

Analisis Perbandingan CSS Framework Tailwind CSS dan Bootstrap dalam Pengembangan Landing Page Website POS Digitaliz

Rholand Deo Eka Putra¹⁾, Ariesta Lestari²⁾, Novera Kristianti³⁾

¹⁾²⁾³⁾ Jurusan Teknik Informatika, Fakultas Teknik, Universitas Palangka Raya
Kampus UPR, Jl. Yos Sudarso Palangka Raya

¹⁾ rholanddeo@mhs.eng.upr.ac.id

²⁾ ariesta@it.upr.ac.id

³⁾ noverakristianti@eng.upr.ac.id

Abstrak

Penelitian ini membandingkan pengalaman pengguna dalam penggunaan dua *framework CSS* populer, *Bootstrap* dan *Tailwind*, melalui studi kasus *landing page*. Penelitian menggunakan metode *user journey mapping* untuk mengamati perjalanan pengguna, yang meliputi *web developer* sebagai responden. Metode ini menghasilkan *journey map* yang merangkum kesulitan dan kepuasan pengguna selama interaksi.

Hasil penelitian menunjukkan bahwa *Tailwind* unggul dalam enam indikator, yaitu skor *journey* yang lebih tinggi, penggunaan *class* yang lebih banyak, lebih sedikit pemakaian *CSS native*, kemiripan desain yang lebih tinggi, tingkat kesulitan yang lebih mudah dari ekspektasi, serta waktu penyelesaian yang lebih singkat. Sementara itu, *Bootstrap* lebih unggul dalam tiga indikator, yaitu jumlah karakter yang lebih sedikit, efek *hover* yang lebih baik, dan responsifitas yang lebih tinggi.

Kata kunci: *CSS Framework, Tailwind CSS, Bootstrap, Perbandingan, Journey Map*

Abstract

This research compares user experiences in using two popular CSS frameworks, Bootstrap and Tailwind, through landing page case studies. The research uses the user journey mapping method to observe the user journey, which includes web developers as respondents. This method produces a journey map that summarizes user difficulties and satisfaction during interactions.

The research results show that Tailwind is superior in six indicators, namely higher journey scores, more class usage, less native CSS usage, higher design similarity, easier difficulty level than expected, and shorter completion time. Meanwhile, Bootstrap is superior in three indicators, namely fewer characters, better hover effects, and higher responsiveness

Keywords: *CSS Framework, Tailwind CSS, Bootstrap, Comparison, Journey Map*

1. PENDAHULUAN

Front-end web development, atau pengembangan *client-side*, melibatkan pembuatan *HTML*, *CSS*, dan *JavaScript* untuk menghasilkan situs atau aplikasi web yang interaktif. Dalam pengembangan *front-end*, pengembang harus mengikuti perkembangan alat dan teknik yang terus berubah. Desain situs bertujuan agar informasi mudah dibaca dan *relevan*, yang menjadi tantangan dengan adanya variasi perangkat, layar, dan resolusi. *Developer* juga harus memperhatikan kompatibilitas lintas *browser* dan perangkat [7].

Seorang *front-end developer* bertugas merancang dan mengembangkan situs atau aplikasi *web* yang dapat dilihat dan diakses langsung oleh pengguna. Tugas utama mencakup membuat *landing page*, yang penting dalam pemasaran digital karena menjadi titik awal pengunjung mengenal produk atau layanan. *Landing page* yang efektif dapat meningkatkan konversi dan penjualan. Untuk mempercepat pengembangan, banyak *developer* menggunakan *CSS Framework*, seperti *Bootstrap* dan *Tailwind*, yang menawarkan berbagai komponen siap pakai dan kemudahan *styling* [4].

Menurut survei *State of CSS* dari 2019 sampai dengan 2023, *Bootstrap* dan *Tailwind* adalah dua *CSS framework* paling populer [10]. Penelitian lebih lanjut oleh Akbar (2023) menyimpulkan bahwa *Bootstrap* cocok untuk pengembangan cepat dengan komponen siap pakai, sementara *Tailwind* lebih fleksibel untuk desain yang kompleks [1]. Studi komparatif yang lebih mendalam diperlukan untuk memahami keunggulan dan kelemahan masing-masing dalam membangun *landing page*, sehingga membantu *developer* memilih *framework* yang sesuai dengan kebutuhan proyek mereka.

