

DAFTAR PUSTAKA

- Al-Faruq, M. N. M., Nur'aini, S., & Aufan, M. H. (2022). Perancangan Ui/Ux Semarang Virtual Tourism Dengan Figma. *Walisongo Journal of Information Technology*, 4(1), 43–52. <https://doi.org/10.21580/wjit.2022.4.1.12079>
- Andrianof, H. (2018). Rancang Bangun Sistem Informasi Promosi dan Penjualan pada Toko Ruminansia Berbasis WEB. *Jurnal Pti (Pendidikan Dan Teknologi Informasi) Fakultas Keguruan Ilmu Pendidikan Universita Putra Indonesia "Yptk" Padang*, 5(1), 11–19. <https://doi.org/10.35134/jpti.v5i1.2>
- Anharudin, A., & Nasser, H. A. (2020). Rancang Bangun Aplikasi Reservasi Kamar Hotel Berbasis Web. *PROSISKO: Jurnal Pengembangan Riset Dan Observasi Sistem Komputer*, 7(1). <https://doi.org/10.30656/prosisko.v7i1.2131>
- Ani, N. (2019). A Preliminary Study : User Interface Design of Online Travel Booking Application. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, December 2019, 261–267. <https://doi.org/10.32628/cseit195645>
- Batubara, I. H., Raihan, E. A., Tanjung, M. I., Fadlurohman, D., & Can, A. (2022). Pemanfaatan Sistem Informasi dalam Pemesanan serta Digitalisasi Tiket Bus Berbasis Website. *Blend Sains Jurnal Teknik*, 1(1), 55–61. <https://doi.org/10.56211/blendsains.v1i1.73>
- Bui, D. (2023). Next.js for *Front-end* and Compatible Backend Solutions. *XAMK*.
- Cholifah, W. N., Yulianingsih, Y., & Sagita, S. M. (2018). Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 3(2), 206. <https://doi.org/10.30998/string.v3i2.3048>
- Crystallize. (2024). *First Contentful Paint (FCP)*. <https://crystallize.com/learn/best-practices/frontend-performance/other-web-vitals/first-contentful-paint-fcp>

- Divisi Humas POLRI. (2023). *Tahukah kamu moda transportasi umum apa yang terbanyak digunakan pemudik?* <https://www.instagram.com/p/Cq5gwioJ4Fa/>
- Et.al, F. Y. L. (2021). Bus Ticketing System with Multiple Routes Suggestion. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(3), 1816–1824. <https://doi.org/10.17762/turcomat.v12i3.1010>
- Faradilla. (2024). *Core Web Vitals: Definisi, Fungsi, dan Penggunaannya*. Www.Hostinger.Co.Id. <https://www.hostinger.co.id/tutorial/core-web-vitals>
- Figma. (n.d.). <https://www.figma.com/>
- Fridhayanti, P., Djajanto, L., & Haris, Z. A. (2022). Perancangan Aplikasi Sistem Informasi Pemesanan Tiket Bus Online (E-Ticketing) Pada PO. Handoyo. *Indonesian Journal Of Economy, Business, Entrepreneurship and Finance*, 2(2), 225–242.
- Jaya, H. (2017). PERANCANGAN HYPERMEDIA BERBASIS WEB PADA MATA KULIAH ELEKTRONIKA DIGITAL JURUSAN PTA-FT UNM. *Jurnal Elektronika Telekomunikasi & Computer*, 12(2).
- Karmila. (2019). *PENGEMBANGAN SISTEM INFORMASI PELAYANAN DINAS SOSIAL PADA PROVINSI SULAWESI BARAT BERBASIS WEB*. Universitas Negeri Makassar.
- Kirman, K., & Saputra, E. E. (2022). Metode SDLC Waterfall Pada Rancang Bangun Sistem Informasi Sekolah SMP Negeri 10 Kaur. *JUSIBI (Jurnal Sistem Informasi Dan E-Bisnis)*, 4(2), 112–118. <https://doi.org/10.54650/jusibi.v4i2.453>
- Kurniawan, E., Nofriadi, N., & Nata, A. (2022). Penerapan System Usability Scale (Sus) Dalam Pengukuran Kebergunaan Website Program Studi Di Stmik Royal. *Journal of Science and Social Research*, 5(1), 43. <https://doi.org/10.54314/jssr.v5i1.817>
- Nopriandi, H. (2018). Perancangan Sistem Informasi Registrasi Mahasiswa. *Jurnal*

