

**SKRIPSI**

**SISTEM INFORMASI *BUS ONLINE TICKETING* BERBASIS *WEB*  
PADA PT. BORLINDO MANDIRI JAYA**

**Disusun dan diajukan oleh:**

**BRYAN VALENTINO  
D041 20 1115**



**PROGRAM STUDI SARJANA TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS HASANUDDIN  
GOWA  
2024**

**LEMBAR PENGESAHAN SKRIPSI****SISTEM INFORMASI *BUS ONLINE TICKETING* BERBASIS *WEB*  
PADA PT. BORLINDO MANDIRI JAYA**

Disusun dan diajukan oleh

**Bryan Valentino**  
**D041 20 1115**

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian  
Studi Program Sarjana Program Studi Teknik Elektro  
Fakultas Teknik Universitas Hasanuddin  
Pada tanggal 12 Agustus 2024  
Dan dinyatakan telah memenuhi syarat kelulusan

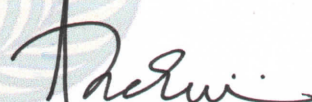
Menyetujui,

Pembimbing Utama,

Pembimbing Pendamping,



Dr. Eng. Wardi, S.T., M.Eng.  
NIP. 19720828 199903 1 003



Andini Dani Achmad, ST., MT.  
NIP. 19880621 201504 2 003

Ketua Program Studi,



Dr. Eng. Ir. Dewtiani, MT. IPM  
NIP. 19691026 199412 2 001

## PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Bryan Valentino  
NIM : D041201115  
Program Studi : Teknik Elektro  
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

### **SISTEM INFORMASI *BUS ONLINE TICKETING* BERBASIS *WEB* PADA PT. BORLINDO MANDIRI JAYA**

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 24 Juli 2024

Yang Menyatakan

  
Bryan Valentino

## ABSTRAK

**BRYAN VALENTINO.** *Sistem Informasi Bus Online Ticketing Berbasis Web pada PT. Borlindo Mandiri Jaya* (dibimbing oleh Dr.Eng. Wardi, S.T., M.Eng. dan Andini Dani Achmad, ST., MT).

Transportasi umum di Indonesia, khususnya bus, memiliki tingkat peminat yang tertinggi. Data Divisi Humas Polri tahun 2023 menunjukkan bus sebagai moda transportasi terpopuler dengan 22,77 juta pemudik (18,39%). Beranjak dari persoalan tersebut, dikembangkanlah sistem informasi *bus online ticketing* yang bekerja sama dengan PT. Borlindo Mandiri Jaya. Penelitian ini bertujuan membuat sistem informasi yang membantu operator dalam manajemen database dan rendering website yang cepat. Metodologi penelitian meliputi perancangan sistem menggunakan *NextJS* dan *MongoDB*, serta pengujian dengan metode *Black-box*, *First Contentful Paint (FCP)*, *Largest Contentful Paint (LCP)*, *Cumulative Layout Shift (CLS)*, dan *System Usability Scale*. Hasil penelitian menunjukkan bahwa framework *NextJS* melakukan proses *rendering* yang cepat, berdasarkan hasil pengujian *FCP*, didapatkan 0.9s pada *mobile* dan 0.2s pada *desktop*. Database *MongoDB* dapat konversi format gambar menjadi format teks sehingga mengubah ukuran file dari awalnya 415.6 kb menjadi 36 kb. Pengujian melalui PageSpeed Insights menghasilkan LCP= 1.8s untuk *mobile* dan 0.4s untuk *desktop* (keduanya kategori *Green/cepat*), serta CLS= 0 untuk *mobile* dan *desktop* (kategori Bagus). Sementara itu, pengujian *System Usability Scale* menghasilkan skor 85.2381, yang menunjukkan kategori Promoter dalam skala *NPS*, *Acceptable* dalam skala *Acceptability*, *Best Imaginable* dalam *Adjective Scale*, dan *Grade A* (Unggul) dalam *Grade Scale*. Hasil-hasil ini mengindikasikan bahwa sistem yang dikembangkan ini memiliki performa yang baik dan diterima positif oleh pengguna.

Kata Kunci: Sistem Informasi, *NextJS*, *MongoDB*, *Black-box*, *LCP*, *CLS*, *SUS*.

## ABSTRACT

**BRYAN VALENTINO.** *Information System Bus Online Ticketing Web-based at PT. Borlindo Mandiri Jaya* (supervised by Dr.Eng. Wardi, S.T., M.Eng. dan Andini Dani Achmad, ST., MT).

Public transport in Indonesia, especially buses, has the highest level of demand. Data from the Public Relations Division of the National Police in 2023 shows buses as the most popular mode of transport with 22.77 million travellers (18.39%). Moving on from this problem, an online ticketing bus information system was developed in collaboration with PT Borlindo Mandiri Jaya. This research aims to create an information system that helps operators in managing databases and fast website rendering. The research methodology includes system design using NextJS and MongoDB, as well as testing with Black-box, First Contentful Paint (FCP), Largest Contentful Paint (LCP), Cumulative Layout Shift (CLS), and System Usability Scale methods. The results showed that the NextJS framework performs a fast rendering process, based on the FCP test results, obtained 0.9s on mobile and 0.2s on desktop. MongoDB database can convert image formats into text formats, thus changing the file size from 415.6 kb to 36 kb. Testing through PageSpeed Insights resulted in LCP = 1.8s for mobile and 0.4s for desktop (both Green/fast categories), and CLS = 0 for mobile and desktop (Good category). Meanwhile, System Usability Scale testing resulted in a score of 85.2381, which indicates the Promoter category in the NPS scale, Acceptable in the Acceptability scale, Best Imaginable in the Adjective Scale, and Grade A (Superior) in the Grade Scale. These results indicate that the developed system performs well and is positively accepted by users.

Keywords: Information System, NextJS, MongoDB, Black-box, LCP, CLS, SUS.

## DAFTAR ISI

LEMBAR PENGESAHAN SKRIPSI.....	i
PERNYATAAN KEASLIAN.....	ii
ABSTRAK .....	iii
ABSTRACT .....	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR .....	viii
DAFTAR TABEL.....	xi
DAFTAR SINGKATAN DAN ARTI SIMBOL .....	xii
DAFTAR LAMPIRAN.....	xiii
KATA PENGANTAR .....	xvii
BAB I PENDAHULUAN .....	1
I.1    Latar Belakang.....	1
I.2    Rumusan Masalah .....	4
I.3    Tujuan Penelitian.....	4
I.4    Manfaat Penelitian.....	4
I.5    Ruang Lingkup .....	5
BAB II TINJAUAN PUSTAKA.....	6
II.1    Sistem Informasi.....	6
II.1.1.    Sistem.....	6
II.1.2.    Informasi .....	7
II.1.3.    Sistem Informasi .....	8
II.2    Reservasi.....	9
II.3    Website .....	9

II.3.1.	<i>Domain</i> .....	10
II.3.2.	<i>Web Hosting</i> .....	10
II.3.3.	Bahasa program.....	10
II.3.4.	Desain <i>Website</i> .....	11
II.3.5.	Basis Data.....	13
II.3.5.1.	Basis Data <i>SQL</i> (Relasional).....	13
II.3.5.2.	Basis Data <i>NoSQL</i> (Non-Relasional).....	13
II.3.6.	<i>Frontend dan Backend</i> .....	14
II.3.6.1.	<i>Frontend</i> .....	14
II.3.6.2.	<i>Backend</i> .....	15
II.3.7.	<i>NextJS</i> .....	15
II.3.8.	<i>MongoDB</i> .....	18
II.3.9.	<i>UI/UX</i> .....	19
II.3.9.1.	User Interface.....	19
II.3.9.2.	User Experience.....	19
II.4	<i>Core Web Vitals</i> .....	20
II.4.1	<i>Largest Contentful Paint (LCP)</i> – Waktu Loading Halaman .....	20
II.4.2	Interaction to Next Paint (INP) – Kemampuan Respons Halaman.	21
II.4.3	<i>Cumulative Layout Shift</i> – Pengalaman Visual Halaman .....	22
II.4.4	<i>First Contentful Paint (FCP)</i> .....	23
II.5	<i>Black-box Testing</i> .....	24
II.6	<i>System Usability Scale</i> .....	26
II.7	Penelitian Relevan.....	29
BAB III METODE PENELITIAN.....		33
III.1	Kerangka Metodologi.....	33

III.1.1.	Alur Penelitian .....	33
III.1.2.	Metode Perancangan .....	34
III.1.3.	Mekanisme Pengujian .....	36
III.2	Lokasi dan Waktu Penelitian.....	38
III.3	Alat dan Bahan Penelitian .....	39
III.4	Rancangan Prototyping .....	40
III.5	Teknik Pengumpulan Data .....	44
BAB IV	HASIL DAN PEMBAHASAN .....	46
IV.1	Analisis Kebutuhan .....	46
IV.2	Prototyping .....	47
IV.3	Evaluasi <i>Prototyping</i> .....	48
IV.4	Implementasi Sistem .....	48
IV.4.1	User Interface (Masyarakat Umum).....	48
IV.4.2	Admin Interface (Internal PT. Borlindo Mandiri Jaya) .....	56
IV.5	Hasil Kinerja Framework dan Database.....	62
IV.6	Pengujian Sistem .....	64
IV.5.1	Pengujian Fungsi.....	64
IV.5.2	<i>Core Web Vitals</i> .....	71
IV.5.3	Kuesioner Usability.....	73
IV.7	Kebutuhan Sistem.....	78
BAB V	KESIMPULAN DAN SARAN.....	80
V.1	Kesimpulan.....	80
V.2	Saran.....	81
DAFTAR	PUSTAKA .....	82
LAMPIRAN	.....	87



## DAFTAR GAMBAR

Gambar II. 1 Figma ( <i>Figma</i> , n.d.).....	12
Gambar II. 2 Nilai <i>LCP</i> dan <i>CLS</i> pada <i>Website</i> tampilan Mobile ( <i>PageSpeed Insights</i> , n.d.) .....	17
Gambar II. 3 Nilai <i>LCP</i> dan <i>CLS</i> pada <i>Website</i> tampilan Desktop ( <i>PageSpeed Insights</i> , n.d.).....	17
Gambar II. 4 Parameter Pengukuran <i>Web Core Vitals</i> .....	20
Gambar II. 5 Perbandingan <i>LCP eligible</i> dan <i>not eligible</i> (Faradilla, 2024).....	21
Gambar II. 6 Skor <i>Interaction to Next Paint (INP)</i> .....	22
Gambar II. 7 Skor <i>First Contentful Paint (FCP)</i> .....	24
Gambar II. 8 Range Skor <i>SUS</i> (Rafid Pratama et al., 2024) .....	29
Gambar III. 1 Alur Penelitian.....	33
Gambar III. 2 Metode <i>System Development Life Cycle (SDLC)</i> Waterfall.....	35
Gambar III. 3 Mekanisme Pengujian .....	36
Gambar III. 4 Contoh tabel pengujian <i>Black-box</i> (Shaleh et al., 2021).....	36
Gambar III. 5 Contoh pengujian <i>Core Web Vitals</i> ( <i>PageSpeed Insights</i> , n.d.).....	37
Gambar III. 6 Contoh hasil pengujian <i>SUS</i> (Saputra, 2019).....	38
Gambar III. 7 <i>DFD Level 0</i> .....	40
Gambar III. 8 <i>DFD Level 1</i> .....	41
Gambar III. 9 <i>DFD Level 2</i> .....	43
Gambar IV. 1 <i>Design UI Website</i> .....	47
Gambar IV. 2 Tampilan Reservasi Tiket .....	49
Gambar IV. 3 Tampilan <i>Experience</i> Penumpang .....	49
Gambar IV. 4 Tampilan Statistik Perusahaan .....	50
Gambar IV. 5 Tampilan Berita Terbaru Perusahaan.....	50
Gambar IV. 6 Tampilan <i>Footer</i> .....	50
Gambar IV. 7 Tampilan <i>Sign In</i> .....	51
Gambar IV. 8 Tampilan <i>Sign Up</i> .....	51
Gambar IV. 9 Tampilan Halaman Tentang Borlindo.....	52
Gambar IV. 10 Halaman Informasi Perwakilan.....	52