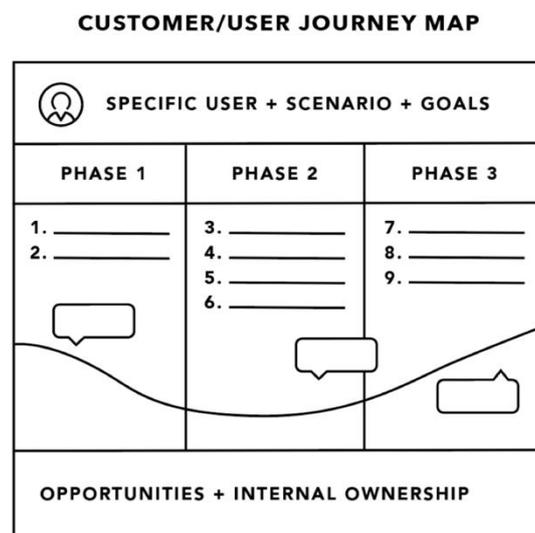
2. TINJAUAN PUSTAKA

Penelitian-penelitian tentang *framework front-end* telah dilakukan untuk memahami performa dan kemudahan penggunaan. Pada 2023, Akbar menemukan bahwa *Bootstrap* cocok untuk pengembangan cepat dengan komponen siap pakai, sedangkan *Tailwind* lebih fleksibel untuk desain kompleks [1]. Kusuma pada 2022 membandingkan *Materialize* dan *Semantic-UI* dan menyimpulkan bahwa *Materialize* lebih pas untuk desain *modern* minimalis, sementara *Semantic-UI* cocok untuk tampilan kompleks [5]. Pada 2020, Chastro dan Darmawan menunjukkan bahwa *Blade Template* memiliki kinerja lebih cepat daripada *Vue.js*, tetapi *Vue* lebih mudah digunakan dan lebih cocok untuk aplikasi web kompleks [3]. Sofi'ie dan Qoiriah pada 2022 menemukan bahwa *Vue* memiliki performa lebih baik, tetapi *React* unggul dalam ekosistem dan komponen yang dapat digunakan kembali [2]. Penelitian oleh Yusuf dkk. Pada 2020 menggunakan metode *SAW* menilai *Bootstrap* sebagai *framework* terbaik berdasarkan kriteria seperti *preprocessor*, *responsivitas*, dan kemudahan penggunaan [6].

2.1 User Journey Mapping

User journey mapping adalah alat visual yang digunakan untuk memahami perjalanan pengguna dari awal hingga akhir saat berinteraksi dengan suatu produk atau layanan yang hasilnya dirangkum dalam sebuah Peta yaitu *journey map*. Peta ini membantu mengidentifikasi titik-titik kesulitan atau kepuasan dalam penggunaan dan dapat menjadi landasan untuk meningkatkan pengalaman pengguna [8].

Dalam bentuknya yang paling dasar, *journey map* dimulai dengan menyusun serangkaian tindakan pengguna ke dalam garis waktu. Selanjutnya, garis waktu disempurnakan dengan pemikiran dan emosi pengguna untuk membuat narasi. Narasi ini dipadatkan dan dipoles, pada akhirnya mengarah pada sebuah visualisasi [9].

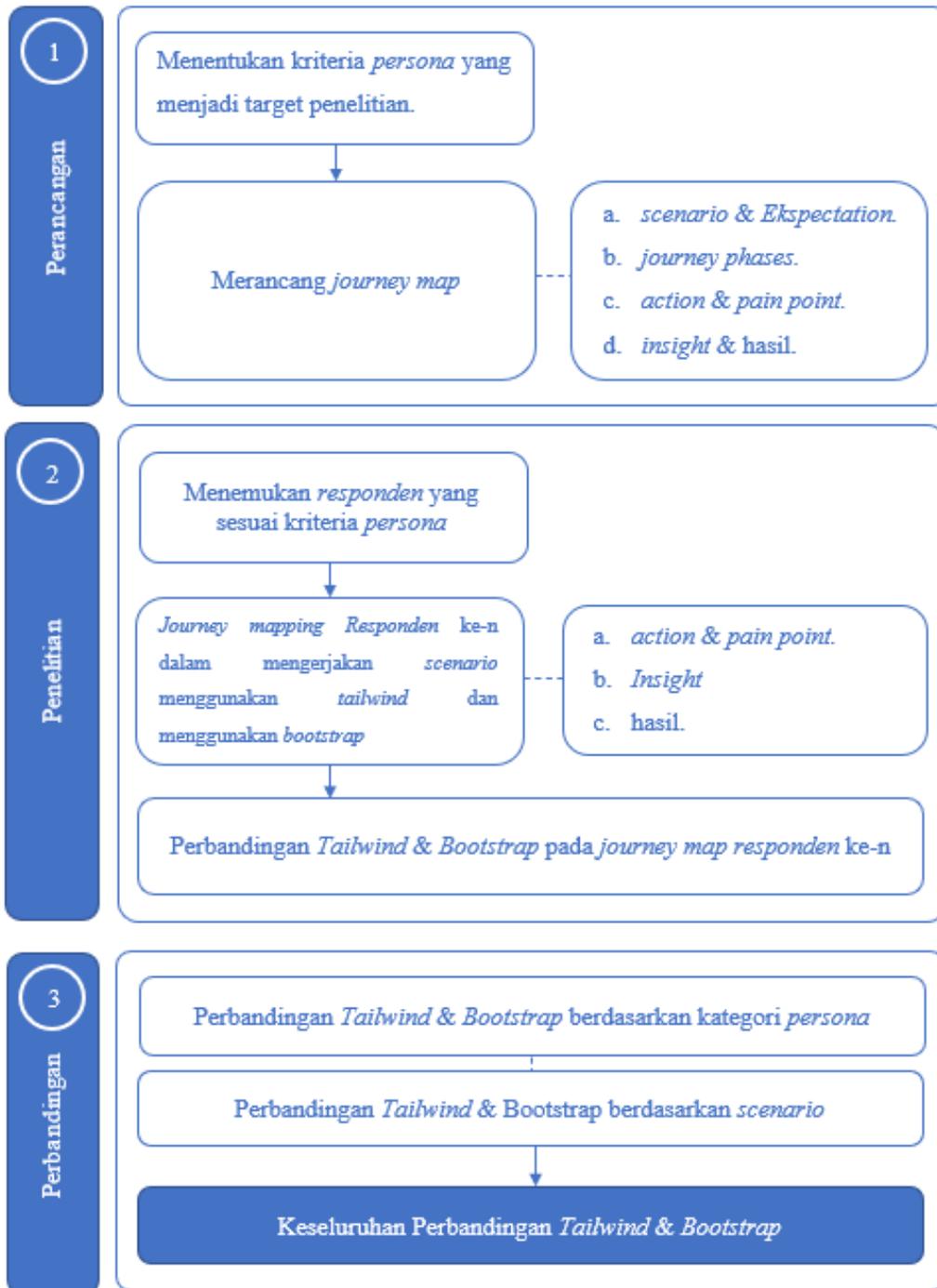


NNGROUP.COM NN/g

Gambar 1. Contoh *User journey map* [9]