- Teknologi Dan Open Source*, 1(1), 73–79. <https://doi.org/10.36378/jtos.v1i1.1>
- PageSpeed Insights*. (n.d.). https://pagespeed.web.dev/analysis/https-talenthub-kemnaker-go-id/y842xyvvnb?form_factor=mobile
- Prasojo, G. A., & Pana Kontesta, R. (2023). Rancang Bangun Aplikasi Pemesanan Tiket Bus Online (BeTik Bus) Berbasis Website. *JIFOTECH (JOURNAL OF INFORMATION TECHNOLOGY)*, 3(1).
- Rafid Pratama, M., Umam, J., & Yakok, R. (2024). Usability Testing pada Aplikasi iJateng Menggunakan Metode System Usability Scale. *Jurnal Sistem Informasi, Manajemen Dan Teknologi Informasi*, 2(1), 15–23. <https://doi.org/10.33020/jsimtek.v2i1.556>
- Sanjaya, M., Ramadar, P., & Saputra, N. (2023). PEMANFAATAN NEXTJS DAN MONGODB DALAM SISTEM INFORMASI WEB MANAJEMEN DATA BERAS PADA UD SRI UTAMI. *INFORMATION SYSTEM FOR EDUCATORS AND PROFESSIONALS*, 8(1), 25–36.
- Saputra, A. (2019). Penerapan Usability pada Aplikasi PENTAS Dengan Menggunakan Metode System Usability Scale (SUS) (Usability Implementation in PENTAS Application Using the System Usability Scale (SUS) Method). *Jurnl Teknologi Informasi Dan Multimedia (JTIM)*, 1(3), 206–212.
- Sari, J., & Ferdian, A. (2023, February 13). *Total Jumlah Bus di Indonesia Tembus 213.830 Unit*. [Www.Kompas.Com. https://otomotif.kompas.com/read/2023/02/13/162100715/total-jumlah-bus-di-indonesia-tembus-213.830-unit](https://otomotif.kompas.com/read/2023/02/13/162100715/total-jumlah-bus-di-indonesia-tembus-213.830-unit)
- Selvinda, D., & Zarlis, M. (2018). Implementation process the design model business of e-ticket transportation bus in Medan. *IOP Conference Series: Materials Science and Engineering*, 420(1). <https://doi.org/10.1088/1757-899X/420/1/012134>
- Shaleh, I. A., Yogi, J. P., Pirdaus, P., Syawal, R., & Saifudin, A. (2021). Pengujian

- Black Box pada Sistem Informasi Penjualan Buku Berbasis Web dengan Teknik Equivalent Partitions. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 4(1), 38. <https://doi.org/10.32493/jtsi.v4i1.8960>
- Silalahi, M. (2018). Perbandingan Performansi Database MongoDB Dan Mysql Dalam Aplikasi File Multimedia Berbasis Web. *Computer Based Information System Journal*, 6(1), 63. <https://doi.org/10.33884/cbis.v6i1.574>
- Soegoto, E. S., Setiawan, R., & Jumansyah, R. (2020). Impact of E-Ticketing Application on Bus Transportation in Bandung. *Proceedings of the International Conference on Business, Economic, Social Science, and Humanities – Economics, Business and Management Track (ICOBEST-EBM 2019)*, 112, 25–28. <https://doi.org/10.2991/aebmr.k.200108.008>
- Sutanto, P. H., Lidwan, N., & Ridwan, W. (2020). Perancangan Reservasi Pariwisata Dan Kamar Hotel Berbasis Web. *JISAMAR (Journal of Information System, Applied, Management, Accounting and Research)*, 4(4), 39–48. <http://journal.stmikjayakarta.ac.id/index.php/jisamarTelp.+62-21-3905050>
- Wibawa, S. C., Fatah, D. A., Huda Firmansyah, M. M., Mustafit, N. H., A., D. L., Putri, E. R., & Laksono, C. S. (2021). Bus Ticket Booking Information System. *International Journal of Science, Engineering and Information Technology*, 6(1), 266–269. <https://doi.org/10.21107/ijseit.v6i1.13155>
- Wijaya, M. H., & Pakereng, M. A. I. (2021). Perancangan Aplikasi E-Ticketing Pada Agen Bus Berbasis Website Menggunakan Laravel. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 8(3), 1384–1396. <https://doi.org/10.35957/jatisi.v8i3.974>
- Wijaya, Y. D., & Astuti, M. W. (2021). Pengujian Blackbox Sistem Informasi Penilaian Kinerja Karyawan Pt Inka (Persero) Berbasis Equivalence Partitions. *Jurnal Digital Teknologi Informasi*, 4(1), 22. <https://doi.org/10.32502/digital.v4i1.3163>
- Zakharia, Z., Widiatry, W., & Sari, N. N. K. (2021). Sistem Informasi Penjadwalan

