleksibel melalui konfigurasi. Bootstrap lebih efisien untuk proyek yang membutuhkan implementasi cepat dengan tampilan konvensional, sementara Tailwind lebih sesuai digunakan pada proyek yang menuntut desain unik dan konsistensi gaya visual yang tinggi. Dengan demikian, hasil penelitian ini menegaskan bahwa pemilihan framework

sebaiknya disesuaikan dengan kebutuhan spesifik proyek dan tujuan pengembangan.

Berdasarkan prototipe (artikel tunggal, artikel berganda, variasi tombol, responsif, dan elemen tambahan), diperoleh rata-rata waktu pengembangan sebagai berikut, lihat tabel 4.

Tabel 4. Rata-rata waktu per Prototype

No	Framework	Rata-rata waktu per Prototype		
		Rata-rata waktu	Jumlah Baris HTML	CSS Tambahan
1	Bootstrap 5	±30 menit	120 baris	0-10 baris
2	Tailwind CSS	±40 menit	150 baris	0 baris

Sumber: Miffah Farooq

Data ini menunjukkan bahwa Bootstrap relatif lebih cepat digunakan karena adanya komponen siap pakai, sedangkan Tailwind memerlukan waktu lebih lama akibat penulisan kelas utilitas yang lebih banyak.

Bagi pengembang pemula, Bootstrap lebih ramah karena dokumentasi dan komponen standarnya dapat langsung digunakan tanpa memahami detail CSS yang kompleks. Hal ini mempercepat kurva pembelajaran. Sebaliknya, bagi pengembang senior, Tailwind memberikan fleksibilitas tinggi dalam mengatur desain detail tanpa perlu membuat stylesheet tambahan, sehingga memudahkan kontrol penuh terhadap konsistensi visual proyek.

Hasil ini sejalan dengan penelitian oleh Harahap, dkk.[23] yang menemukan bahwa Bootstrap lebih unggul dalam aspek kecepatan prototyping, sementara Tailwind lebih unggul dalam fleksibilitas desain jangka panjang.

SIMPULAN

Penelitian ini menyimpulkan bahwa Bootstrap 5 dan Tailwind CSS memiliki keunggulan dan keterbatasan masing-masing dalam pengembangan antarmuka web. Bootstrap lebih unggul dalam hal kecepatan implementasi awal dan konsistensi tampilan karena ketersediaan komponen siap pakai. Sementara itu, Tailwind CSS memberikan fleksibilitas lebih tinggi melalui pendekatan utility-first yang memungkinkan kustomisasi desain secara detail, meskipun membutuhkan struktur kode yang lebih panjang. Dengan demikian, pemilihan framework sebaiknya disesuaikan dengan tujuan dan kebutuhan proyek: Bootstrap tepat digunakan untuk pengembangan cepat dengan tampilan standar, sedangkan Tailwind CSS lebih cocok untuk proyek yang menekankan desain unik, adaptif, dan konsistensi gaya visual.

Kontribusi penelitian ini adalah memperluas literatur mengenai perbandingan framework CSS modern dengan memberikan analisis empiris berbasis implementasi langsung pada lima variasi prototipe. Temuan ini dapat menjadi referensi praktis bagi pengembang maupun peneliti yang ingin memahami trade-off antara efisiensi pengembangan dan fleksibilitas desain.

Namun, penelitian ini memiliki keterbatasan karena hanya menguji prototipe sederhana dalam lingkup artikel dan elemen antarmuka dasar. Hasilnya mungkin berbeda apabila framework digunakan dalam proyek berskala besar dengan kompleksitas fitur dan integrasi yang lebih tinggi.

Untuk penelitian lanjutan, disarankan dilakukan analisis performa pada proyek skala besar serta eksplorasi integrasi Bootstrap maupun Tailwind dengan framework JavaScript modern seperti React atau Vue, sehingga dapat diperoleh gambaran yang lebih komprehensif mengenai efektivitas kedua framework dalam ekosistem pengembangan web masa kini.

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada Universitas Bina Sarana Informatika atas dukungan fasilitas dan lingkungan akademik yang memungkinkan penelitian ini dapat diselesaikan dengan baik.

DAFTAR PUSTAKA

- [1] M. F. Santoso, "Jurnal Media Informatika [JUMIN] Implementasi Teknologi Frontend Modern pada Website Yellowweb: Kolaborasi Boostrap 5 Framework dan jQuery," *J. Media Inform.*, vol. 6, no. 2, pp. 873–883, 2025, [Online]. Available: https://www.researchgate.net/publication/388654337_Implementasi_Teknologi_Frontend_Modern_pada_Website_Yellowweb_Kolaborasi_Boostrap_5_Framework_dan_jQuery
- [2] M. F. Santoso, "MF Optimization of Delivery Management Through a Web-Based Monitoring Application Using PHP and Bootstrap," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 7, no. 3, pp. 355–363, 2025, doi: 10.47233/jteksis.v7i3.1686.
- [3] M. F. Santoso, "TEKNIK SINGLE PAGE APPLICATION (SPA) LAYOUT WEB DENGAN MENGGUNAKAN REACT JS DAN BOOTSTRAP," *J. Khatulistiwa Inform.*, vol. 9, pp. 107–114, Dec. 2021.
- [4] R. Akbar and M. Mukhtar, "Perancangan E-Tracer Study berbasis Sistem Cerdas," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 9, no. 1, pp. 8–12, 2020, doi: 10.32736/sisfokom.v9i1.631.
- [5] H. Nuryansyah and E. Hermawan, "Perancangan Sistem Informasi Manajemen Ekstrakurikuler Berbasis Web Pada Sekolah Menengah Kejuruan Negeri 5 Kota Bandung," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 10, no. 3, pp. 298–305, 2021, doi: 10.32736/sisfokom.v10i3.1199.
- [6] R. Kurniawan, Y. Sholva, and E. E. Pratama, "Pengembangan Front-End Aplikasi Lomba Mewarnai menggunakan HTML Canvas dengan Library React JS," vol. 11, no. 2, pp. 211–229, 2025.
- [7] M. A. Irwansyah, T. Tursina, and A. Perwitasari, "Pengembangan Website Dinas Kesehatan Kota