3. METODE PENELITIAN

Penelitian dilakukan dengan membandingkan *User journey* penggunaan *CSS Framework Tailwind* dan *Bootstrap* menggunakan metode *User journey mapping*. *User* yang akan diteliti merupakan *Responden* yang memenuhi kriteria *persona*. Setiap *journey* dari *Responden* akan dirangkum kedalam sebuah *journey map*. Adapapun komponen utama dalam sebuah *journey map* yaitu *persona*, *scenario*, *expectation*, *journey phases*, *action*, *pain point*, *insight* [9]. Setelah itu proses dan hasil *journey* yang telah dikerjakan menggunakan kedua *CSS Framework Tailwind* dan *Bootstrap* akan dibandingkan. Maka digambarkan alur penelitian seperti berikut.



Gambar 2. Alur Penelitian

1. Perancangan

- a. Menentukan kriteria *persona* yang menjadi target penelitian.
- b. Merancang *journey map*
 - 1) Membuat *scenario* & mengkategorikan *Expectation*.
 - 2) Menentukan *journey phases*.
 - 3) Mendeteksi *action* & *pain point*.
 - 4) Menemukan *insight* & hasil.
2. Proses penelitian
 - a. Menemukan *Responden* yang sesuai dengan target kriteria *persona*.
 - b. Memetakan *journey* dari proses pengerjaan *scenario* tiap *Responden*.
 - 1) Mendeteksi *action* dan *pain point*
 - 2) Menemukan *Insight*
 - 3) Observasi Hasil
 - c. Membandingkan proses *journey* dan hasil *journey* tiap *Responden*.
3. Hasil analisa perbandingan *CSS Framework Tailwind* dan *Bootstrap* melalui perbandingan *User journey mapping*.

4. PEMBAHASAN

4.1 Perancangan

Journey map						
Persona	Deskripsi <i>Persona</i>					
Scenario	Deskripsi <i>Scenario</i>					
Ekspektasi	Deskripsi Ekspektasi					
Journey Phases	Instalasi	Dokumentasi	Komponen	Template/library	Kelas	Bantuan
Action
Pain Point
Insight	1.... 2....					
Realita	Deskripsi Realita					
Hasil	Hasil Pengerjaan <i>Scenario</i>					

Gambar 3. Kerangka *journey map*

4.1.1 Kriteria Persona

Persona adalah dokumentasi yang berisi penjelasan tentang karakteristik *User* secara detail [11]. *Persona* sering digunakan dalam berbagai bidang, seperti desain, *marketing*, dan pengembangan produk, untuk membantu memahami kebutuhan dan perilaku pengguna. Adapun *persona* yang ditentukan pada penelitian ini yaitu:

- A1** Seseorang yang [*mampu*] mengerjakan *web development* dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*], sebagai [*full-stack / front-end / back-end*] *web developer*. [*Berpengalaman*] sebagai *professional web developer* dengan status [*full-time / part-time / intern / freelance*]. [*Memahami*] *CSS* dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*], [*pernah*] menggunakan *CSS Framework*, [*bisa*] menggunakan *Framework Tailwind* dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*], dan [*bisa*] menggunakan *Framework Bootstrap* dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*].
- A2** Seseorang yang [*mampu*] mengerjakan *web development* dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*], sebagai [*full-stack / front-end / back-end*] *web developer*. [*Berpengalaman*] sebagai *professional web developer* dengan status [*full-time / part-time / intern / freelance*]. [*Memahami*] *CSS* dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*]. [*pernah*] menggunakan *CSS Framework*, [*bisa*] menggunakan *Framework Tailwind*

- dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*], dan [*belum bisa*] menggunakan *Framework Bootstrap*.
- A3** Seseorang yang [*mampu*] mengerjakan *web development* dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*], sebagai [*full-stack / front-end / back-end*] *web developer*. [*Berpengalaman*] sebagai *professional web developer* dengan status [*full-time / part-time / intern / freelance*]. [*Memahami*] CSS dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*], [*pernah*] menggunakan *CSS Framework*, [*belum bisa*] menggunakan *Framework Tailwind*, dan [*bisa*] menggunakan *Framework Bootstrap* dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*].
- A4** Seseorang yang [*mampu*] mengerjakan *web development* dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*], sebagai [*full-stack / front-end / back-end*] *web developer*. [*Berpengalaman*] sebagai *professional web developer* dengan status [*full-time / part-time / intern / freelance*]. [*Memahami*] CSS dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*], [*belum pernah*] menggunakan *CSS Framework*, [*belum bisa*] menggunakan *Framework Bootstrap*, dan [*belum bisa*] menggunakan *Framework Bootstrap*.
- B1** Seseorang yang [*mampu*] mengerjakan *web development* dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*], sebagai [*full-stack / front-end / back-end*] *web developer*. [*Tidak berpengalaman*] sebagai *professional web developer*. [*Memahami*] CSS dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*], [*pernah*] menggunakan *CSS Framework*, [*bisa*] menggunakan *Framework Tailwind* dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*], dan [*bisa*] menggunakan *Framework Bootstrap* dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*].
- B2** Seseorang yang [*mampu*] mengerjakan *web development* dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*], sebagai [*full-stack / front-end / back-end*] *web developer*. [*Tidak berpengalaman*] sebagai *professional web developer*. [*Memahami*] CSS dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*], [*pernah*] menggunakan *CSS Framework*, [*bisa*] menggunakan *Framework Tailwind* dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*], dan [*belum bisa*] menggunakan *Framework Bootstrap*.
- B3** Seseorang yang [*mampu*] mengerjakan *web development* dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*], sebagai [*full-stack / front-end / back-end*] *web developer*. [*Tidak berpengalaman*] sebagai *professional web developer*. [*Memahami*] CSS dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*], [*pernah*] menggunakan *CSS Framework*, [*belum bisa*] menggunakan *Framework Tailwind*, dan [*bisa*] menggunakan *Framework Bootstrap* dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*].
- B4** Seseorang yang [*mampu*] mengerjakan *web development* dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*], sebagai [*full-stack / front-end / back-end*] *web developer*. [*Tidak berpengalaman*] sebagai *professional web developer*. [*Memahami*] CSS dengan tingkat kemampuan [*pemula / cukup mahir / mahir / sangat mahir*], [*belum pernah*] menggunakan *CSS Framework*, [*belum bisa*] menggunakan *Framework Bootstrap*, dan [*belum bisa*] menggunakan *Framework Bootstrap*.