Gambar IV. 11 Halaman Informasi <i>Dashboard</i> .....	53
Gambar IV. 12 Halaman Pemilihan Bus.....	54
Gambar IV. 13 Halaman Pemilihan <i>Seat</i> .....	54
Gambar IV. 14 Halaman Pengisian Data Pemesan dan Penumpang .....	55
Gambar IV. 15 Halaman Pembayaran .....	55
Gambar IV. 16 Halaman <i>Waiting Verification Payment</i> .....	55
Gambar IV. 17 Halaman Informasi <i>Dashboard</i> setelah melakukan pemesanan ..	56
Gambar IV. 18 Tampilan E-Ticket setelah transaksi disetujui oleh Admin .....	56
Gambar IV. 19 Halaman Kontrol Bus.....	57
Gambar IV. 20 Tampilan Tambah Bus .....	57
Gambar IV. 21 Tampilan Atur Kursi .....	57
Gambar IV. 22 Halaman Kontrol Transaksi .....	58
Gambar IV. 23 Tampilan Detail Transaksi .....	58
Gambar IV. 24 Halaman Kontrol Rute .....	59
Gambar IV. 25 Tampilan Tambah Rute.....	59
Gambar IV. 26 Tampilan Edit Rute .....	59
Gambar IV. 27 Halaman Kontrol Berita .....	60
Gambar IV. 28 Tampilan Tambah Berita .....	60
Gambar IV. 29 Tampilan Edit Berita.....	60
Gambar IV. 30 Halaman Kontrol Statistik sebelum diisi data.....	61
Gambar IV. 31 Tampilan Tambah Statistik .....	61
Gambar IV. 32 Halaman Kontrol Statistik setelah diisi data .....	61
Gambar IV. 33 Hasil Pengujian <i>FCP</i> pada mode <i>Mobile</i> .....	62
Gambar IV. 34 Hasil Pengujian <i>FCP</i> pada mode <i>Desktop</i> .....	62
Gambar IV. 35 Total Penyimpanan Folder Berita MongoDB .....	64
Gambar IV. 36 Hasil Pengujian <i>LCP</i> dan <i>CLS</i> pada mode <i>Mobile</i> .....	71
Gambar IV. 37 Hasil Pengujian <i>LCP</i> dan <i>CLS</i> pada mode <i>Desktop</i> .....	72
Gambar IV. 38 Skor Pengujian <i>LCP</i> .....	72
Gambar IV. 39 Nilai <i>CLS</i> yang baik adalah 0,1 atau kurang. Nilai yang buruk lebih dari 0,25.....	73
Gambar IV. 40 Persentase Jenis Kelamin Responden .....	73

Gambar IV. 41 Persentase Umur Responden.....	73
Gambar IV. 42 Domisili Responden.....	74
Gambar IV. 43 Status Responden.....	74
Gambar IV. 44 Skala Skor <i>SUS</i> .....	77

## DAFTAR TABEL

Tabel II. 1 Pertanyaan Kuesioner <i>SUS</i> (Saputra, 2019).....	26
Tabel II. 2 Skor <i>SUS</i> (Saputra, 2019).....	27
Tabel II. 3 Range Skor <i>SUS</i> .....	28
Tabel II. 4 State of the Art .....	29
Tabel III. 2 Alat dan Bahan Penelitian.....	39
Tabel III. 3 Pertanyaan kepada pihak perusahaan.....	44
Tabel IV. 1 Ukuran Gambar Isi Menu Berita .....	63
Tabel IV. 2 Pengujian <i>Black-box</i> Home Page.....	65
Tabel IV. 3 Tabel IV. 1 Pengujian <i>Black-box</i> Informasi Perwakilan .....	65
Tabel IV. 4 Tabel IV. 1 Pengujian <i>Black-box</i> Sign Up Page.....	65
Tabel IV. 5 Pengujian <i>Black-box</i> Sign in Page .....	66
Tabel IV. 6 Pengujian <i>Black-box</i> Home Page - Reservasi Tiket .....	66
Tabel IV. 7 Pengujian <i>Black-box</i> Admin Page - Informasi .....	67
Tabel IV. 8 Pengujian <i>Black-box</i> Admin Page - Pemesanan.....	68
Tabel IV. 9 Pengujian <i>Black-box</i> Admin Page - Kontrol Bus .....	68
Tabel IV. 10 Pengujian <i>Black-box</i> Admin Page - Kontrol Transaksi.....	69
Tabel IV. 11 Pengujian <i>Black-box</i> Admin Page - Kontrol Rute.....	69
Tabel IV. 12 Pengujian <i>Black-box</i> Admin Page - Kontrol Berita.....	70
Tabel IV. 13 Pengujian <i>Black-box</i> Admin Page - Kontrol Statistik .....	70
Tabel IV. 14 Analisis Hasil Responden terhadap masing-masing pertanyaan .....	74
Tabel IV. 15 Hasil Kuesioner <i>SUS</i> .....	77

**DAFTAR SINGKATAN DAN ARTI SIMBOL**

Lambang/Singkatan	Arti dan Keterangan
<i>JS</i>	<i>JavaScript</i>
<i>UI</i>	<i>User Interface</i>
<i>UX</i>	<i>User Experience</i>
<i>JSON</i>	<i>JavaScript Object Notation</i>
<i>BSON</i>	<i>Binary JavaScript Object Notation</i>
<i>LCP</i>	<i>Largest Contentful Paint</i>
<i>CLS</i>	<i>Cumulative Layout Shift</i>
<i>FCP</i>	<i>First Contentful Paint</i>
<i>SUS</i>	<i>System Usability Scale</i>
<i>DFD</i>	<i>Data Flow Diagram</i>
<i>NoSQL</i>	<i>Not Only Structured Query Language</i>
<i>SEO</i>	<i>Search Engine Optimization</i>

## DAFTAR LAMPIRAN

Lampiran 1. Surat Pengambilan Data .....	87
Lampiran 1. a Surat Usulan Pengambilan Data Penelitian .....	87
Lampiran 1. b Surat Pengambilan Data Penelitian pada PT. Borlindo Mandiri Jaya .....	88
Lampiran 2. Surat Permohonan Izin Publikasi <i>Website</i> .....	89
Lampiran 3. Kode Pemrograman <i>Front-end Website</i> .....	90
Lampiran 3. a Kode Pemrograman <i>Front-end</i> , ( <i>package.jsON</i> ) – Inisiasi Library .....	90
Lampiran 3. b Kode Pemrograman <i>Front-end</i> , ( <i>tailwind.config.js</i> ) – Konfigurasi Template Styling <i>Website</i> . <i>Source Code</i> terlampir hanya bagian awal dan akhir saja .....	91
Lampiran 3. c Kode Pemrograman <i>Front-end</i> , ( <i>next.config.mjs</i> ) – Konfigurasi Build. <i>Source Code</i> terlampir hanya bagian awal dan akhir saja. Autentikasi kami rahasiakan untuk kepentingan bersama.....	93
Lampiran 3. d Kode Pemrograman <i>Front-end</i> , ( <i>.env</i> ) – <i>API Endpoints</i> . Autentikasi kami rahasiakan untuk kepentingan bersama.....	93
Lampiran 3. e Kode Pemrograman <i>Front-end</i> , <i>src-api-auth-(signin.js)</i> – Menyesuaikan data <i>signin</i> .....	94
Lampiran 3. f Kode Pemrograman <i>Front-end</i> , <i>src-api-auth-(signup.js)</i> – Menyesuaikan data <i>signup</i> .....	94
Lampiran 3. g Kode Pemrograman <i>Front-end</i> , <i>src-api-auth-(signup.js)</i> – Menyesuaikan data <i>signup</i> .....	95
Lampiran 3. h Kode Pemrograman <i>Front-end</i> , <i>src-api-dashboard-(get-dashboard- statistic.js)</i> - Memanggil data dashboard dari <i>Back-end</i> .....	95
Lampiran 3. i Kode Pemrograman <i>Front-end</i> , <i>src-api-news-(get-news-statistic.js)</i> - Memanggil data berita dari <i>Back-end</i> .....	96

Lampiran 3. j Kode Pemrograman <i>Front-end</i> , <i>src-api-route-(get-all-route.js)</i> - Memanggil data rute dari <i>Back-end</i> .....	96
Lampiran 3. k Kode Pemrograman <i>Front-end</i> , <i>src-api-statistic-(get-all-statistic.js)</i> - Memanggil data statistik dari <i>Back-end</i> .....	97
Lampiran 3. l Kode Pemrograman <i>Front-end</i> , <i>src-api-transaction-(get-list-transaction.js)</i> – Memanggil data transaksi dari <i>Back-end</i> .....	97
Lampiran 3. m Kode Pemrograman <i>Front-end</i> , <i>src-components-navbar-(navbar.js)</i> – Struktur komponen navigation-bar .....	98
Lampiran 3. n Kode Pemrograman <i>Front-end</i> , <i>src-app-about-(page.js)</i> – Tampilan Halaman Tentang Borlindo. <i>Source Code</i> terlampir hanya bagian awal dan akhir saja.....	101
Lampiran 3. o Kode Pemrograman <i>Front-end</i> , <i>src-components-atomic-molecules-representative-table-(index.js)</i> – Tampilan Halaman Informasi Perwakilan. <i>Source Code</i> terlampir hanya bagian awal dan akhir saja.....	102
Lampiran 3. p Kode Pemrograman <i>Front-end</i> , <i>src-components-atomic-molecules-landingpage-(section-two.js)</i> – Tampilan Halaman Experience. <i>Source Code</i> terlampir hanya bagian awal dan akhir saja. ....	103
Lampiran 3. q Kode Pemrograman <i>Front-end</i> , <i>src-components-atomic-molecules-landingpage-(section-three.js)</i> – Tampilan Halaman Statistik Perusahaan.....	105
Lampiran 3. r Kode Pemrograman <i>Front-end</i> , <i>src-components-atomic-molecules-news-section-(index.js)</i> – Tampilan Halaman Berita Perusahaan. <i>Source Code</i> terlampir hanya bagian awal dan akhir saja. ....	106
Lampiran 3. s Kode Pemrograman <i>Front-end</i> , <i>src-components-atomic-dashboard-(home.js)</i> – Tampilan Halaman Informasi. <i>Source Code</i> terlampir hanya bagian awal dan akhir saja.....	108
Lampiran 3. t Kode Pemrograman <i>Front-end</i> , <i>src-components-atomic-dashboard-bus-control-(bus-control.js)</i> – Tampilan Halaman Kontrol Bus.....	113
Lampiran 3. u Kode Pemrograman <i>Front-end</i> , <i>src-components-atomic-dashboard-transaction-control-(transaction-control.js)</i> – Tampilan Halaman Kontrol Transaksi .....	115

Lampiran 3. v Kode Pemrograman <i>Front-end</i> , src-components-atomic-dashboard-rute control - (rute-control.js) – Tampilan Halaman Kontrol Rute.....	116
Lampiran 3. w Kode Pemrograman <i>Front-end</i> , src-components-atomic-dashboard-news control - (news-control.js) – Tampilan Halaman Kontrol Berita.....	118
Lampiran 3. x Kode Pemrograman <i>Front-end</i> , src-components-atomic-dashboard-statistic control - (statistic-control.js) – Tampilan Halaman Kontrol Statistik ...	121
Lampiran 3. y Kode Pemrograman <i>Front-end</i> , src-components-footer-(footer.js) – Struktur komponen footer .....	122
Lampiran 4. Kode Pemrograman <i>Back-end Website</i> .....	124
Lampiran 4. a Kode Pemrograman <i>Back-end</i> , (Package.jsON) – Inisiasi Library .....	124
Lampiran 4. b Kode Pemrograman <i>Back-end</i> , (.env) – API Endpoints ENV tersamarkan.....	124
Lampiran 4. c Kode Pemrograman <i>Back-end</i> , controllers-Auth-(auth.controllers.js) - Validasi Data User .....	125
Lampiran 4. d Kode Pemrograman <i>Back-end</i> , controllers-Bus-(bus.controllers.js) - Validasi Data Bus.....	126
Lampiran 4. e Kode Pemrograman <i>Back-end</i> , controllers-dashboard-(dashboard.controllers.js) - Validasi Data dashboard.....	128
Lampiran 4. f Kode Pemrograman <i>Back-end</i> , controllers-news-(news.controllers.js) - Validasi Data Berita. <i>Source Code</i> terlampir hanya bagian awal dan akhir saja. ....	128
Lampiran 4. g Kode Pemrograman <i>Back-end</i> , controllers-route control-(route.controllers.js) - Validasi Data Rute.....	130
Lampiran 4. h Kode Pemrograman <i>Back-end</i> , controllers-statistic-(statistic.controllers.js) - Validasi Data Statistik.....	131
Lampiran 4. i Kode Pemrograman <i>Back-end</i> , controllers-transaction-(transaction.controllers.js) - Validasi Data Transaksi. <i>Source Code</i> terlampir hanya bagian awal dan akhir saja.....	133
Lampiran 4. j Kode Pemrograman <i>Back-end</i> , controllers-(multer.js) - Validasi Data Upload File.....	134



Lampiran 4. k Kode Pemrograman <i>Back-end</i> , routes-Auth-(auth.routes.js) - Navigasi / URL Mapping Informasi <i>signin / signup</i> .....	135
Lampiran 4. l Kode Pemrograman <i>Back-end</i> , routes-Bus-(bus.routes.js) - Navigasi / URL Mapping Bus.....	135
Lampiran 4. m Kode Pemrograman <i>Back-end</i> , routes-dashboard-(dashboard.routes.js) - Navigasi / URL Mapping dashboard .....	136
Lampiran 4. n Kode Pemrograman <i>Back-end</i> , routes-news-(news.routes.js) - Navigasi / URL Mapping Berita .....	137
Lampiran 4. o Kode Pemrograman <i>Back-end</i> , routes-route control-(route.routes.js) - Navigasi / URL Mapping Rute .....	137
Lampiran 4. p Kode Pemrograman <i>Back-end</i> , routes-statistic-(statistic.routes.js) - Navigasi / URL Mapping Statistik.....	138
Lampiran 4. q Kode Pemrograman <i>Back-end</i> , routes-transaction-(transaction.routes.js) - Navigasi / URL Mapping Transaksi .....	139
Lampiran 4. r Kode Pemrograman <i>Back-end</i> , (index.js) – Entrypoint Utama....	139
Lampiran 5. Bukti Responden Pengujian <i>System Usability Scale</i> .....	142
Lampiran 5. a Bukti Responden Secara Tatap Muka.....	142
Lampiran 5. b Bukti Responden Via Sosial Media WhatsApp.....	145
Lampiran 5. c Bukti Responden Via Sosial Media TikTok.....	146
Lampiran 5. d Bukti Responden Via Sosial Media Telegram .....	147
Lampiran 5. e Bukti Responden Via Sosial Media Instagram.....	151
Lampiran 6. Hasil Pengujian <i>Black-Box</i> .....	152
Lampiran 7. Persentase Nilai Setiap Pertanyaan <i>SUS</i> .....	162

## KATA PENGANTAR

Puji serta syukur yang tiada hentinya penulis panjatkan kepada Allah Subhanahu Wa Ta'ala, atas berkat dan limpahan rahmat, kesehatan, dan petunjuk sehingga penulis dapat menyelesaikan tugas akhir ini dengan judul “SISTEM INFORMASI *BUS ONLINE TICKETING* BERBASIS *WEB* PADA PT. BORLINDO MANDIRI JAYA”. Penyusunan tugas akhir ini merupakan salah satu syarat kelulusan pada pendidikan strata satu (S1) pada Departemen Teknik Elektro Fakultas Teknik Universitas Hasanuddin sehingga penulisan tugas akhir ini penulis laksanakan guna menyelesaikan studi sarjana.