Supir Bus Po Logos Berbasis Website. *Journal of Information Technology and Computer Science*, 1(1), 29–38.
<https://doi.org/10.47111/jointecom.v1i1.2512>

LAMPIRAN**Lampiran 1. Surat Pengambilan Data****Lampiran 1. a Surat Usulan Pengambilan Data Penelitian**

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

DEPARTEMEN TEKNIK ELEKTRO

Jl. Poros Malino KM.6 Gowa, 92171, Sulawesi Selatan
<http://eng.unhas.ac.id/elektro>, email: elektro@unhas.ac.id

Nomor : 29136/UN4.7.7.1/PT.01.06/2023
Lamp : -
Hal : Usulan Pengambilan Data Penelitian

Kepada Yth.
Wakil Dekan Bidang Akademik dan Kemahasiswaan

Di –
Fakultas Teknik,
Jl. Malino No.8 F, Romang Lompoo, Kec. Bontomarannu, Kabupaten Gowa, Sulawesi Selatan
92171

Dengan Hormat,
Kiranya Mahasiswa tersebut dibawah ini
Nama/Stambuk : Bryan Valentino/ D041201115
Judul TA : Perancangan Sistem Informasi Bus Online Ticketing berbasis Website.
Dapat diberikan surat pengantar untuk pengambilan Data/ Penelitian Skripsi/ tugas akhir pada:
1. PT. Borlindo Mandiri Jaya.

Adapun data/penelitian yang diperlukan :
Pengambilan data berupa Deskripsi Perusahaan, Bus, serta Pengemudi, Visi-misi Perusahaan, Deskripsi Rute, agen, dan Nomor Telepon tiap agen, serta diskusi singkat bersama pihak perusahaan.

Demikian penyampaian kami, atas perhatian dan kerjasamanya diucapkan terima kasih

Gowa, 13 Desember 2023
Sekretaris Departemen Teknik Elektro



Dr. Ir. Ikhlas Kitta, S.T., M.T.
NIP. 197609142008011006

Lampiran 1. b Surat Pengambilan Data Penelitian pada PT. Borlindo Mandiri Jaya



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

Jl. Poros Malino KM.6 Gowa, 92171, Sulawesi Selatan
Telp. (0411) 586015. 586262 Fax (0411) 586015
<http://eng.unhas.ac.id/elektro>, email: elektro@unhas.ac.id

Nomor : 29138/UN4.7.7.1/PT.01.06/2023 Gowa, 13 Desember 2023
Lamp : -
Hal : Pengambilan Data Penelitian Mahasiswa

Kepada Yth.
PT. Borlindo Mandiri Jaya
Di –
Jl. Jalur, Lingkar Barat, Tamalanrea, Kota Makassar, Sulawesi Selatan, 90245.

Dengan hormat, kami sampaikan bahwa dalam rangka penyelesaian skripsi/tugas akhir pada Departemen Teknik Elektro Fakultas Teknik Universitas Hasanuddin, maka kami mohon atas kebijaksanaan Bapak/Ibu kiranya berkenan memberikan kesempatan melakukan pengambilan data penelitian:

Nama/Stambuk : Bryan Valentino / D041201115

Judul TA : Perancangan Sistem Informasi Bus Online Ticketing berbasis Website.