4.1.2 Journey Mapping

1. Scenario

Tabel 1. Scenario

Scenario	Detail
Scenario 1 (S1)	Membuat Section Navbar
Scenario 2 (S2)	Membuat Section Hero

Scenario menggambarkan situasi yang dituju oleh *journey map* [9]. Responden akan diminta mengerjakan kedua *scenario* dua kali yaitu pertama menggunakan *framework Tailwind* dan yang kedua menggunakan *framework Bootstrap*.



Gambar 4. Tampilan UI navbar section



Gambar 5. Tampilan UI komponen hero section

2. Journey phases, action dan pain point

Journey phase adalah serangkaian langkah yang akan atau diharapkan *responden* melaluinya tanpa kendala [9]. Dalam penelitian ini *journey phase* ditentukan berdasarkan langkah-langkah yang harus di lewati untuk menyelesaikan tugas, langkah-langkah yang kemungkinan akan dilakukan dan langkah-langkah yang tidak diharapkan dilakukan. Adapun langkah-langkahnya yaitu: proses instalasi, dokumentasi, penggunaan komponen, penggunaan *tenplate/library* pihak ketiga, penggunaan *class* dan permintaan bantuan (*additional action*, boleh dilakukan boleh juga tidak).

Dari keenam *phases* ini akan dipetakan *action* apa saja yang akan dilakukan oleh responden dan apakah dalam menjalani aksi tersebut, *Responden* mengalami kendala (*pain point*). Untuk memberikan bobot *Journey*, maka digunakan sistem *journey Score* yang dikembangkan sendiri dalam penelitian ini agar memberikan bobot nilai pada *Journey*.

- Jika melakukan aksi maka poin akan ditambah satu (1), jika tidak melakukan atau tidak menjawab maka tidak akan diberikan poin (0).
 - Jika terdapat *pain point* dalam melakukan aksi, maka poin akan dikurangi satu (-1) dan jika tidak terdapat maka *poin* akan bertambah satu (1).
 - Untuk *additional action* (meminta bantuan) jika melakukan maka tidak akan diberikan poin (0) dan jika tidak melakukan akan diberikan satu poin (1).
3. Ekspektasi dan realita

Untuk ekspektasi dan realita digunakan untuk mengukur selisih tingkat kesulitan sebelum dan sesudah pengerjaan *scenario* berdasarkan pandangan *Responden*. Semakin tinggi selisih realita maka akan semakin bagus. Adapun tingkat kesulitan yang ditanyakan sebagai ekspektasi dan realita yaitu: 1) Sangat mudah; 2) Mudah; 3) Cukup mudah; 4) Netral; 5) Cukup sulit; 6) Sulit; 7) Sangat sulit.

- c. Untuk *hover*, apabila komponen memiliki tampilan tersendiri ketika dilakukan *hover* maka akan diberi nilai 100, jika tidak ada akan diberi nilai 0. Kemudian semua komponen akan dihitung rata-rata tampilan *hover*-nya.

$$\frac{(\text{hover komponen 1} + \text{hover komponen n..})}{(\text{jumlah komponen})} \times 100\% \quad (5)$$

- d. Jika komponen memiliki tampilan *mobile responsif*, maka akan diberi nilai 100, dan jika tidak akan diberi nilai 0. Kemudian semua komponen akan dihitung rata-rata tampilan *mobile responsif*-nya.

$$\frac{(\text{mobile responsif komponen 1} + \text{mobile responsif komponen n..})}{(\text{jumlah komponen})} \times 100\% \quad (6)$$

4.1.3 Indikator Perbandingan

Setelah semua proses diatas dilalui maka akan didapatkan kesimpulan perbandingan yang dirangkum dalam tabel berikut.

Tabel 4. Indikator Perbandingan

No	Indikator Perbandingan	Nilai		Selisih (Tailwind - Bootstrap)	Framework Yang Lebih Unggul	
		Tailwind	Bootstrap			
1	Journey Score					
2	Jumlah karakter					
3	Persentase Penggunaan Class (%)					
4	Persentase Penggunaan Style CSS native (%)					
5	Persentase Kemiripan (%)					
6	Persentasi Tampilan Hover (%)					
7	Persentase Responsifitas (%)					
8	Selisih ekpektasi dan realita tingkat kesulitan					
9	Lama penyelesaian (menit)					
Jumlah indikator penilaian yang lebih unggul						

Penjelasan:

- Indikator perbandingan yang pertama adalah *journey Score*, indikator ini didapatkan dari proses *journey mapping* tiap *Responden*. Yang lebih unggul adalah *Framework* yang memiliki *journey score* lebih tinggi. Semakin tinggi *journey score* artinya *Framework* tersebut semakin mudah digunakan karena minim kendala dalam proses penggunaannya.

$$\text{journey point tailwind} - \text{journey point bootstrap} \quad (7)$$

Jika hasil lebih besar dari 0 maka *Tailwind* lebih unggul, jika hasil lebih kecil dari 0 maka *Bootstrap* lebih unggul.