Tugas akhir ini dilakukan penelitian yang bertujuan membuat sistem informasi yang membantu operator dalam manajemen *database* dan juga sistem informasi yang memiliki perbedaan diantara sistem yang ada sebelumnya menggunakan *framework NextJS* dan *database MongoDB* terutama dalam rendering halaman dengan pengujian pada platform *PageSpeed Insights* serta melihat *feedback* masyarakat terhadap sistem yang telah dibuat menggunakan metode pengujian *System Usability Scale*.

Dalam penyelesaian tugas akhir, penulis menyadari banyaknya dukungan dan bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih sebesar-besarnya kepada:

1. Orang tua yang sangat penulis sayangi, Ibu Sitti Johar, SE dan Bapak Siegfried Selamat Santosos Widjaja (Alm.), serta keluarga besar yang tak henti-hentinya memberikan semangat, dukungan, dan do'a sehingga tugas akhir ini dapat terselesaikan dengan baik
2. Bapak Dr.Eng. Wardi, S.T., M.Eng. dan Ibu Andini Dani Achmad, ST., MT. selaku dosen pembimbing penulis yang telah meluangkan waktunya untuk bertukar pikiran serta memberikan inspirasi, masukan, dan evaluasi dalam penyelesaian tugas akhir ini

3. Bapak Prof. Dr. Ir. H. Andani Achmad, MT. dan Ida Rachmaniar Sahali, ST., MT. selaku dosen penguji yang telah meluangkan waktunya untuk memberikan saran, koreksi, dan arahan dalam penyelesaian tugas akhir penulis
4. Bapak Royi Sumule dan Bapak Hermon Sumule selaku Pemilik Utama dari PT. Borlindo Mandiri Jaya beserta jajarannya yang telah mengizinkan penulis untuk melakukan penelitian serta pengambilan data pada perusahaan PT. Borlindo Mandiri Jaya tanpa hambatan dan kendala sedikit pun.
5. Seluruh dosen pengajar dan pegawai Departemen Teknik Elektro Universitas Hasanuddin atas bimbingan, didikan, kemudahan, dan bantuan yang telah diberikan kepada penulis selama menempuh perkuliahan di Universitas Hasanuddin.
6. Sahabat laboratorium Jaringan Komputer angkatan 2020 yang telah menjadi wadah untuk melampiaskan keluh, kesah, canda dan tawa selama berjalannya perkuliahan serta dukungan selama penyelesaian tugas akhir
7. Rekan-rekan laboratorium Antena angkatan 2020 yang telah mengizinkan penulis untuk bergabung untuk bertukar ilmu, mendapatkan banyak pengalaman dan pelajaran baru sekaligus berbagi tempat selama laboratorium penulis tidak dapat digunakan kala itu
8. Seluruh rekan-rekan hebat PROCEZ20R yang telah menemani penulis dalam berjuang menempuh perkuliahan sejak awal dan selalu ada sebagai wadah untuk berbagi wawasan, cerita, canda, dan tawa.
9. Tidak lupa penulis juga sangat berterima kasih kepada para responden yang meluangkan waktu untuk mengisi kuesioner sebagai bentuk pengujian dalam penelitian ini.

Gowa, 24 Juli 2024  
Penulis,

Bryan Valentino

## **BAB I**

### **PENDAHULUAN**

#### **I.1 Latar Belakang**

Bus merupakan salah satu moda transportasi umum yang akan memiliki peminat di setiap waktu, terlebih jika memasuki musim mudik. Menurut informasi yang diedarkan oleh Divisi Hubungan Masyarakat Kepolisian Republik Indonesia mengungkapkan bahwa Bus diminati oleh 22,77 juta pemudik (18,39%), Kereta Api 14,47 juta pemudik (11,69%), Angkutan Penyebrangan 6,67 juta pemudik (5,39%), Pesawat Terbang 6,19 juta pemudik (5,00%), Angkutan Laut 1,66 juta pemudik (1,34%) (Divisi Humas POLRI, 2023). Oleh karena hal tersebut, tidak sedikit Perusahaan Otobus (PO) yang berlomba-lomba untuk menambah jumlah armada yang mereka miliki. Data per 16 Desember 2023 menyebutkan bahwa terdapat sebanyak 266.635 total keseluruhan bus. Bila dibandingkan dengan jumlah total bus pada 9 Juli 2023 yang hanya berjumlah 260.720, menandakan terdapat selisih sebanyak 5.915 atau mengalami kenaikan sebesar 2,27% Data tersebut merupakan akumulasi dari total bus yang berada di delapan pulau, yaitu Sumatera, Jawa, Kalimantan, Sulawesi, Bali, Nusa Tenggara, Maluku, dan Papua. Dimana Sulawesi mendapatkan posisi kelima dengan total kepemilikan bus sebanyak 8.436 (Sari & Ferdian, 2023). Situasi ini menimbulkan tantangan serius bagi pemerintah dan operator bus. Ada kebutuhan mendesak untuk meningkatkan infrastruktur, memperluas jaringan bus, meningkatkan frekuensi layanan, dan mengembangkan sistem transportasi yang lebih terintegrasi.

Provinsi Sulawesi Selatan merupakan sebuah provinsi yang memiliki kabupaten maupun kota-kota besar yang sering dikunjungi oleh masyarakat lokal, domestik, maupun mancanegara. Kunjungan ini dapat berupa kunjungan wisata, urusan pekerjaan, dan juga beberapa diantaranya yang ingin pulang ke kampung halaman. Salah satu moda transportasi yang sering digunakan oleh masyarakat di Sulawesi Selatan ini adalah Bus. Bus termasuk kendaraan darat besar yang dimaksudkan untuk mengangkut hingga 30 penumpang.

Terdapat banyak jenis bus yang sering kita temui seperti *medium bus* hingga *big bus*. Bukan hanya jenis bus yang beragam, namun layanan yang ditawarkan oleh pihak perusahaan juga beragam dan sesuai dengan jenis busnya. Misalkan bus besar, disediakan untuk jarak jauh seperti Antar Kota Dalam Provinsi (AKDP) dan Antar Kota Antar Provinsi (AKAP). Bus dapat digunakan untuk perjalanan perkotaan, jarak jauh, sekolah, persewaan, atau pariwisata. Pada umumnya, penjualan tiket bus masih dilakukan dengan cara tradisional, yaitu dengan mendatangi perwakilan bus ataupun memesan tiket bus langsung di terminal bus (Batubara et al., 2022).

Seiring berkembangnya zaman, perusahaan bus di Sulawesi Selatan ini berlomba-lomba untuk membuat perusahaan semakin berkembang agar tetap diminati oleh masyarakat. Masyarakat peminat bus di Sulawesi Selatan, bukan hanya masyarakat lokal yang memiliki banyak waktu dan kesempatan untuk memesan tiket bus. Namun, terdapat juga masyarakat yang tidak memiliki waktu luang untuk memesan tiket. Contoh *kasus* seperti apabila masyarakat tersebut masih dalam perjalanan pesawat tetapi sudah harus memesan tiket terlebih dahulu sebelum pesawat yang ia naiki sampai di tujuan. Oleh karena hal tersebut, banyak oknum yang memanfaatkan kesempatan untuk membuat reservasi tiket palsu yang disebar di beberapa sosial media sehingga tidak sedikit masyarakat yang terkena penipuan tersebut. Salah satu perusahaan yang merasakan keresahan tersebut ialah PT. Borlindo Mandiri Jaya.

PT. Borlindo Mandiri Jaya merupakan salah satu perusahaan yang bergerak pada bidang transportasi bus. Perusahaan ini didirikan oleh seorang pebisnis asal Makale, Tana Toraja pada 5 Juni 2017. Walaupun perusahaan ini baru berdiri selama kurang lebih 7 tahun, namun telah memiliki 58 unit bus dan 15 rute perjalanan hingga saat ini. Dengan banyaknya rute yang dimiliki, secara otomatis jumlah penumpang dari bus ini pasti tidak sedikit. Berdasarkan data yang didapatkan dari pihak Borlindo, diketahui bahwa rata-rata penumpang pada musim reguler sebanyak 750 orang, dan pada musim mudik rata-rata penumpang di klaim mencapai 1000 orang lebih.

Dengan banyaknya *traffic* penumpang tersebut pasti akan menimbulkan beberapa masalah, mulai dari kesulitan penumpang dalam memesan tiket dan pihak administrasi yang kewalahan akibat lonjakan penumpang yang ada. Solusi terbaik untuk mengatasi beberapa keresahan, mulai dari penipuan pemesanan tiket, penumpang yang kesusahan reservasi dan administrasi yang kewalahan ialah pengadaan *website* resmi oleh perusahaan. *Website* merupakan halaman berisi informasi yang dapat diakses oleh masyarakat umum dimanapun dan kapanpun. Dalam *website* tersebut, calon penumpang dapat melihat berbagai paparan informasi perusahaan, pemesanan tiket, hingga melakukan komplain secara *online*. Oleh karena itu, dalam rangka penelitian ini penulis akan merancang sebuah sistem informasi pada PT. Borlindo Mandiri Jaya yang berbasis *Website*.

Bahasa pemrograman yang digunakan yaitu *JavaScript* dengan alasan memiliki sifat yang ringan sehingga membuat *website* lebih dinamis, *NextJS* sebagai kerangka kerja (*framework*) *Website* yang mana memiliki beberapa keunggulan seperti konsep *SSR* (*Server-side rendering*) dengan sifat *seofriendly* serta *automatic code splitting* sebagai pemecah kode, sehingga browser dapat menampilkan halaman *website* yang lebih cepat. Performa *website* akan lebih baik dan memiliki kemudahan dalam *deploy* dan *setup* sebuah *project*. Kemudian *database* menggunakan *MongoDB* dengan karakteristik *NoSQL* dengan format *JSON*, dimana data disimpan dalam bentuk dokumen dan tidak memiliki relasi antar tabel (Sanjaya et al., 2023).

Berdasarkan latar belakang di atas, maka dibuatlah skripsi yang berjudul “**Sistem Informasi Bus Online Ticketing Berbasis Web Pada PT. Borlindo Mandiri Jaya**”. Dengan adanya sistem informasi yang dirancang ini, diharapkan dapat memberikan kemudahan bagi penumpang dalam melakukan reservasi tiket dan terhindar dari penipuan dengan modus agen palsu serta kemudahan bagi pihak PT. Borlindo Mandiri Jaya dalam manajemen data penumpang,

## I.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka didapatkan beberapa rumusan masalah sebagai berikut:

1. Bagaimana kinerja yang didapatkan apabila sistem informasi *bus online ticketing* dibuat menggunakan *framework NextJs* dan *database MongoDB*?
2. Bagaimana sistem informasi *bus online ticketing* yang dibuat dapat dikatakan berjalan sesuai dengan fungsi yang diinginkan?
3. Bagaimana hasil pengukuran metrik *Core Web Vitals* yang didapatkan dari sistem informasi *bus online ticketing* yang telah dibuat?
4. Bagaimana tingkat *usability* yang didapatkan dari sistem informasi *bus online ticketing* yang telah dibuat?

## I.3 Tujuan Penelitian

Berdasarkan rumusan masalah dalam penelitian ini, maka dapat ditentukan tujuan penelitian sebagai berikut:

1. Untuk mengetahui hasil kinerja dari *framework NextJs* dan *database MongoDB* dalam merancang sistem informasi *bus online ticketing*.
2. Untuk mengetahui sistem informasi *bus online ticketing* yang dibuat dapat berjalan sesuai fungsi menggunakan metode pengujian *Black-Box*.
3. Untuk mengetahui hasil pengukuran *Largest Contentful Paint (LCP)* dan *Cumulative Layout Shift (CLS)* sebagai parameter *Core Web Vitals* dari *website* yang dibuat menggunakan *software PageSpeed Insights*.
4. Untuk mengetahui hasil pengujian *Sistem Usability Scale (SUS)* sebagai tolak ukur tingkat *usability* sistem informasi *bus online ticketing*.