- Singkawang,” *J. Edukasi dan Penelit. Inform.*, vol. 6, no. 2, p. 175, 2020, doi: 10.26418/jp.v6i2.28821.
- [8] A. R. Chrismanto, A. Wibowo, L. Chrisantyo, and M. N. A. Rini, “Implementasi Feature Driven Development untuk Mempermudah Ekualitas Fitur dan Adaptasi pada Pengembangan Portal Dutatani Web dan Mobile,” *J. Edukasi dan Penelit. Inform.*, vol. 8, no. 1, p. 62, 2022, doi: 10.26418/jp.v8i1.50715.
- [9] D. Derisma, “Analyzing User Experience of Website for the Learning of Programming Language,” *J. Edukasi dan Penelit. Inform.*, vol. 6, no. 1, p. 1, 2020, doi: 10.26418/jp.v6i1.40000.
- [10] D. Dahnial, “Aplikasi E-Voting Untuk Pemilihan Ketua Osis Di Sma Xyz Berbasis Web Responsive,” *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 9, no. 1, pp. 144–151, 2020, doi: 10.32736/sisfokom.v9i1.819.
- [11] D. Nasien and A. Andi, “Optimization of Genetic Algorithm in Courses Scheduling,” *IT J. Res. Dev.*, vol. 6, no. 2, pp. 151–161, 2022, doi: 10.25299/itjrd.2022.7896.
- [12] S. Azhariyah and Muhammad Mukhlis, “Framework CSS: Tailwind CSS Untuk Front-End Website Store PT. XYZ,” *J. Inform.*, vol. 3, no. 1, pp. 30–36, 2024, doi: 10.57094/ji.v3i1.1601.
- [13] P. P. Arhandi, S. N. Arief, and A. T. Firdausi, “JIP (Jurnal Informatika Polinema) PENGEMBANGAN WEBSITE PENDUKUNG MASTERY BASED LEARNING UNTUK PEMBELAJARAN MAHASISWA,” *J. Inform. Polinema*, vol. 1, no. 9, pp. 51–58, 2022.
- [14] K. Arya Bayu Wirya Kesuma, I. N. Y. A. Wijaya, and I. G. J. E. Putra, “Implementasi Next.Js, Typescript, Dan Tailwind Css Untuk Pengembangan Aplikasi Frontend Sistem Inventory Perusahaan Apar (Studi Kasus: CV Indoka Surya Jaya),” *Jikom J. Inform. dan Komput.*, vol. 14, no. 2, pp. 95–108, 2024, doi: 10.55794/jikom.v14i2.195.
- [15] J. Firmando, B. Franko, S. Pratama Tanzil, N. Wilyanto, H. Christianto Tan, and E. M. Hartati Kom, “Pembuatan Website Menggunakan Visual Studio Code di SMA Xaverius 3 Palembang,” *Fordicate*, vol. 3, no. 1, pp. 1–8, 2023.
- [16] M. Dicky and S. Lubis, “Jurnal Widya DESIGN PERANCANGAN SISTEM INFORMASI WEBSITE TEXT EDITOR VISUAL STUDIO CODE Jurnal Widya,” vol. 6, no. April, pp. 34–52, 2025.
- [17] A. A. Kurniawan, “Analisis Performa Progressive Web Application (Pwa) Pada Perangkat Mobile,” *J. Ilm. Inform. Komput.*, vol. 25, no. 1, pp. 18–31, 2020, doi: 10.35760/ik.2020.v25i1.2510.
- [18] A. Herwanto, “Analisis Perbandingan Kinerja Browser: Studi Kasus Google Chrome , Mozilla Firefox,” vol. 4, no. 3, pp. 3545–3549, 2025.
- [19] K. W. Prasastyo, “Analisis Anteseden Loyalitas Penggunaan Web Browser Google Chrome dengan Job Related It sebagai Variabel Moderating,” *J. Bisnis dan Akunt.*, vol. 18, no. 1, pp. 63–72, 2016, [Online]. Available: <https://jurnaltsm.id/index.php/JBA/article/view/38>
- [20] A. T. Saputro and M. Novita, “Comparative Analysis of Express and Hono Framework Performance in Simple Registration Application,” *Sinkron*, vol. 9, no. 1, pp. 406–412, 2025, doi: 10.33395/sinkron.v9i1.14333.
- [21] Y. Darmi, K. Pinandita, and U. Muhammadiyah Bengkulu, “Implementasi Perbandingan Metode GraphQL Dan Rest Api Pada Teknologi Nodejs Comparative Implementation of GraphQL and Rest Api Methods in Nodejs Technology,” *J. Inf. Technol. Comput. Sci.*, vol. 7, no. 1, 2024.
- [22] I. F. Balino and Y. R. Beeh, “Perancangan Sistem Aplikasi Gudang Toko Angkasa Jaya Motor Berbasis Web Menggunakan Node.Js,” *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 9, no. 4, pp. 2220–2232, 2024, doi: 10.29100/jupi.v9i4.5587.
- [23] P. Mata, K. Perancangan, W. Design, B. Harahap, M. R. Ramadhan, and N. Kurniawan, “Analisis Framework , Library Front-End Populer : Bootstrap , Tailwind,” vol. 4, no. 2, pp. 329–334, 2025.