- Indikator yang kedua adalah Jumlah karakter, indikator ini didapatkan dari hasil obeservasi baris kode pengerjaan tugas dalam tiap *scenario* pada tiap *responden*. Yang lebih unggul adalah *Framework* yang memiliki jumlah karakter lebih sedikit, karena semakin sedikit baris kode akan semakin cepat di baca, semakin cepat dibaca akan semakin cepat pula untuk melakukan perubahan ataupun perbaikan jika ada kendala.

$$\text{jumlah karakter tailwind} - \text{jumlah karakter bootstrap} \quad (8)$$

Jika hasil lebih besar dari 0 maka *Bootstrap* lebih unggul, jika hasil lebih kecil dari 0 maka *Tailwind* lebih unggul.

3. Indikator ketiga adalah penggunaan *class* bawaan yang disediakan *Framework*, indikator ini juga didapatkan dari hasil obeservasi baris kode pengerjaan tugas dalam tiap *scenario* pada tiap *responden*. Semakin banyak kelas yang digunakan maka akan semakin bagus, karena artinya *Framework* tersebut sudah memberikan banyak *class* untuk memenuhi kebutuhan *styling*.

$$\text{jumlah class tailwind} - \text{jumlah class bootstrap} \quad (9)$$

Jika hasil lebih besar dari 0 maka *Tailwind* lebih unggul, jika hasil lebih kecil dari 0 maka *Bootstrap* lebih unggul.

4. Indikator keempat adalah penggunaan *manual CSS native*, indikator ini juga didapatkan dari hasil obeservasi baris kode pengerjaan tugas dalam tiap *scenario* pada tiap *responden*. Yang lebih unggul adalah *Framework* yang memiliki jumlah karakter lebih sedikit menggunakan *CSS native*, karena jika banyak menggunakan *CSS* manual artinya *Framework* tersebut tidak menyediakan banyak *class* untuk melakukan customisasi *styling* atau *class* nya sulit dipahami.

$$\text{jumlah class tailwind} - \text{jumlah class bootstrap} \quad (10)$$

Jika hasil lebih besar dari 0 maka *Bootstrap* lebih unggul, jika hasil lebih kecil dari 0 maka *Tailwind* lebih unggul.

5. Indikator kelima adalah kemiripan, indikator ini didapatkan dari hasil observasi tampilan komponen yang dikerjakan dalam *scenario* tiap *responden*. Makin mirip tampilan dengan *UI* makin bagus.

$$\text{kemiripan tailwind} - \text{kemiripan bootstrap} \quad (11)$$

Jika hasil lebih besar dari 0 maka *Tailwind* lebih unggul, jika hasil lebih kecil dari 0 maka *Bootstrap* lebih unggul.

6. Indikator keenam adalah tampilan *hover*, indikator ini juga didapatkan dari hasil observasi tampilan komponen yang dikerjakan dalam *scenario* tiap *responden*. Yang lebih unggul adalah komponen yang memiliki lebih banyak tampilan *hover* didalamnya, untuk *hover* tidak ada kondisi atau gambar *UI* khusus, artinya tidak ada patokan tampilan *UI* hanya berdasarkan kreatifitas *Responden*.

$$\text{hover tailwind} - \text{hover bootstrap} \quad (12)$$

Jika hasil lebih besar dari 0 maka *Tailwind* lebih unggul, jika hasil lebih kecil dari 0 maka *Bootstrap* lebih unggul.

7. Indikator ketujuh perbandingan adalah tampilan *mobile responsive*, indikator ini juga didapatkan dari hasil observasi tampilan komponen yang dikerjakan dalam *scenario* tiap *responden*. Yang unggul adalah tampilan komponen yang dapat terlihat dalam bentuk *mobile*, tidak ada kondisi atau gambar *UI* khusus, artinya tidak ada patokan tampilan *UI* hanya berdasarkan kreatifitas *Responden*.

$$\text{responsifitas tailwind} - \text{responsifitas bootstrap} \quad (13)$$

Jika hasil lebih besar dari 0 maka *Tailwind* lebih unggul, jika hasil lebih kecil dari 0 maka *Bootstrap* lebih unggul.

8. Indikator kedelapan adalah selisih ekspektasi dan realita tingkat kesulitan mengerjakan tugas dalam *scenario* tiap *responden*, indikator ini juga didapatkan dari hasil kuisisioner yang diberikan setelah selesai mengerjakan tugas dalam *scenario* tiap *responden*, jika selisih mengarah ke positif maka realita kesulitan pengerjaan tugas dalam *scenario* dinyatakan lebih mudah daripada ekspektasi *Responden*, sebaliknya jika mengarah ke negatif artinya realita kesulitan pengerjaan tugas dalam *scenario* dinyatakan lebih sulit daripada ekspektasi *Responden*.

$$\text{selisih tailwind} - \text{selisih bootstrap} \quad (14)$$

(Jika hasil lebih besar dari 0 maka *Tailwind* lebih unggul, jika hasil lebih kecil dari 0 maka *Bootstrap* lebih unggul)

9. Indikator kesembilan adalah lama waktu penyelesaian tugas, indikator ini didapatkan dari hasil kuisioner yang diberikan setelah selesai mengerjakan tugas dalam *scenario* tiap *responden*, jadi waktu didapatk berdasarkan *POV Responden*. Semakin cepat waktu penyelesain maka akan semakin baik.

$$\text{waktu tailwind} - \text{waktu bootstrap} \tag{15}$$

Jika hasil lebih besar dari 0 maka *Bootstrap* lebih unggul, jika hasil lebih kecil dari 0 maka *taiwind* lebih unggul

Dari 9 indikator ini yang lebih banyak mengungguli yang akan dijadikan *Framework* lebih baik dalam pembuatan *Landing page Website POS Digitaliz*.