## I.4 Manfaat Penelitian

Manfaat yang diharapkan pada penelitian “Sistem Informasi *Bus Online Ticketing* Berbasis *Web* Pada PT. Borlindo Mandiri Jaya” ini sebagai berikut:

1. Bagi Penulis  
Melatih penulis dalam menganalisis sebuah masalah, mendesain, merancang sebuah sistem informasi mulai dari *Front-end*, *Back-end*, hingga *deploying*.

## 2. Bagi Perusahaan

Perusahaan dapat dikenal lebih luas oleh masyarakat karena informasi yang dipaparkan dalam sistem informasi. Selain itu, perusahaan dapat dengan mudah memanajemen data penumpang karena telah berbasis digital.

## 3. Bagi Penumpang

Penumpang dapat melakukan pemesanan tiket *online* serta pembayaran dengan metode transfer dimanapun dan kapanpun dengan praktis dan cepat tanpa khawatir terkena modus penipuan.

### **I.5 Ruang Lingkup**

Dalam penelitian ini, pembahasannya dibatasi pada:

1. Penelitian ini hanya dilakukan pada PT. Borlindo Mandiri Jaya.
2. Penelitian ini hanya menguji fungsionalitas sistem menggunakan metode pengujian *Black-box*.
3. Penelitian ini hanya menguji tolak ukur tingkat *usability* sistem informasi dengan metode pengujian *Sistem Usability Scale (SUS)*.
4. Penelitian ini hanya membahas pengaruh *framework NextJS* pada *Largest Contentful Paint (LCP)* dan *Cumulative Layout Shift (CLS)* sebuah *website* menggunakan *software PageSpeed Insights*.
5. Penelitian ini terbatas pada beberapa fitur yang disediakan hanya untuk keperluan pengujian saja.



## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1 Sistem Informasi**

##### **II.1.1. Sistem**

Sistem merupakan sebuah pendekatan prosedur dan pendekatan komponen. Pendekatan prosedural didefinisikan sebagai sekumpulan prosedur dengan satu tujuan. Sistem akuntansi, misalnya, adalah kumpulan prosedur untuk penerimaan kas, pengeluaran kas, penjualan, dan buku besar. Metode komponen dicirikan sebagai kumpulan komponen yang saling berhubungan yang menciptakan satu kesatuan untuk mencapai tujuan tertentu, seperti sistem komputer sebagai kumpulan perangkat keras dan perangkat lunak. Sistem sendiri, terdiri dari dua komponen penting yaitu struktur dan proses. Pendekatan prosedur akan lebih cocok untuk menggambarkan sistem yang lebih menekankan pada sebuah proses. Dalam sistem yang bentuknya fisik dapat kita lihat dengan jelas, maka pendekatan komponen dapat digunakan untuk menggambarkan sistemnya (Karmila, 2019).

Sistem adalah sebuah jaringan daripada element-element yang memiliki hubungan yang saling membentuk satu kesatuan untuk melaksanakan tujuan pokok yang sama dari sistem tersebut. Untuk mengetahui hal tersebut dikatakan sistem atau bukan, dapat kita lihat dari bentuk ciri-cirinya. Ada beberapa rumusan mengenai ciri-ciri dari sistem ini yang pada dasarnya satu sama lain saling melengkapi. Pada umumnya ciri-ciri sistem ini adalah bertujuan, punya batas, terbuka, tersusun dari sub sistem, ada yang saling berkaitan dan tergantung merupakan kebulatan yang sistematis (Nopriandi, 2018). Terdapat beberapa karakteristik sistem diantaranya:

1. Komponen atau elemen (*Component*). Suatu sistem terdiri dari komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan.

2. Batas Sistem (*boundary*). Batas sistem (*boundary*) merupakan daerah yang membatasi antara satu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Dengan kata lain, batas sistem merupakan ruang lingkup atau scope dari sistem atau subsistem itu sendiri.
3. Lingkungan Luar Sistem (*environment*). Lingkungan luar sistem adalah segala sesuatu diluar batas sistem yang satu dengan yang mempengaruhi operasi suatu sistem. Lingkungan luar sistem dapat bersifat menguntungkan atau merugikan.
4. Penghubung Sistem (*interface*). Penghubung sistem merupakan suatu media (penghubung) antara satu subsistem dengan subsistem lainnya yang membentuk satu kesatuan, sehingga sumber-sumber daya mengalir dari subsistem yang satu ke yang lainnya.
5. Masukan (*input*). *Input* adalah sesuatu yang dimasukkan kedalam suatu sistem yang dapat berupa masukan.
6. Luaran (*output*). Merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi luaran yang berguna, luaran juga merupakan tujuan akhir dari sistem.
7. Pengolah (*process*). Suatu sistem mempunyai bagian pengolah yang akan mengubah *input* menjadi *output*.
8. Sasaran (*objective*). Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang dihasilkan sistem. Sistem dikatakan berhasil bila mengenai sasaran atau tujuannya (Andrianof, 2018).

### **II.1.2. Informasi**

Sumber dari informasi adalah data. Data adalah fakta atau kenyataan yang menggambarkan suatu kejadian-kejadian yang mempunyai arti tersendiri. Informasi adalah data yang telah diproses sedemikian rupa, sehingga memiliki arti yang lebih bermanfaat bagi penggunaannya. Kualitas dari suatu informasi dilihat dari tiga hal:

1. Akurat. Informasi harus bebas dari kesalahan-kesalahan. Akurat harus mencerminkan maksud dan penyampaiannya harus akurat, dari sumber sampai penerima informasi.
2. Tepat Waktu. Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah tidak berlalu tidak akan mempunyai nilai lagi karena informasi merupakan landasan didalam pengambilan keputusan. Bila keputusan terlambat, maka akan berakibat fatal bagi organisasi atau perusahaan.
3. Relevan. Informasi harus mempunyai manfaat bagi pemakainya. Relevansi informasi untuk tiap-tiap orang berbeda (Andrianof, 2018).

### **II.1.3. Sistem Informasi**

Sistem informasi terdiri dari beberapa komponen. Pengertian sistem informasi adalah sistem informasi yang diciptakan oleh para pengolahan data anak, data kegiatan, data sponsor, data pemeriksaan kesehatan, pemeriksaan sosio emosi, data perpindahan anak, data pejabat, data tutor, bantuan pelayanan dan data inventarisasi barang analisis dan manajer guna melaksanakan tugas khusus tertentu yang sangat esensial bagi berfungsinya organisasi.

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Rancangan sistem Informasi adalah merancang atau membuat sistem baru yang diterapkan untuk mengatasi masalah yang lama. Perancangan sistem dapat diartikan sebagai tahap setelah analisis dari sirklus pengembangan sistem, pendefinisian dari kebutuhankebutuhan fungsionalis, persiapan untuk rancangan bangunan implementasi, menggambarkan bagaimana suatu sistem dibentuk (penggambaran, perencanaan, pembatasan sketsa) termasuk mengkonfigurasi komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem. (Nopriandi, 2018)

## II.2 Reservasi

Reservasi atau pemesanan dalam bahasa Inggris adalah *reservation* yang berasal dari kata "*to reserve*" yaitu menyediakan atau mempersiapkan tempat sebelumnya. Sedangkan *reservation* yaitu pemesanan suatu tempat fasilitas. Jadi secara umum *reservation* yaitu pemesanan fasilitas yang diantaranya hotel, akomodasi, meal, seat pada pertunjukan, pesawat terbang, kereta api, bus, hiburan, *night club*, *discoutegue* dan sebagainya. Kata *reservation* atau pemesanan dalam dunia pariwisata disebut juga *booking* (Anharudin & Nasser, 2020).

Reservasi memberikan arti pesan, interaksi berupa komunikasi baik yang dilakukan secara lisan maupun tulisan dari seseorang menuju orang lain. Berikut ini manfaat reservasi adalah:

1. Dengan tujuan dapat menarik dan mendapatkan pengunjung sebanyak dan sesering mungkin serta mengusahakan keramah-tamahan dan berbagai fasilitas yang mendukung kemudahan bagi pelanggan.
2. Informasi biaya sangat penting bagi pengunjung untuk melakukan perbandingan dan memprediksi biaya yang dikeluarkan.
3. Dari sisi penyelenggara, dapat segera tanggap dengan adanya pemesanan dan mempersiapkan sambutan pengunjung, mempersiapkan agenda, paket wisata dan persiapan dari permintaan dari konsumen (Sutanto et al., 2020).

## II.3 Website

*Web* atau Situs *Website* atau situs dapat diartikan sebagai kumpulan halaman halaman yang digunakan untuk menampilkan informasi, teks, gambar diam atau bergerak, animasi, suara, dan atau gabungan dari semuanya itu, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling berkait dimana masing masing dihubungkan dengan jaringan jaringan halaman/*hyperlink*.

### **II.3.1. Domain**

Nama *domain* atau biasa disebut dengan *Domain Name* atau *URL* adalah alamat unik di dunia internet yang digunakan untuk mengidentifikasi sebuah *website*, atau dengan kata lain *domain name* adalah alamat yang digunakan untuk menemukan sebuah website pada dunia internet. Contoh <http://www.unm.ac.id/> dan <http://www.detik.com/>. Nama *domain* diperjual-belikan secara bebas di internet dengan status sewa tahunan. Nama *domain* sendiri mempunyai identifikasi ekstensi/akhiran sesuai dengan kepentingan dan lokasi keberadaan *website* tersebut, contoh nama *domain* berekstensi lokasi negara Indonesia adalah *co.id* (untuk nama *domain website* perusahaan), *ac.id* (nama *domain website* pendidikan), *go.id* (nama *domain website* instansi pemerintahan), *or.id* (nama *domain website* organisasi) (Jaya, 2017).

### **II.3.2. Web Hosting**

*Hosting* dapat diartikan sebagai ruangan yang terdapat dalam harddisk tempat menyimpan berbagai data, *file*, gambar dan lain sebagainya yang akan ditampilkan di *website*. Besarnya data yang bisa dimasukkan tergantung dari besarnya *web hosting* semakin besar pula data yang dapat dimasukkan dan ditampilkan dalam *website*. *Web Hosting* juga juga diperoleh dengan menyewa besarnya *hosting* ditentukan ruangan harddisk dengan ukuran MB (*Mega Byte*) atau GB (*Giga Byte*). Lama penyewaan *web hosting* rata rata dihitung per tahun. Penyewaan *hosting* dilakukan dari perusahaan perusahaan penyewa *web hosting* yang banyak dijumpai baik di Indonesia maupun luar negeri (Jaya, 2017).

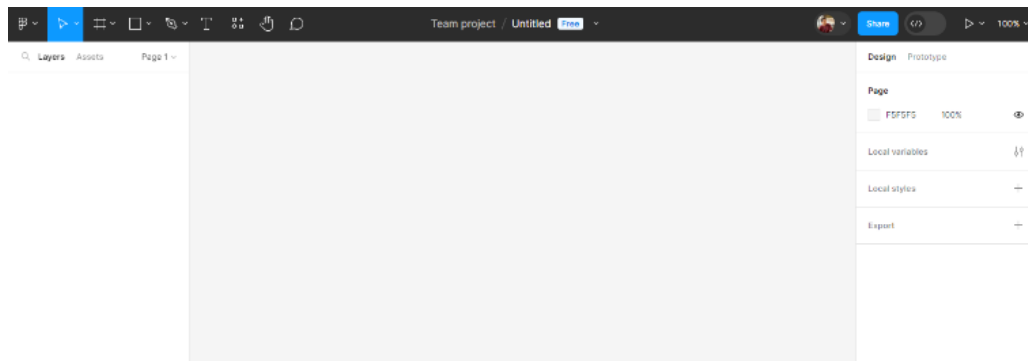
### **II.3.3. Bahasa program**

Bahas program adalah bahasa yang digunakan untuk menerjemahkan setiap perintah dalam *website* pada saat diakses. Jenis bahasa program sangat menentukan statis, dinamis, atau interaktifnya sebuah *website*. Semakin banyak ragam bahasa program yang digunakan maka akan terlihat *website* semakin dinamis dan interaktif serta terlihat bagus. Beragam bahasa program saat ini telah hadir untuk mendukung kualitas *website*. Jenis jenis bahasa program yang banyak dipakai para desainer

*website* antara lain HTML, ASP, PHP, JSP, Java Scripts, Java Applets, dan sebagainya. Bahasa dasar yang yang dipakai setiap situs adalah HTML, sedangkan PHP, ASP, JSP dan lainnya merupakan bahasa pendukung yang bertindak sebagai pengatur dinamis, dan interaktifnya situs. Bahasa program ASP, PHP, JSP atau lainnya bisa dibuat sendiri. Bahasa program ini biasanya digunakan untuk membangun portal berita, artikel, forum diskusi, buku tamu, anggota organisasi, *email*, *mailing list*, dan lain sebagainya yang memerlukan update setiap saat (Jaya, 2017).