Tujuan : Pengambilan data berupa Deskripsi Perusahaan, Bus, serta Pengemudi, Visi-misi Perusahaan, Deskripsi Rute, agen, dan Nomor Telepon tiap agen, serta diskusi singkat bersama pihak perusahaan.

Atas perhatian dan kerjasama yang baik kami sampaikan terima kasih

a.n. Dekan,
Wakil Dekan Bidang Akademik dan Kemahasiswaan



Dr. Amil Ahmad Ilham, S.T., M.IT.
NIP 19731010 199802 1 001

Tembusan: 1. Dekan FT-UH
2. Ketua Departemen Teknik Elektro FT-UH

Lampiran 2. Surat Permohonan Izin Publikasi Website



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN
TEKNOLOGI UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK ELEKTRO
Jl. Poros Malino KM.6 Gowa, 92171, Sulawesi Selatan
<http://eng.unhas.ac.id/elektro>, email: elektro@unhas.ac.id



LEMBAR PERMOHONAN IZIN PUBLIKASI WEBSITE SEMENTARA

Kepada Yth.

PT. Borlindo Mandiri Jaya

Di –

Jl. Jalur, Lingkar Barat, Tamalanrea, Kota Makassar, Sulawesi Selatan, 90245.

Dengan senang hati, kami sampaikan bahwa perancangan *website borlindo.my.id* telah berhasil kami selesaikan. Selanjutnya kami akan melakukan beberapa pengujian berupa pengujian *First Input Delay*, *Black-box Testing*, serta pengujian *System Usability Scale*. Tentunya pengujian ini kami lakukan apabila *website borlindo.my.id* kami telah dipublikasikan. Oleh karena itu, kami mohon atas kebijaksanaan Bapak/Ibu kiranya berkenan memberikan kami izin untuk mempublikasikan *website* kami tersebut. *Website* tersebut tidak akan kami komersialkan dan/atau mengambil keuntungan dikarenakan tidak terdapat proses transaksi jual/beli secara resmi dalam sistem, serta Nomor Rekening yang ada akan kami sensor untuk keamanan bersama. Dengan melakukan tanda tangan di bawah ini, maka Bapak/Ibu setuju dengan pernyataan yang ada.

Nama / NIM Peneliti : Bryan Valentino / D041201115

Judul Penelitian : “Sistem Informasi Bus Online Ticketing Berbasis *Web* Pada PT.
Borlindo Mandiri Jaya”

Domain Website : *borlindo.my.id*

Jangka Waktu : 1 – 2 bulan

Atas perhatian dan kerjasama yang baik dari Bapak/Ibu kami ucapkan banyak terima kasih.

Makassar, 11 / 06 / 2024
a.n. PT. Borlindo Mandiri Jaya

SADRAN TAFAN 5

Lampiran 3. Kode Pemrograman *Front-end* Website

Lampiran 3. a Kode Pemrograman *Front-end*, (package.JSON) – Inisiasi Library

```
{
  "name": "borlindo_fe",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint"
  },
  "dependencies": {
    "@headlessui/react": "^1.7.19",
    "@heroicons/react": "^2.1.1",
    "@react-pdf/renderer": "^3.4.4",
    "@tanstack/react-query": "^5.21.4",
    "@tanstack/react-query-devtools": "^5.21.5",
    "axios": "^1.6.8",
    "CLSx": "^2.1.1",
    "formik": "^2.4.6",
    "framer-motion": "^11.0.5",
    "js-cookie": "^3.0.5",
    "moment": "^2.30.1",
    "next": "14.1.0",
    "next-auth": "^4.24.6",
    "nookies": "^2.5.2",
    "react": "^18",
    "react-dom": "^18",
    "react-toastify": "^10.0.5"
  },
  "devDependencies": {
    "autoprefixer": "^10.0.1",
    "eslint": "^8",
    "eslint-config-next": "14.1.0",
    "postcss": "^8",
    "tailwindcss": "^3.3.0"
  }
}
```