4.2 Proses Perbandingan

4.2.1 Responden

Tabel 5. List Responden

Kriteria Persona	No	Nama
A1	R1	Hendra Gunawan
A3	R2	Dina Meiliana
A4	R3	Ray Nickolson
B1	R4	Febriani Auria
B3	R5	Sihar Haryanto Pandiangan
B4	R6	Amar Filonga Septaraja
Total	6	

Dari 6 *Responden* yang menjadi *subject* penelitian, 1 orang mewakili kriteria *persona* A1 (*Responden* R1), 1 orang mewakili kriteria *persona* A3 (*Responden* R2), 1 orang mewakili kriteria *persona* A4 (*Responden* R3), 1 orang mewakili kriteria *persona* B1 (*Responden* R4), 1 orang mewakili kriteria *persona* B3 (*Responden* R5), dan 1 orang mewakili kriteria *persona* B4 (*Responden* R6). Berikut Penjelasan *persona* tiap *Responden* yang menjadi *subject* penelitian.

4.2.2 Perbandingan hasil pengerjaan keseluruhan *scenario*

Tabel 6. Perbandingan *Tailwind* dan *Bootstrap* dalam pengerjaan kedua *scenario* oleh tiap responden

Framework	Komponen	Persona	Responden	Journey	Hasil						Selisih Ekspektasi & Realita	Waktu (menit)	
				Journey Score	Jumlah karakter (%)	Rasio Class (%)	Rasio Style (%)	Kemiripan (%)	Tampilan Hover (%)	Responsifitas (%)			
Tailwind (S1-T, S2-T)	Navbar (S1-T)	A1	R1	10	2535	58,03	0	100	100	67	0	60	
		A3	R2	9	2899	59,71	0	100	33	100	2	120	
		A4	R3	9	907	38,15	0	76	67	75	-1	300	
		B1	R4	6	2911	51,53	0	93	67	100	0	180	
		B3	R5	9	1061	37,51	0	100	100	50	1	20	
		B4	R6	3	2887	50,92	0	87	67	100	-2	30	
	Hero (S2-T)	A1	R1	11	2958	32,18	0,61	96	0	100	0	60	
		A3	R2	8	2662	47,26	4,62	60	25	69	1	90	
		A4	R3	9	1588	43,95	2,27	27	75	6	0	180	
		B1	R4	6	2982	19,38	26,16	100	0	62,5	0	180	
		B3	R5	6	2530	25,62	32,57	71	50	69	1	40	
		B4	R6	10	2996	19,29	26,03	98	0	37,5	-2	120	
		Total			96	28916	483,53	92,26	1008	584	836	0	1380
		Rata-rata			8	2409,67	40,29	7,69	84	48,67	69,67	0	115
Bootstrap (S1-B, S2-B)	Navbar (S1-B)	A1	R1	9	1986	12,44	21,9	80	100	100	0	30	
		A3	R2	5	1653	19,48	0	53	100	100	-1	60	
		A4	R3	6	1556	24,36	0	47	100	100	-1	240	
		B1	R4	4	1553	22,41	0,77	73	100	100	0	180	
		B3	R5	2	1735	11,64	15,33	75	100	100	0	20	
		B4	R6	2	1545	22,52	0,78	73	100	100	0	60	
	Hero (S2-B)	A1	R1	8	3907	17,56	10,88	98	50	100	-1	60	
		A3	R2	5	2449	10,9	34,01	69	50	12,5	-1	180	
		A4	R3	5	1503	27,15	15,9	14,5	100	44	-1	240	
		B1	R4	5	3374	14,55	31,65	83	75	50	0	180	
		B3	R5	8	2174	13,16	26,91	75	75	69	0	20	
		B4	R6	3	2519	24,06	12,07	23	0	0	-2	300	
		Total			62	25954	220,23	170,2	763,5	950	875,5	-7	1570
		Rata-rata			5,17	2162,83	18,35	14,18	63,63	79,17	72,96	-0,58	130,83

- Journey Score* dari action dan *pain point* yang didapatkan pada *journey* tiap *Responden* dalam mengerjakan semua *scenario*.
- Rekapitulasi observasi baris kode pada *journey* tiap *Responden* dalam mengerjakan semua *scenario*.
- Rekapitulasi observasi hasil tampilan pada *journey* tiap *Responden* dalam mengerjakan semua *scenario*.
- Ekspektasi dan Realita pada *journey* tiap *Responden* dalam mengerjakan semua *scenario*.
- Lama pengerajan *journey* tiap *Responden* dalam mengerjakan semua *scenario*.
- Total dan rata-rata nilai *journey* semua *Responden* dalam mengerjakan tugas pembuatan komponen *navbar* dan *hero* dalam *scenario S1* dan *scenario S2* menggunakan *Tailwind (S1-T dan S2-T)*.
- Total dan rata-rata nilai *journey* semua *Responden* dalam mengerjakan tugas pembuatan komponen *navbar* dan *hero* dalam *scenario S1* dan *scenario S2* menggunakan *Bootstrap (S1-B dan S2-B)*.