#### **II.3.4. Desain Website.**

Setelah melakukan penyewaan *domain name* dan *web hosting* serta penguasaan bahasa program, unsur *website* yang penting adalah dan utama adalah desain. Desain *website* menentukan kualitas dan keindahan sebuah *website*. Untuk membuat *website* biasanya dapat dilakukan sendiri atau menyewa jasa *website designer*. Perlu diketahui bahwa kualitas situs sangat ditentukan oleh kualitas designer (Jaya, 2017). Figma adalah salah satu design *tool* yang biasanya digunakan untuk membuat tampilan aplikasi *mobile*, *desktop*, *website* dan lain-lain. Figma bisa digunakan di sistem operasi windows, linux ataupun mac dengan terhubung ke internet. Figma memiliki keunggulan yaitu untuk pekerjaan yang sama dapat dikerjakan oleh lebih dari satu orang secara bersama-sama walaupun ditempat yang berbeda. Hal tersebut bisa dikatakan kerja kelompok dan karena kemampuan aplikasi figma tersebut lah yang membuat aplikasi ini menjadi pilihan banyak *UI/UX designer* untuk membuat *prototype website* atau aplikasi dengan waktu yang cepat dan efektif. (Al-Faruq et al., 2022)



Gambar II. 1 Figma (Figma, n.d.)

Keberadaan situs tidak ada gunanya dibangun tanpa dikunjungi atau dikenal oleh pengunjung internet. Untuk mengenalkan situs kepada masyarakat memerlukan apa yang disebut publikasi atau promosi. Publikasi situs di masyarakat dapat dilakukan dengan berbagai cara seperti dengan pamflet, selebaran, baliho dan lain sebagainya, tapi cara ini bias dikatakan masih kurang efektif dan sangat terbatas. Cara yang biasanya dilakukan dan paling efektif dengan tak terbatas ruang atau waktu adalah publikasi langsung di internet melalui search engine seperti *yahoo*, *google*, dan sebagainya. Cara publikasi di search engine ada yang gratis dan ada pula yang membayar, yang gratis biasanya terbatas dan cukup lama untuk bias masuk dan dikenali di search engine terkenal seperti yahoo dan google. Cara efektif publikasi adalah dengan membayar walaupun harus sedikit mengeluarkan biaya, akan tetapi situs dapat cepat masuk ke search engine dan dikenal oleh pengunjung.

Untuk mendukung kelanjutan dari situs diperlukan pemeliharaan setiap waktu sesuai yang diinginkan seperti penambahan informasi, berita, artikel, link, gambar dan lain sebagainya, tanpa pemeliharaan yang baik situs akan terkesan membosankan atau monoton juga akan segera ditinggalkan pengunjung. Pemeliharaan situs dapat dilakukan per periode tertentu seperti tiap hari, tiap minggu, atau sebulan sekali secara rutin atau secara periodic tergantung kebutuhan. Pemeliharaan rutin biasanya dipakai oleh situs situs berita, penyedia artikel, organisasi atau lembaga pemerintah, sedangkan pemeliharaan periodic biasanya untuk situs situs penjualan, dan sebagainya. (Jaya, 2017)

### **II.3.5. Basis Data**

Basis data adalah satu kumpulan data terhubung (*interrelated data*) yang disimpan secara bersama-sama pada suatu media. Data disimpan dengan cara-cara tertentu sehingga mudah digunakan atau ditampilkan kembali. Data dapat digunakan oleh satu atau lebih program-program aplikasi secara optimal serta dapat disimpan tanpa mengalami ketergantungan dengan program yang menggunakannya (Silalahi, 2018).

#### **II.3.5.1. Basis Data *SQL* (Relasional)**

Basis data relasional adalah koleksi data item yang diorganisasikan sebagai seperangkat tabel yang terdeskripsi secara formal dimana data dapat diakses atau *disusun* kembali dengan banyak cara tanpa harus mengorganisasikan kembali tabel-tabel basis data. Basis data relasional mengorganisasikan data ke satu atau lebih tabel (atau relasi) yang berisi kolom dan baris, dengan kunci yang unik untuk setiap barisnya. Relasi bermanfaat untuk menjaga kelompok data sebagai koleksi yang tetap dengan bantuan tabel data yang berisi informasi yang terstruktur, menghubungkan semua masukan dengan cara memberikan nilai ke atribut (Silalahi, 2018).

#### **II.3.5.2. Basis Data *NoSQL* (Non-Relasional)**

Perusahaan-perusahaan *Web 2.0* yang besar telah mengembangkan atau mengadopsi berbagai jenis basis data *NoSQL* untuk data mereka yang terus bertambah dan untuk kebutuhan infrastruktur mereka, misalnya Google (BigTable), LinkedIn (Voldemort), Facebook (Cassandra), Amazon (Dynamo), dan sebagainya. Penggunaan *NoSQL* memungkinkan para pengembang aplikasi untuk mengembangkan tanpa harus mengkonversi struktur di memori ke struktur relasional. Basis data *NoSQL* dirancang untuk mengatasi isu “Big Data” dengan memanfaatkan mesin-mesin yang terdistribusi. Basis data *NoSQL* juga sering disebut basis data cloud yang dikembangkan untuk mengatasi permasalahan basis data relasional yang mengalami masalah performa dalam pemrosesan data tidak terstruktur yang terus bertambah secara eksponensial, seperti dokumen, email,



multimedia atau media sosial. *NoSQL* menyimpan dan mengambil data dalam format-format yang berbeda. Terdapat 4 kategori basis data *NoSQL*, yaitu:

1. *Key value store*, merupakan penyimpanan data *NoSQL* yang paling sederhana untuk digunakan dari perspektif API. Penyimpanan data berupa blob yang hanya menyimpan begitu saja tanpa peduli atau perlu mengetahui apa yang disimpan.
2. *Column-family*. Sebuah sistem sparse matrix yang menggunakan baris dan kolom sebagai kunci.
3. *Graph databases*. Data disimpan menggunakan struktur grafis dengan nodes (entitas), properties (informasi tentang entitas) dan garis (hubungan antar entitas).
4. *Document stores*, merupakan mekanisme untuk menyimpan struktur data hierarki langsung ke basis data (Silalahi, 2018).

### **II.3.6. Frontend dan Backend**

#### **II.3.6.1. Frontend**

*Frontend* adalah bagian dari aplikasi *web* atau situs *web* yang berinteraksi langsung dengan pengguna. Hal ini juga dikenal sebagai pengembangan sisi klien. Pengembangan *frontend* melibatkan pembuatan antarmuka pengguna situs *web* atau aplikasi *web*, seperti tombol, tautan, animasi, dan lainnya (Bui, 2023).

Pengembang *frontend* akan menggunakan teknologi *web* untuk membangun situs *web* atau aplikasi *web* seperti HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), dan JavaScript. Tujuan dari pengembangan *frontend* adalah untuk menciptakan pengalaman yang menarik secara visual dan interaktif bagi pengguna. Beberapa komponen utama dari *frontend* antara lain:

- a. HTML. Ini adalah blok bangunan dasar untuk pengembangan *frontend* - digunakan untuk membuat struktur dan konten halaman *web*.
- b. CSS. Ini digunakan untuk menata konten HTML dan membuat elemen visual, seperti font, warna, dan tata letak.

- c. JavaScript. Ini adalah bahasa pemrograman skrip. Bahasa ini digunakan untuk menambahkan interaktivitas dan fungsionalitas dinamis pada halaman *web*, seperti animasi, validasi *input* pengguna, dan manipulasi data.
- d. Kerangka kerja dan pustaka *frontend* - Ini adalah komponen dan alat yang sudah dibuat sebelumnya yang menyederhanakan dan mempercepat pengembangan *frontend*, seperti *Angular*, *Vue.js*, *React*, *NextJS*, *Bootstrap*, *Materialize*, dan *Tailwind CSS* (Bui, 2023).

### **II.3.6.2. Backend**

Pengembangan *backend* dikenal sebagai pengembangan sisi server. Ini adalah bagian penting dalam membuat aplikasi *web*. Hal ini melibatkan perancangan dan implementasi arsitektur aplikasi, membuat API (*Application Programming Interface*) yang memungkinkan bagian depan aplikasi berinteraksi dengan bagian belakang, dan mengelola penyimpanan data dan konten. Berbeda dengan pengembangan *frontend*, yang berfokus pada aspek yang berhadapan dengan pengguna dari situs *web* atau aplikasi, pengembangan *backend* berkaitan dengan fungsionalitas yang mendasari yang mendukung pembangunan *frontend*. Pengembangan *frontend* dan *backend* merupakan komponen penting dalam membangun aplikasi *web* atau situs *web* yang sukses (Bui, 2023).

### **II.3.7. NextJS**

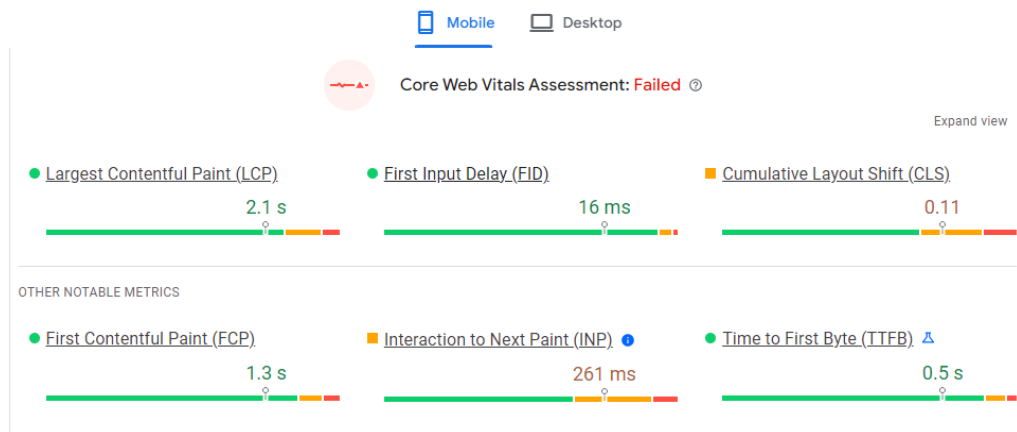
*NextJS* adalah *framework frontend React* yang bersifat open source. Berdasarkan dokumentasinya, *NextJS* adalah *framework* baru yang didesain untuk mengatasi keterbatasan *React* dalam membangun *server-side* atau *rendering server-side* secara penuh. *NextJS* dibangun berdasarkan library *React*, yang berarti *framework* ini memanfaatkan *React* dan menambahkan beberapa fitur. Semua paket dan *file* konfigurasi dibundel dengan rapi dan logis menggunakan *NextJS*. *Framework* ini berbeda dari *framework* aplikasi *web* lainnya karena menawarkan kepada para pengembang sebuah lokasi untuk membangun kode *frontend* dan *backend* untuk sebuah aplikasi. Akibatnya, hal ini meringankan beban kerja pengembang dan membantu dalam pengiriman produk jadi yang paling cepat. Ini

dirancang untuk memungkinkan pengembang membuat aplikasi *React* yang dirender di server dengan mudah. *NextJS* menawarkan banyak fitur, termasuk *Server-side rendering (SSR)*, Pemisahan kode otomatis, Pembuatan situs statis (*SSG*), Perutean sisi klien (*CSR*), dukungan CSS dan Sass, plugin, dan integrasi.

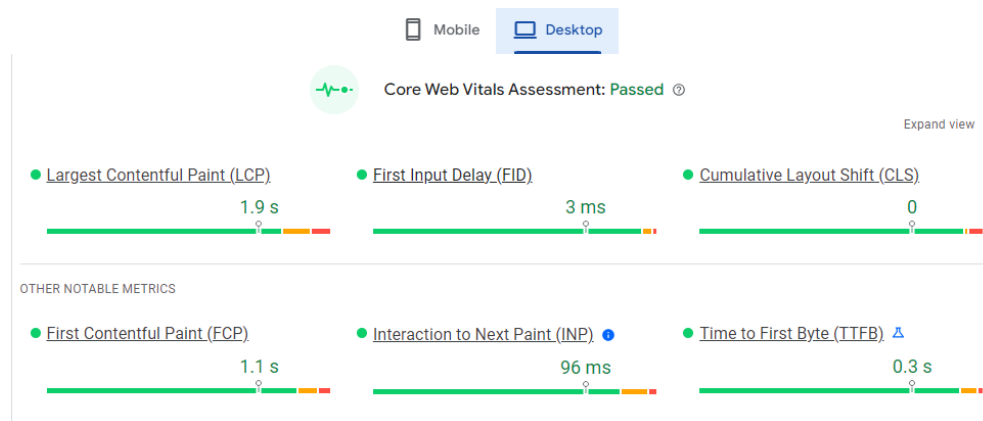
- a. *SSR*: *NextJS* memungkinkan pengembang untuk merender komponen *React* di server, yang dapat meningkatkan waktu pemuatan awal aplikasi. Ini berarti bahwa HTML untuk halaman dibuat di server dan dikirim ke klien, alih-alih klien merender seluruh halaman. Pendekatan ini bisa sangat berguna untuk tujuan SEO.
- b. Pemisahan kode otomatis: Ketika pengguna menavigasi melalui aplikasi, *application NextJS* akan secara otomatis membagi kode JavaScript menjadi beberapa blok kecil sehingga hanya kode yang diperlukan untuk halaman tertentu yang dimuat. Hal ini membantu mengurangi waktu muat awal aplikasi dan meningkatkan kinerja.
- c. *SSG*: *NextJS* mendukung pembuatan situs statis, yang menghasilkan *file HTML* untuk halaman pada saat pembuatan. Ini membantu mengurangi beban server, meningkatkan kinerja, dan lebih aman.
- d. *CSR*: Ketika pengguna mengklik tautan, halaman dimuat secara dinamis tanpa penyegaran halaman penuh. Hal ini dapat meningkatkan pengalaman pengguna dan membuat aplikasi terasa lebih seperti aplikasi *web* tradisional.
- e. Dukungan CSS dan Sass: *NextJS* memiliki dukungan bawaan untuk CSS dan Sass, sehingga memudahkan untuk menata gaya aplikasi. Ini juga mendukung modul CSS, yang membantu mencegah duplikasi nama kelas CSS dan membuatnya lebih mudah untuk mempertahankan gaya Anda.
- f. Dukungan TypeScript: *NextJS* memiliki dukungan bawaan untuk TypeScript, sehingga lebih mudah untuk membangun aplikasi yang aman dan jelas (Bui, 2023).