4.3 Hasil

Tabel 7. Hasil perbandingan *Tailwind* dan *Bootstrap*

No	Indikator Perbandingan	Nilai rata-rata		Selisih (Tailwind - Bootstrap)	Framework Yang Lebih Unggul	
		Tailwind	Bootstrap			
1	<i>Journey Score</i>	8	5,17	2,83	<i>Tailwind</i>	
2	Jumlah karakter	2409,67	2162,83	246,83		<i>Bootstrap</i>
3	Persentase Penggunaan <i>Class</i> (%)	40,29	18,35	21,94	<i>Tailwind</i>	
4	Persentase Penggunaan <i>Style CSS native</i> (%)	7,69	14,18	-6,50	<i>Tailwind</i>	
5	Persentase Kemiripan (%)	84	63,63	20,38	<i>Tailwind</i>	
6	Persentasi Tampilan <i>Hover</i> (%)	48,67	79,17	-30,50		<i>Bootstrap</i>
7	Persentase <i>Responsifitas</i> (%)	69,67	72,96	-3,29		<i>Bootstrap</i>
8	Selisih ekpektasi dan realita tingkat kesulitan	0,00	-0,58	0,58	<i>Tailwind</i>	
9	Lama penyelesaian (menit)	115,00	130,83	-15,83	<i>Tailwind</i>	
Jumlah Indikator Penilaian Yang Lebih Unggul					6	3

Berdasarkan keseluruhan perbandingan *journey mapping* semua *Responden* dalam tabel 6, maka didapatkan hasil perbandingan *Tailwind* dan *Bootstrap* yang dapat dilihat pada tabel 7.

Dapat diketahui bahwa secara keseluruhan *Tailwind* lebih unggul dalam 6 indikator yaitu *journey score* yang lebih tinggi, penggunaan *class* yang lebih banyak, penggunaan *style CSS native* yang lebih sedikit, tingkat kemiripan yang lebih tinggi, realita tingkat kesulitan yang lebih mudah dibandingkan ekpektasi, serta waktu penyelesaian yang lebih singkat.

Dibandingkan *Bootstrap* yang hanya unggul dalam 3 indikator yaitu jumlah karakter yang lebih sedikit, tampilan *hover* yang lebih baik, dan *responsifitas* yang lebih tinggi.

5. KESIMPULAN

Dari proses analisis yang panjang terhadap 6 *subject* penelitian (*responden R1-R6*) didapatkan kesimpulan perbandingan yaitu:

1. Berdasarkan metode *journey mapping*, *User Experience* menggunakan *Tailwind* untuk membuat *styling* komponen *website* yang sudah memiliki desain *UI* seperti *navbar* dan *hero* lebih baik dibandingkan *Bootstrap*, ini dibuktikan dengan proses *journey phases* yang berjalan dengan baik dan minim kendala (*pain point*) yang didapatkan saat menjalankan *scenario* pembuatan komponen *navbar* dan *hero*, yang keseluruhan-nya dirangkum dalam *journey score*, adapun rata-rata *journey score* dari ke-6 *responden* melakukan *styling* komponen menggunakan *Tailwind* yaitu 8, *score* ini memiliki selisih 2,83 poin lebih tinggi dibandingkan *journey score styling* komponen menggunakan *Bootstrap* yang hanya memiliki nilai 5,17.
2. *Bootstrap* unggul dalam indikator penilaian jumlah rata-rata karakter yang lebih singkat dalam *file html* komponen yang dikerjakan menggunakan *Bootstrap* oleh ke-6 *responden* namun selisihnya tidak jauh berbeda hanya sekitar 247 karakter atau 10% lebih singkat dari rata-rata jumlah karakter dari *file html* komponen yang dikerjakan menggunakan *Tailwind*.
3. Penggunaan *class* yang disediakan oleh *Framework Tailwind* lebih *usefull* dibandingkan *Bootstrap* dengan bukti penggunaan *class* yang lebih banyak yaitu rata-rata 40,29% dari total jumlah karakter.
4. Sedikitnya penggunaan *styling manual* menggunakan *CSS native* yaitu rata-rata hanya 7,69% pada baris kode *file html* komponen yang dikerjakan menggunakan *Tailwind*.
5. Hasil observasi tampilan hasil pengerjaan komponen dalam *scenario* yaitu berupa *file html* dari ke-6 *responden*. Ketika *file html* dijalankan di *web browser*, setiap komponen yang *distyling* menggunakan *Framework Tailwind* memiliki rata-rata tingkat kemiripan 84% terhadap desain *UI* yang diberikan. Poin tersebut lebih tinggi 20,38% jika dibandingkan komponen yang *distyling* menggunakan *Framework Bootstrap* yang hanya memiliki tingkat kemiripan rata-rata 63,63% terhadap desain *UI*.
6. Hasil pengerjaan komponen menggunakan *Bootstrap* juga unggul dalam tampilan *hover* yang lebih baik 30,5% namun untuk indikator *hover* tidak memiliki patokan desain *UI* jadi nilai dari indikator ini hanya kecil mempengaruhi kesimpulan analisa.
7. Hasil pengerjaan komponen menggunakan *Bootstrap* juga sedikit lebih unggul dalam *responsifitas* walaupun hanya selisih 3,29 persen lebih tinggi dibandingkan hasil komponen menggunakan *Tailwind*.
8. Realita kesulitan mengerjakan tugas menggunakan *Tailwind* ternyata lebih mudah dari pada ekspektasi *responden* yang dapat dilihat dari nilai selisih ekspektasi dan realita tingkat kesulitan pengerjaan *scenario* menggunakan *Tailwind* mencapai nilai 0,58 dibandingkan nilai selisih ekspektasi dan realita tingkat kesulitan pengerjaan *scenario* menggunakan *Bootstrap* yang hanya memiliki nilai 0.
9. Waktu pengerjaan tugas pembuatan komponen dalam *scenario* menggunakan *Tailwind* sedikit lebih cepat yaitu 15,83 menit dibandingkan waktu pengerjaan tugas pembuatan komponen dalam *scenario* menggunakan *Bootstrap*.