*NextJS* adalah kerangka kerja pengembangan *web* yang kuat yang menawarkan begitu banyak fitur dan manfaat. *Framework* ini memudahkan untuk membuat aplikasi *web* yang cepat, aman, terukur, dan dioptimalkan yang dapat

menangani beban lalu lintas yang tinggi (Bui, 2023). Sebagai pembuktian untuk pengaruh akibat menggunakan *framework NextJS*, penulis mengambil contoh dari sebuah *website* yang dibangun juga menggunakan *framework* tersebut. Contohnya seperti *website* dengan link URL <https://talenthub.kemnaker.go.id/>. Dimana *website* ini apabila dilakukan pengujian menggunakan *PageSpeed Insight* menghasilkan *Largest Contentful Paint (LCP)* yang tergolong bagus yaitu 2,1 detik untuk tampilan *mobile* dan 1,9 detik untuk tampilan *desktop* serta *Cumulative Layout Shift (CLS)* yang tergolong cukup bagus yaitu 0,11 untuk tampilan *mobile* dan 0 untuk tampilan *desktop* seperti yang ditampilkan oleh Gambar II.2 dan Gambar II.3 di bawah ini.



Gambar II. 2 Nilai *LCP* dan *CLS* pada *Website* tampilan *Mobile* (*PageSpeed Insights*, n.d.)



Gambar II. 3 Nilai *LCP* dan *CLS* pada *Website* tampilan *Desktop* (*PageSpeed Insights*, n.d.)

### II.3.8. *MongoDB*

*MongoDB* adalah contoh terkenal dari aplikasi basis data *NoSQL* berorientasi dokumen sumber terbuka. Berdasarkan situs *webnya*, *MongoDB* memiliki kinerja, fleksibilitas, dan skalabilitas yang sangat baik, dan menyimpan data dalam gaya seperti *JSON*. *MongoDB* dirancang untuk menangani volume data yang sangat besar dan memberikan kemampuan kepada para programmer untuk membuat aplikasi yang dapat mengelola kueri dan struktur data yang kompleks. *MongoDB* juga memungkinkan pengindeksan dan memiliki bahasa kueri yang kuat. Setiap bahasa memiliki dokumen resmi di situs *webnya*, yang membuat *MongoDB* menjadi salah satu *database* paling terkenal di dunia.

*MongoDB* adalah basis data dokumen yang menyediakan performa tinggi dan ketersediaan tinggi, skalabilitas yang mudah. *MongoDB* adalah sebuah basis data open source yang banyak digunakan untuk menangani data yang besar. *MongoDB* memberikan performa yang tinggi karena penggunaan indexing, aggregation, load balancing, dan sebagainya. *MongoDB* dibuat menggunakan bahasa C++ dan dirilis 2009. *MongoDB* juga merupakan basis data berorientasi dokumen yang terdiri dari kumpulan koleksi dan sebuah koleksi terdiri dari kumpulan dokumen. *MongoDB* menyimpan dokumen dalam format BSON (bentuk binary dari *JSON*). BSON mendukung tipe data yang berbeda seperti integer, float, string, Boolean, date, dan sebagainya. *MongoDB* mempunyai fitur-fitur sebagai berikut:

1. *Document Oriented*, *MongoDB* tidak mengambil dan memecah entitas menjadi beberapa struktur relasional, tetapi *MongoDB* menyimpannya dalam jumlah dokumen yang minimal.
2. *Ad Hoc Queries*, *MongoDB* mendukung pencarian berdasarkan *field*, *range queries*, dan *regular expression*. Hasil dari query dapat berupa field-field tertentu dari dokumen, termasuk penggunaan fungsi-fungsi JavaScript.
3. *Indexing*, Setiap field di dokumen *MongoDB* dapat diberi indeks yang menyediakan efisiensi dalam pencarian data.

4. *Replication*, *MongoDB* memberikan ketersediaan yang tinggi untuk kumpulan-kumpulan replika. Sebuah replika berisi dua atau lebih salinan data.
5. *Load Balancing*, Skalabilitas *MongoDB* bersifat horizontal menggunakan sharding. Pengguna memilih sebuah kunci shard, untuk menentukan bagaimana data dalam sebuah koleksi akan didistribusikan. (Silalahi, 2018)

Dengan penggunaan *database MongoDB* dapat mempermudah dalam pengembangan sistem aplikasi karena kedinamisannya sehingga *website* lebih responsif dan interaktif. Sistem dibuat berdasarkan perancangan sistem dan sistem berjalan dengan baik yang dibuktikan dari hasil pengujian *Blackbox testing* (Sanjaya et al., 2023).

### **II.3.9. UI/UX**

*User Interface* dan *User Experience (UI/UX)* memiliki peranan penting dalam pembuatan sebuah aplikasi, karena desain pada sebuah aplikasi harus rapi dan terorganisir sehingga pengguna dapat dengan mudah memakai fitur-fitur yang telah disediakan oleh sebuah aplikasi. Desain *User Interface* dan *User Experience (UI/UX)* juga harus sesuai dengan kebutuhan pengguna dari aplikasi yang akan dibangun mulai dari desain tampilan, fitur-fitur, dan berbagai kebutuhan lainnya (Al-Faruq et al., 2022).

#### **II.3.9.1. User Interface**

*User Interface* merupakan sistem yang digunakan oleh pengguna yang bisa didengar, dilihat bahkan disentuh. Maka dapat disimpulkan bahwa *User Interface* adalah sebuah sistem yang mengatur tampilan antarmuka sekaligus memfasilitasi pengguna untuk berinteraksi dengan sistem (Al-Faruq et al., 2022).

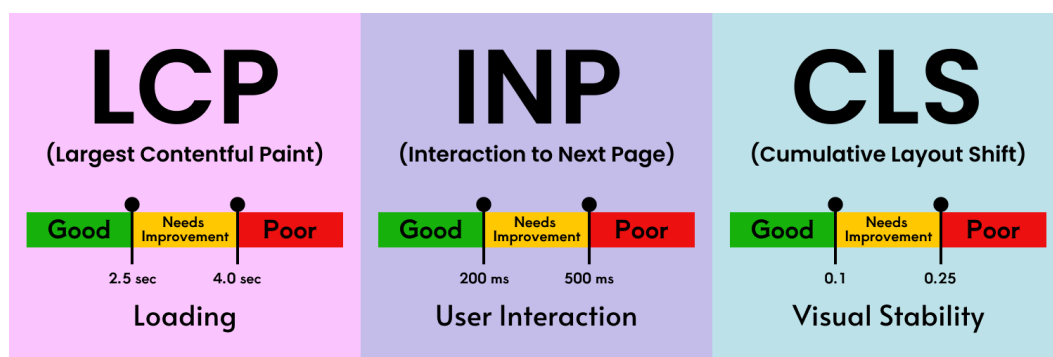
#### **II.3.9.2. User Experience**

*User Experience* adalah sistem yang mengatur pengalaman yang dirasakan pengguna terhadap penggunaan perangkat lunak yang menilai tentang tingkat

kemudahan dan kenyamanan terhadap fungsionalitas dari sebuah perangkat lunak (Al-Faruq et al., 2022).

## II.4 Core Web Vitals

*Core Web Vitals* adalah serangkaian faktor utama yang ditetapkan Google untuk menilai user experience (pengalaman pengguna) sebuah website. *Core Web Vitals* mewakili 3 elemen yang memengaruhi cara pengguna berinteraksi dengan website, yaitu kecepatan loading, interaksi pengguna, dan stabilitas visual. Google menyebutkan bahwa metrik *Core Web Vitals* bisa berkembang seiring waktu, tergantung pada hal-hal apa saja yang dianggap sebagai “pengalaman halaman yang baik” menurut pengguna. Untuk saat ini, pemilik dan developer *website* perlu menilai 3 metrik *Core Web Vitals* berikut ini.



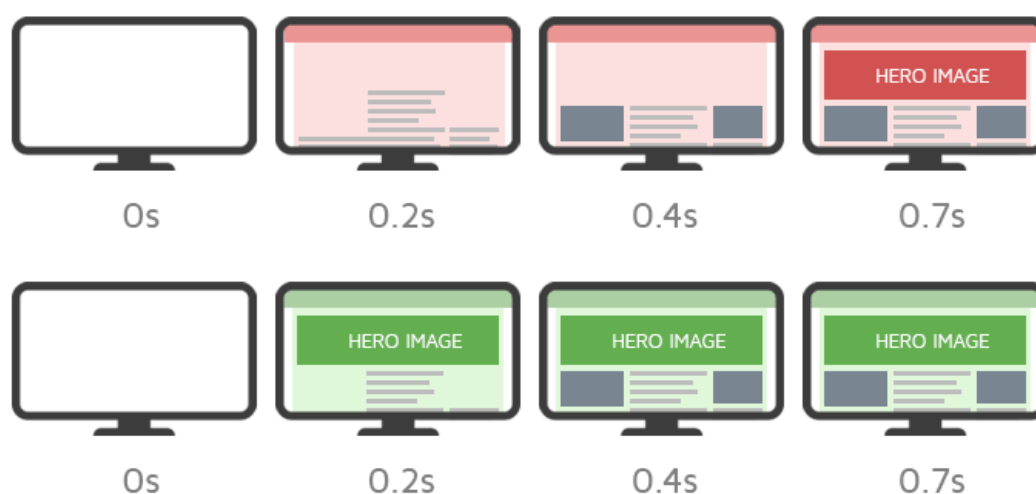
Gambar II. 4 Parameter Pengukuran Web Core Vitals

### II.4.1 Largest Contentful Paint (LCP) – Waktu Loading Halaman

*Largest Contentful Paint (LCP)* mengecek waktu render konten utama halaman ketika halaman mulai dimuat. Konten utama umumnya berupa blok gambar atau teks yang ukurannya paling besar di viewport, yaitu area halaman web yang sedang dilihat di perangkat pengguna. Data *Largest Contentful Paint* utamanya terdiri dari elemen-elemen berikut:

- Elemen gambar
- Teks block-level
- Gambar poster video
- Gambar latar belakang

Pemilik website perlu memastikan bahwa nilai *Core Web Vitals* website miliknya hanya 2,5 detik atau kurang. Dengan begitu, website bisa memberikan pengalaman pengguna yang baik dan mencapai kecepatan ideal. Meskipun konten yang dimuat terakhir biasanya lebih besar daripada elemen lain di halaman, kenyataannya tidak selalu begitu. Jadi, menerapkan preloading untuk konten terbesar akan membantu halaman mencapai nilai *LCP* yang lebih baik dan meningkatkan pengalaman halaman.



Gambar II. 5 Perbandingan *LCP* eligible dan not eligible (Faradilla, 2024)

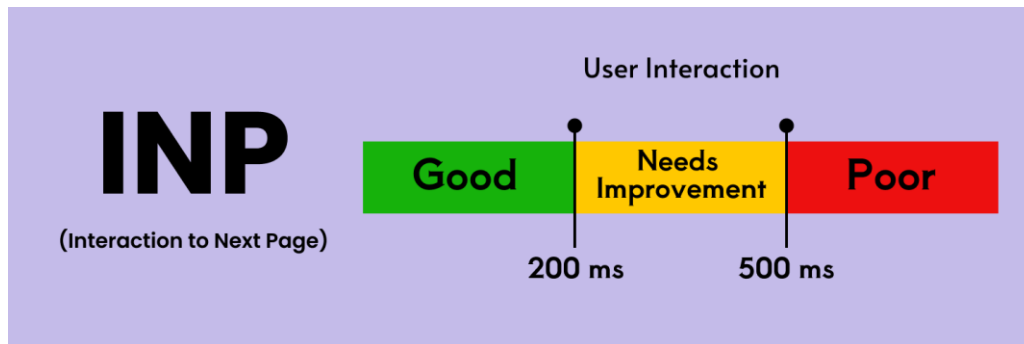
Faktor umum yang menyebabkan nilai *Largest Contentful Paint* menjadi buruk adalah waktu respons server yang lambat, rendering sisi klien (client-side), dan gambar berukuran besar.

#### II.4.2 Interaction to Next Paint (INP) – Kemampuan Respons Halaman

Interaction to Next Paint (INP) menilai seberapa cepat halaman Anda bereaksi terhadap interaksi pengguna, misalnya mengklik link atau tombol CTA tertentu. Skor INP yang tinggi berarti pengguna harus menunggu beberapa saat hingga halaman diperbarui setelah interaksi mereka. Hal ini bisa merusak pengalaman pengguna dan meningkatkan bounce rate website Anda. Skor INP dinilai seperti berikut:



- a. Hasil bagus – di bawah 200 milidetik.
- b. Perlu perbaikan – antara 200 dan 500 milidetik.
- c. Buruk – lebih dari 500 milidetik.



Gambar II. 6 Skor Interaction to Next Paint (INP)

Untuk memperbaiki skor INP, evaluasi ukuran dan kompleksitas website Anda. Pastikan untuk melakukan minify file JavaScript, mengaktifkan kompresi GZip, dan menggunakan CDN.

#### II.4.3 *Cumulative Layout Shift* – Pengalaman Visual Halaman

*Cumulative Layout Shift (CLS)* menilai stabilitas visual dan mengecek apakah ada pergeseran layout (tata letak) yang tidak terduga di halaman. Gerakan tak terduga terjadi ketika elemen halaman seperti teks, tombol, dan gambar terdorong ke bawah saat halaman masih loading. Ini bisa cukup mengesalkan bagi pengguna, karena terkadang mereka menjadi tidak sengaja mengklik elemen lain. Nilai *Cumulative Layout Shift* yang dijadikan patokan adalah 0,1 atau kurang.

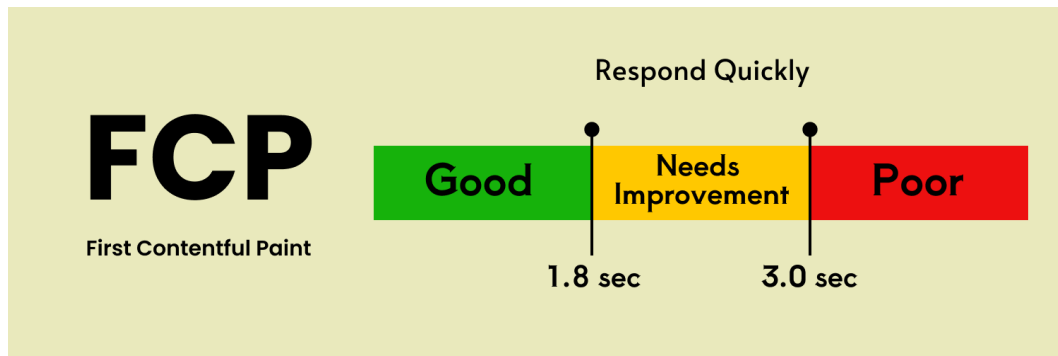
Sebelumnya, Google menilai metrik *Core Web Vitals* ini dengan menghitung jumlah nilai pergeseran layout individual di seluruh halaman. Sistem ini telah menilai jendela sesi sejak tahun 2021. ‘Jendela sesi’ adalah istilah yang digunakan untuk menyebut satu rangkaian pergeseran layout individual yang terjadi selama maksimal 5 detik, dengan jarak satu detik antarjendela. Faktor paling umum yang menyebabkan terjadinya pergeseran layout adalah adanya gambar, widget, dan banner iklan tanpa dimensi.

*Core Web Vitals* adalah bagian dari sinyal pengalaman halaman yang memberitahukan sejauh mana website bisa diakses secara mobile-friendly, aman untuk browsing, dan tanpa interstitial yang mengganggu. Sejak Agustus 2021, metrik pengalaman halaman telah menjadi pertimbangan ranking yang penting untuk perangkat seluler. Google bahkan menyelesaikan sistem peringkat ini untuk desktop pada Maret 2022. Pembaruan pengalaman halaman Google tersebut juga menunjukkan bahwa mereka masih memprioritaskan halaman yang kontennya paling relevan. Namun, pengalaman halaman hanya bisa meningkatkan visibilitas kalau beberapa halaman memiliki tingkat relevansi yang sama. Jadi, bisa kita simpulkan bahwa *Core Web Vitals* memainkan peran penting dalam faktor peringkat search engine optimization (SEO). *Core Web Vitals* juga membantu Anda memastikan bahwa website memberikan performa yang cepat dan stabil, sehingga bisa meningkatkan tingkat konversi dan engagement. Bahkan, lebih dari 50% pengguna perangkat mobile akan langsung menutup halaman kalau loadingnya memakan waktu lebih dari 3 detik. Apabila lebih lama 1 sampai 3 detik saja, ini bisa meningkatkan *bounce rate* hingga 32%. (Faradilla, 2024)

#### ***II.4.4 First Contentful Paint (FCP)***

Selain ketiga metrik utama dari *Core Web Vitals* yang telah dijelaskan sebelumnya, terdapat juga satu metrik lain yang digunakan dalam penelitian ini sebagai acuan dari hasil yang didapatkan ketika menggunakan *framework NextJS*. Metrik pengukuran tersebut ialah *First Contentful Paint (FCP)* yang merupakan metrik performa digunakan untuk mengukur waktu yang dibutuhkan halaman web untuk menampilkan konten pertamanya di layar pengguna. Ini menunjukkan seberapa cepat *browser* dapat *rendering* elemen konten pertama (misalnya, teks, gambar, atau kanvas) setelah pengguna membuka halaman tersebut.

*FCP* yang lebih cepat umumnya menghasilkan pengalaman pengguna yang lebih baik, karena membantu pengguna merasakan halaman dimuat dengan cepat. Untuk memberikan pengalaman pengguna yang baik, Google menyarankan situs harus berupaya memiliki *FCP* 1,8 detik atau kurang. (Crystallize, 2024)



Gambar II. 7 Skor First Contentful Paint (FCP)

## II.5 *Black-box Testing*

Pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program untuk mengetahui apakah fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Metode *Blackbox Testing* merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari data yang di harapkan, estimasi banyaknya data uji dapat dihitung melalui banyaknya *field* data entri yang akan diuji, aturan entri yang harus dipenuhi serta kasus batas atas dan batas bawah yang memenuhi. Dan dengan metode ini dapat diketahui jika fungsionalitas masih dapat menerima masukan data yang tidak diharapkan maka menyebabkan data yang disimpan kurang valid.

Solusi praktis peningkatan akurasi perlu dilakukan segera guna memperbaiki celah error yang telah ditemukan, selanjutnya dilakukan pengujian keamanan secara intensif melalui jaringan internal (*whitebox penetration testing*) secara berkala oleh *System Administrator* atau Pengelola Sistem Informasi, khususnya bagi yang mengelola perangkat lunak tersebut dan Untuk mencapai tingkat akurasi, dimana semua parameter akurasi yang terkait aspek kerahasiaan, integritas data, dan avalibilitas data dapat terpenuhi, maka harus dipertirnbangkan metode lain yang dapat dijadikan tolak ukur standar keamanan informasi.

Pengujian perangkat lunak merupakan elemen kritis dalam menentukan kualitas suatu perangkat lunak. Pengujian fungsional dilakukan untuk menemukan apakah ada kesalahan-kesalahan pada aplikasi yang telah dibangun. Tahap

pengujian adalah salah satu tahapan pada siklus pengembangan sebuah perangkat lunak. Tahap ini merupakan rangkaian kegiatan yang dilakukan untuk menemukan kesalahan atau kekurangan dari sebuah perangkat lunak atau aplikasi komputer. Aryandana bahkan menyebutkan bahwa *software Testing is helpful to alleviate the developers' task in improving the system*. Dapat disimpulkan bahwa kegiatan pengujian ini merupakan tahapan yang sangat penting dalam proses pengembangan perangkat lunak. Metode *Black Box Testing* terdiri atas beberapa cara antara lain *Equivalence Partitioning, Boundary Value Analysis, Comparison Testing, Sample Testing, Robustness Testing*, dan lain-lain. Metode *BlackboxTesting* merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari data yang di harapkan,Estimasi banyaknya data uji dapat dihitung melalui banyaknya *field* data entri yang akan diuji, aturan entri yang harus dipenuhi serta kasus batas atas dan batas bawah yang memenuhi. *Black-box testing* adalah pengujian sistem atau aplikasi yang berfokus pada spesifikasi fungsional sistem atau aplikasi. Penguji sistem dapat mendefinisikan satu set kondisi *input* dan menguji spesifikasi fungsional sistem (Cholifah et al., 2018).

Teknik pengujian *black box* yang digunakan dalam penelitian ini adalah Teknik *Equivalence Partitioning*. Teknik *equivalence partitioning* ialah menguji kualitas aplikasi yang akan dilakukan dokumentasi pengujian perangkat lunak dengan ditemukannya kesalahan pada setiap form yang dibagi menjadi tiga model kesalahan, yaitu kesalahan pada fungsi, struktur data dan antarmuka. Pengujian *Equivalence Partitioning* merupakan pengujian berdasarkan *inputan* setiap menu yang terdapat pada sistem. Metode *Black Box* berbasis *Equivalensi partitions* menguji tingkat akurasi yang akan dilakukan dokumentasi pengujian perangkat lunak dengan ditemukannya kesalahan pada setiap *form* yang dibagi menjadi lima model kesalahan, diantaranya: kesalahan pada Fungsi, Struktur data, *Interface*, Inisialisasi, dan *Performance*. Dalam teknik *Equivalence Class Partitioning* (ECP) perlu menguji hanya satu syarat dari setiap subset atau kelas yang dipartisi. Teknik ini membantu kita mengurangi jumlah kasus uji. Ini pada dasarnya berfungsi membagi *domain input* program berdasarkan nilai *input* ke dalam kelas kesetaraan. *Equivalence Class Partitioning* berusaha untuk mendefinisikan kasus uji yang

mengungkap kelas kesalahan dan dengan demikian mengurangi jumlah kasus uji yang dibutuhkan. Hal ini didasarkan pada evaluasi kelas-kelas ekuivalensi untuk suatu kondisi *input* (Y. D. Wijaya & Astuti, 2021).

## II.6 *System Usability Scale*

*System Usability Scale (SUS)* merupakan kuesioner yang dapat digunakan untuk mengukur usability sistem komputer menurut sudut pandang subyektif pengguna. *SUS* dikembangkan oleh John Brooke sejak 1986. Hingga saat ini, *SUS* banyak digunakan untuk mengukur usability dan menunjukkan beberapa keunggulan, antara lain: *SUS* dapat digunakan dengan mudah, karena hasilnya berupa skor 0–100. *SUS* sangat mudah digunakan, tidak membutuhkan perhitungan yang rumit. *SUS* tersedia secara gratis, tidak membutuhkan biaya tambahan, dan *SUS* terbukti valid dan *reliable*, walau dengan ukuran sampel yang kecil (Kurniawan et al., 2022).

Objek Pertanyaan nomor ganjil merupakan pertanyaan yang bernada positif dan pertanyaan nomor genap merupakan pertanyaan bernada negatif. Daftar pertanyaan dari *System Usability Scale (SUS)* yang akan digunakan dan disebarkan ke responden mengacu pada instrumen dari Brooke, yang ditunjukkan pada Tabel II.1 di bawah ini:

Tabel II. 1 Pertanyaan Kuesioner *SUS* (Saputra, 2019)

No.	Pertanyaan	Skor
1.	Saya pikir saya ingin menggunakan aplikasi ini	1 – 5
2.	Saya menemukan bahwa aplikasi ini tidak di buat serumit ini	1 – 5
3.	Saya pikir aplikasi ini mudah untuk digunakan	1 – 5
4.	Saya pikir saya perlu bantuan orang teknis dalam menggunakan sistem ini	1 – 5
5.	Saya menemukan berbagai fungsi diaplikasi ini terintegrasi dengan baik	1 – 5
6.	Saya pikir terlalu banyak ketidak konsistenan dalam sistem ini	1 – 5
7.	Saya akan membayangkan bahwa kebanyakan orang akan belajar dengan mudah dalam mempelajari aplikasi ini	1 – 5
8.	Saya menemukan aplikasi ini sangat tidak praktis	1 – 5
9.	Saya merasa sangat percaya diri dalam menggunakan aplikasi ini	1 – 5
10.	Saya perlu banyak belajar sebelum menggunakan aplikasi ini	1 – 5

Dari instrument pertanyaan pada Tabel II.1 di atas, di mana responden diberikan pilihan skala 1–5 untuk dijawab berdasarkan pada seberapa banyak responden setuju dengan setiap pernyataan tersebut terhadap aplikasi atau fitur yang di uji. Nilai 1 berarti sangat tidak setuju dan nilai 5 berarti sangat setuju dengan pernyataan tersebut. *System Usability Scale* mempunyai 5 jawaban yaitu sangat setuju, setuju, ragu-ragu, tidak setuju dan sangat tidak setuju. Skor dari pilihan jawaban tersebut dapat dilihat pada Tabel II.2 berikut ini.

Tabel II. 2 Skor *SUS* (Saputra, 2019)

Jawaban	Skor
Sangat Tidak Setuju (STS)	1
Tidak Setuju (TS)	2
Ragu-Ragu (RG)	3
Setuju (S)	4
Sangat Setuju (SS)	5

Setelah data-data kuesioner yang diberikan kepada responden terkumpul, selanjutnya akan melakukan konversi tanggapan responden dengan cara:

- a. Pernyataan ganjil, yaitu: 1, 3, 5, 7, dan 9 skor yang diberikan oleh responden dikurangi dengan 1.

$$\text{skor } SUS \text{ ganjil} = \Sigma Px - 1 \quad (1)$$

Dimana  $Px$  adalah jumlah pertanyaan ganjil.

- b. Pernyataan genap, yaitu 2, 4, 6, 8, dan 10 skor yang diberikan oleh responden digunakan untuk mengurangi 5.

$$\text{skor } SUS \text{ genap} = \Sigma 5 - Pn \quad (2)$$

Dimana  $Pn$  adalah jumlah pertanyaan genap.