Dalam total 12 *scenario* (*scenario S1, S2*) x *responden R1, R2, R3, R4, R5, R6* (6) yang dikerjakan oleh *responden* yang dikerjakan 2 kali yaitu pertama menggunakan *Tailwind* dan kedua menggunakan *Bootstrap* (total 24 kali pengerjaan *scenario*). *Tailwind* menang telak dengan dimana 10 *scenario* (*scenario S1* yang dikerjakan oleh *Responden R1, R2, R3, R4, R5, R6* (6) dan *scenario S2* yang dikerjakan oleh *Responden R1, R2, R3, R4* (4), lebih unggul dikerjakan menggunakan *Tailwind*, hanya 1 *scenario* dimana *Bootstrap* lebih unggul yaitu *scenario S2* yang dikerjakan oleh *responden R5*, dan 1 *scenario* seimbang yaitu *scenario S2* yang dikerjakan oleh *Responden R6*.

Jadi secara garis besar perbedaan utama dari *Tailwind* dan *Bootstrap* adalah *Tailwind* lebih baik dalam kostumisasi *styling* yang memiliki tampilan desain *UI* yang harus di samakan, terbukti dari banyaknya penggunaan *class* dan minimnya penggunaan *styling* manual menggunakan *CSS native* dan hasil tampilan yang hampir sama dengan *UI*.

Sedangkan *Bootstrap* cocok digunakan jika tidak ada desain ataupun tidak memiliki *UI* yang rumit, ini terbukti dengan tampilan *hover* yang lebih baik dan responsifitas yang sedikit lebih baik dengan catatan tanpa desain *UI*, dibandingkan komponen yang dikerjakan menggunakan *Tailwind*.

Hasil akhirnya *Tailwind* tetap lebih unggul, karena jika tampilan *hover* dan *mobile responsif* memiliki desain *UI*, maka berdasarkan penelitian ini *Tailwind* akan lebih unggul dibandingkan *Bootstrap*.

Singkatnya jika membuat *webiste* yang mempunyai desain *UI* yang *proper* dan rumit gunakan *CSS Framework Tailwind*. Jika membuat *website* sederhana yang tidak berfokus kepada *UI* seperti hanya untuk belajar ataupun membuat *website individu* yang tidak memiliki desain *UI*, *CSS Framework Bootstrap* dapat menjadi pilihan. Jika ingin memenuhi keduanya gunakan *CSS Framework Tailwind*.

DAFTAR PUSTAKA

- [1] Akbar, T. A., "ANALISIS PERBANDINGAN *FRAMEWORK CSS BOOTSTRAP* DAN *TAILWIND* DALAM PENGEMBANGAN *WEBSITE* PORTOFOLIO", Bachelor Thesis, Universitas AMIKOM Yogyakarta, 2023. [Online]. Available: <https://eprints.amikom.ac.id/id/eprint/18506/>
- [2] Axza, F., Sofi'ie, F., & Qoiriah, A., "Analisis Perbandingan *Framework Front-end Javascript React* dan *Vue* Pada Pengembangan *Website*" *Journal of Informatics and Computer Science (JINACS)*, vol. 05, no. 02, pp. 157–164, 2023. [Online]. Available: <https://doi.org/10.26740/JINACS.V5N02.P157-164>
- [3] Chastro, C., Darmawan, E., Kom, S., & #2, M. T. (2020). Perbandingan Pengembangan Front End Menggunakan *Blade Template* dan *Vue js*. *Jurnal STRATEGI - Jurnal Maranatha*, 2(2), 302–313. [Online]. Available: <https://strategi.itmaranatha.org/index.php/strategi/article/view/181>
- [4] Cody Lindley, "Front-end Developer Handbook 2019 - Learn the Entire *Javascript, CSS* and *HTML Development Practice!*" n.d. Accessed October 3, 2023. [Online]. Available: <https://frontendmasters.com/guides/front-end-handbook/2019/#2>.
- [5] Kusuma, I. J., "ANALISIS PERBANDINGAN *FRAMEWORK CSS materialize* DAN *semantic-ui* PADA FRONT END *WEBSITE* PENDAFTARAN MAHASISWA BARU PADA UTM (UNIVERSITAS TEKNOLOGI MATARAM)," Bachelor Thesis, Universitas Teknologi Digital Indonesia, 2020. [Online]. Available: <https://eprints.utdi.ac.id/9046/>
- [6] L. Yusuf, T. Hidayatulloh, D. Nurlaela, L. D. Utami, F. N. Hasan, "Simple Additive Weighting untuk *Front-end Framework* Terbaik.," *ILKOM Jurnal Ilmiah*, vol. 12, no.2, pp. 136–142, 2020. [Online]. Available: <https://doi.org/10.33096/ilkom.v12i2.593.136-142>
- [7] MDN, "*Front-end Web Developer*." *Mozilla Developer*, 2020. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/Front-end_web_developer.
- [8] Samson, S., Granath, K., & Alger, A., "*journey mapping the User Experience*," *College & Research Libraries*, vol. 78, no. 4, pp. 459. [Online]. Available: <https://doi.org/10.5860/crl.78.4.459>
- [9] Sarah Gibbons, "Journey mapping 101," Nielsen Norman Group, 2018. [Online]. Available: <https://www.nngroup.com/articles/journey-mapping-101/>.
- [10] State Of CSS, "State of CSS 2023: *CSS Frameworks*," 2023. [Online]. Available: <https://2023.stateofCSS.com/en-US/CSS-Frameworks/>.
- [11] Zaki Izzani Akbar, "Apa itu Persona dan Apa Saja Komponen pada Persona," 2020. [Online] Available: <https://sis.binus.ac.id/2020/04/29/apa-itu-persona-dan-apa-saja-komponen-pada-persona/>