- c. Hasil dari konversi tersebut selanjutnya dijumlahkan untuk setiap responden kemudian dikalikan dengan 2,5 agar mendapatkan rentang nilai antara 0 – 100.

$$(\Sigma \text{skor ganjil} - \Sigma \text{skor genap}) \times 2,5 \quad (3)$$

- d. Setelah skor dari masing-masing responden telah diketahui langkah selanjutnya adalah mencari skor rata-rata dengan cara menjumlahkan semua hasil skor dan dibagi dengan jumlah responden yang ada. Perhitungan ini dapat dilihat dengan rumus sebagai berikut:

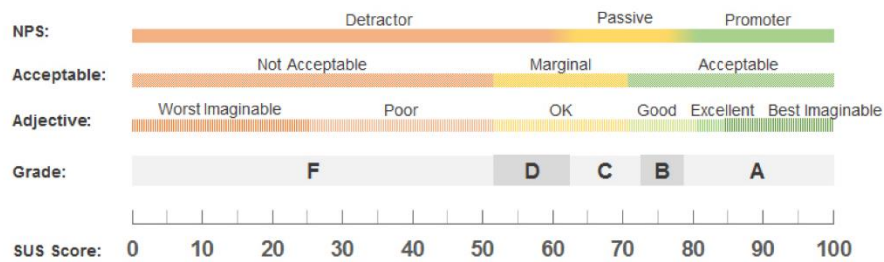
$$\bar{X} = \frac{\sum x}{n} \quad (4)$$

Di mana  $\bar{X}$  adalah skor rata-rata,  $\sum x$  adalah jumlah skor *System Usability Scale* dan  $n$  adalah jumlah dari responden.

Dari hasil tersebut akan diperoleh suatu nilai rata-rata dari seluruh penilaian skor responden. Untuk menentukan *Grade* hasil penilaian ada 2 (dua) cara yang dapat digunakan. Penentuan pertama dilihat dari sisi tingkat penerimaan pengguna, *Grade* skala dan adjektif rating yang terdiri dari tingkat penerimaan pengguna terdapat tiga kategori yaitu not *Acceptable*, marginal dan *Acceptable*. Sedangkan dari sisi tingkat *Grade* skala terdapat enam skala yaitu A, B, C, D, E dan F. Dan dari adjektif rating terdiri dari worst imaginable, poor, ok, good, excellent dan best imaginable. Penentuan yang kedua dilihat dari sisi percentile range (*SUS* skor) yang memiliki *Grade* penilaian yang terdiri dari A, B, C, D dan E. Penentuan hasil penilaian berdasarkan *SUS* score percentile rank dilakukan secara umum berdasarkan hasil perhitungan penilaian pengguna. Kedua penentuan ini dapat dilihat pada Tabel II.3 dan Gambar II.6 berikut ini.

Tabel II. 3 Range Skor *SUS*

<i>Grade</i>	<b>Keterangan</b>
A	skor $\geq 80,3$
B	skor $\geq 74$ dan $< 80,3$
C	skor $\geq 68$ dan $< 74$
D	skor $\geq 51$ dan $< 68$
E	skor lebih $< 51$



Gambar II. 8 Range Skor *SUS* (Rafid Pratama et al., 2024)

## II.7 Penelitian Relevan

Tabel II. 4 State of the Art

No.	Judul dan Nama Peneliti	Hasil Penelitian	Perbedaan Yang Didapatkan
1.	Pemanfaatan Sistem Informasi dalam Pemesanan serta Digitalisasi Tiket Bus Berbasis <i>Website</i>  Ismail Hanif Batubara (2022) (Batubara et al., 2022)	Jurnal ini membahas pengembangan sistem pemesanan tiket bus berbasis <i>website</i> menggunakan teknologi web seperti PHP, MySQL, Bootstrap, dan CodeIgniter. Penelitian ini mencakup analisis sistem, pembangunan sistem, penulisan kode program, dan pengujian sistem. Hasilnya adalah sebuah sistem pemesanan tiket bus yang memudahkan pengguna dalam memesan tiket dan mendapatkan informasi lain yang dibutuhkan. Artikel juga mencakup gambaran halaman-halaman utama <i>website</i> dan memberikan kesimpulan serta saran terkait pemesanan tiket bus melalui <i>website</i> .	Penelitian sebelumnya hanya menggunakan <i>framework</i> HTML dan CSS, sedangkan penulis menambahkan <i>framework</i> NextJS. Perbedaan kedua pada <i>UI frontend</i> , penulis menambahkan tampilan map sederhana dan menambahkan detail deskripsi tujuan akhir penumpang.
2.	Perancangan Aplikasi Sistem Informasi Pemesanan Tiket Bus <i>Online (E-Ticketing)</i> Pada Po. Handoyo	Hasil dari penelitian ini adalah untuk merancang dan mengembangkan aplikasi sistem informasi pemesanan tiket bus <i>online (E-ticketing)</i> untuk perusahaan PO. Handoyo. Penelitian ini menggunakan	Metode penulis menggunakan metode perancangan <i>SDLC Waterfall</i> , dan penggunaan <i>Database MongoDB</i> . Perbedaan selanjutnya, yaitu pada



No.	Judul dan Nama Peneliti	Hasil Penelitian	Perbedaan Yang Didapatkan
	Primagita Fridhayanti, dkk (2022) (Fridhayanti et al., 2022)	model ADDIE ( <i>Analysis, Design, Development, Implementation, Evaluation</i> ) dan bertujuan untuk meningkatkan kinerja perusahaan dengan menerapkan sistem pemesanan tiket <i>online</i> . Sistem informasi akuntansi, laporan keuangan, serta penggunaan <i>MySQL</i> untuk manajemen basis data.	metode pembayaran. Penulis hanya menggunakan metode pembayaran secara manual untuk menghindari adanya kolaborasi dari pihak ketiga.
3.	Impact of <i>E-Ticketing</i> Application on Bus Transportation in Bandung  Eddy Soeryanto Soegoto (2019) (Soegoto et al., 2020)	Hasil dari penelitian ini adalah untuk menentukan dampak dari implementasi sistem <i>e-ticketing</i> pada transportasi bus bagi masyarakat Bandung. Hasil penelitian menunjukkan bahwa implementasi sistem <i>e-ticketing</i> pada transportasi umum memberikan kemudahan dan kenyamanan dalam layanan pemesanan tiket, yang dapat mendorong masyarakat untuk menggunakan transportasi umum	Penelitian sebelumnya lebih membahas pada dampak yang ditimbulkan oleh pembelian tiket secara <i>online</i> , sedangkan penulis lebih berfokus pada perancangan sistem <i>website</i> .
4.	Perancangan Aplikasi <i>E-Ticketing</i> Pada Agen Bus Berbasis <i>Website</i> Menggunakan Laravel  Muhammad Haidar Wijaya1, Magdalena A. Ineke Pakereng (2021) (M. H. Wijaya & Pakereng, 2021)	Pengembangan sebuah sistem <i>e-ticketing</i> berbasis <i>website</i> menggunakan <i>framework</i> Laravel untuk memungkinkan pelanggan melakukan transaksi pembelian tiket bus secara <i>online</i> . Pengujian sistem dilakukan dengan metode <i>blackbox</i> dan semua form dalam aplikasi diuji tanpa error.	Perbedaan pada <i>framework</i> , penelitian terdahulu menggunakan laravel sedangkan penulis menggunakan <i>NextJS</i> .
5.	A Preliminary Study: User Interface Design of <i>Online</i> Travel Booking Application  Nur Ani (2019) (Ani, 2019)	Menyajikan desain antarmuka pengguna dari aplikasi pemesanan perjalanan <i>online</i> yang sesuai dengan karakteristik masyarakat Indonesia. Penelitian ini mencakup evaluasi aplikasi pemesanan perjalanan <i>online</i> yang sudah ada dan mengusulkan model awal desain antarmuka pengguna yang mencakup menu	Penelitian ini tidak membahas tentang <i>backend website</i> . Fokus utama dari penelitian ini adalah pada desain antarmuka pengguna aplikasi pemesanan perjalanan <i>online</i> yang sesuai

No.	Judul dan Nama Peneliti	Hasil Penelitian	Perbedaan Yang Didapatkan
		utama, penggeser konten, formulir pemesanan <i>online</i> , badan, dan footer.	dengan karakteristik masyarakat Indonesia.
6.	Bus Ticket Booking Information System  Setya Chendra Wibawa, dkk (2021) (Wibawa et al., 2021)	Pengembangan sistem informasi pemesanan tiket bus untuk mengatasi keterbatasan mobilitas dan dampak pandemi Covid-19 terhadap transportasi. Mengurangi kebutuhan untuk mengunjungi outlet penjualan tiket dan meminimalkan kerumunan.	<i>Framework</i> yang digunakan hanya adalah HTML, sedangkan penelitian saat ini menggunakan tambahan <i>framework NextJS</i> .
7.	Bus Ticketing System with Multiple Routes Suggestion  Foo Yoke Ling, dkk (2021) (Et.al, 2021)	Pengembangan sistem pemesanan tiket bus dengan saran rute ganda menggunakan algoritma generasi rute canggih. Sistem ini bertujuan untuk memberikan rute yang direkomendasikan jika tujuan yang diinginkan tidak memiliki bus langsung, dan mencakup fitur-fitur seperti pemilihan rute, implementasi <i>SQL</i> , dan pemesanan tiket bus.	Penelitian terdahulu menggunakan basis data <i>SQL</i> , sedangkan penelitian saat ini menggunakan <i>NoSQL</i> .
8.	Implementation process the design model business of <i>e-ticket</i> transportation bus in Medan  Desilia Selvida, Muhammad Zarlis (2018) (Selvida & Zarlis, 2018)	Mengimplementasikan model bisnis <i>e-ticketing</i> untuk transportasi bus di Medan dengan tujuan untuk meningkatkan layanan transportasi di kota tersebut. Penelitian ini juga menganalisis proses pembelian tiket saat ini, metodologi penelitian yang digunakan, implementasi sistem, dan model bisnis perusahaan.	Hanya membahas <i>FrontEnd website</i> , sedangkan penelitian saat ini membahas <i>FrontEnd</i> dan <i>BackEnd</i> sistem.
9.	Rancang Bangun Aplikasi Pemesanan Tiket Bus <i>Online</i> (BeTik Bus) Berbasis <i>Website</i>  Gregorius Aryo P, dkk (2023) (Prasojo & Pana Kontesta, 2023)	Pengembangan aplikasi pemesanan tiket bus berbasis <i>website</i> menggunakan metode waterfall dan <i>framework PHP</i> . Pemesanan tiket bus berbasis <i>online</i> ini dapat memberikan pelayanan yang mudah kepada publik, khususnya di kabupaten bengkayang agar dapat melakukan booking tiket bus lebih cepat dan mudah.	Penambahan <i>framework NextJS</i> dan metode pengujian <i>SUS</i> .

No.	Judul dan Nama Peneliti	Hasil Penelitian	Perbedaan Yang Didapatkan
10.	Sistem Informasi Penjadwalan Supir Bus Po Logos Berbasis <i>Website</i>  Zakharia, dkk (2021) (Zakharia et al., 2021)	Pengembangan sistem informasi penjadwalan supir bus menggunakan algoritma genetika untuk mengatasi masalah penjadwalan yang tidak merata pada perusahaan PO Logos. Mereka menggunakan metode pengujian <i>Black Box Testing</i> untuk mendeteksi error dalam sistem	<i>Framework</i> yang digunakan adalah PHP, sedangkan penelitian saat ini menggunakan <i>NextJS</i> . Metode pengujian juga hanya menggunakan <i>blackbox</i> , sedangkan penelitian saat ini terdapat pengujian <i>SUS</i> .
11.	Metode <i>SDLC</i> Waterfall Pada Rancang Bangun Sistem Informasi Sekolah SMP Negeri 10 Kaur  Kirman (2022) (Kirman & Saputra, 2022)	Penelitian yang telah dijalankan dan tujuan yang ingin dicapai pada penelitian ini yaitu pengembangan prototipe sistem informasi ini berhasil, dengan menggunakan pengembangan metode <i>SDLC</i> Waterfall, yang dibuktikan dengan hasil pengujian <i>black box</i> , yaitu keseluruhan dari sistem yang dikembangkan tidak memiliki bug/error pada sistem.	<i>Website</i> yang dirancang membahas tentang sistem informasi sekolah, tidak membahas tentang reservasi tiket <i>online</i> . Metode pengujian hanya <i>blackbox</i> , tidak terdapat pengujian <i>SUS</i> .
12	Pemanfaatan <i>NextJS</i> Dan <i>MongoDB</i> Dalam Sistem Informasi Web Manajemen Data Beras Pada Ud Sri Utami  Munif Sanjaya, Pelsri Ramadar Noor Saputra (2023) (Sanjaya et al., 2023)	Merancang sebuah <i>website database</i> beras dengan <i>framework NextJS</i> dan databse <i>MongoDB</i> , serta pengujian menggunakan metode <i>Blackbox</i> .	Perbedaan pada penelitian saat ini ialah <i>website</i> membahas persoalan reservasi tiket <i>online</i> serta penambahan metode pengujian <i>SUS</i> .