Lampiran 3. b Kode Pemrograman *Front-end*, (tailwind.config.js) – Konfigurasi Template Styling Website. Source Code terlampir hanya bagian awal dan akhir saja

```

/** @type {import('tailwindcss').Config} */

module.exports = {
  content: [
    './src/pages/**/*.{js,ts,jsx,tsx,mdx}',
    './src/components/**/*.{js,ts,jsx,tsx,mdx}',
    './src/app/**/*.{js,ts,jsx,tsx,mdx}',
  ],
  theme: {
    container: {
      center: true,
      padding: '1rem',
      screens: {
        xs: '375px',
        // => @media (max-width: 375px) { ... }

        sm: '640px',
        // => @media (max-width: 640px) { ... }
      },
    },
    screens: {
      xs: '320px',
      // => @media (min-width: 375px) { ... }
      sm: '640px',
      // => @media (min-width: 640px) { ... }
      md: '768px',
      // => @media (min-width: 768px) { ... }
    },
    extend: {
      backgroundImage: {
        'bus-hero': "url('/Borlindo2.png')",
        crown: "url('/logo.png')",
        'acsen-1': "url('/acsen-1.svg')",
        'acsen-2': "url('/acsen-2.svg')",
        'acsen-3': "url('/acsen-3.svg')",
        'acsen-4': "url('/acsen-4.svg')",
      },
      colors: {
        Primary: {
          50: '#FBEAE9',
          100: '#F7D6D6',
          200: '#F0B0AF',
          300: '#E98989',
          400: '#E26362',
          500: '#DB3C3B',
          600: '#B72322',
        },
      },
    },
  },
}

```

```

      700: '#831918',
      800: '#50F0F',
      900: '#1C0505',
    },
    Secondary: {
      50: '#FFF9EA',
      100: '#FFF2D3',
      200: '#FFE4A5',
      300: '#FFD677',
      400: '#FFC849',
      500: '#FFBA1B',
      600: '#DD9A00',
      700: '#A06F00',
      800: '#624500',
      900: '#251A00',
    },
  },
  .
  .
  .
},
fontSize: {
  'display-1-mobile': ['36px', '40px'],
  'h1-mobile': ['28px', '36px'],
  'h2-mobile': ['22px', '28px'],
  'p1-mobile': ['16px', '24px'],
  'pr-mobile': ['14px', '20px'],
  'cap-1-mobile': ['10px', '12px'],
  'cap-2-mobile': ['12px', '14px'],

  // Desktop Font
  'display-1-desktop': ['60px', '68px'],
  'display-2-desktop': ['56px', '60px'],
  'h1-desktop': ['42px', '52px'],
  'h2-desktop': ['36px', '44px'],
  'h3-desktop': ['32px', '40px'],
},
keyframes: {
  shimmer: {
    '100%': {
      transform: 'translateX(100%)',
    },
  },
},
},
},
},
},
},
},
},
},
},
plugins: [],
};

```

Lampiran 3. c Kode Pemrograman *Front-end*, (next.config.mjs) – Konfigurasi Build. Source Code terlampir hanya bagian awal dan akhir saja. Autentikasi kami rahasiakan untuk kepentingan bersama.

```

/** @type {import('next').NextConfig} */
const nextConfig = {
  // async rewrites() {
  //   return [
  //     {
  //       source: "/api/:path*",
  //       destination: "/src/app/api/:path*",
  //     },
  //   ];
  // },
  // },
  eslint: {
    ignoreDuringBuilds: true,
  },
  env: {
    API_URL: "-",
    // API_URL: "-",

    // AUTH_SECRET: "-o=",
    NEXTAUTH_SECRET: "-",
    // NEXTAUTH_URL: "-",
  },
  images: {
    domains: [
      "images.unsplash.com",
      "i.ibb.co",
      "-",
    ],
  },
  .
  .
  .
};

export default nextConfig;

```

Lampiran 3. d Kode Pemrograman *Front-end*, (.env) – API Endpoints. Autentikasi kami rahasiakan untuk kepentingan bersama.

```

API_URL: "-"
# API_URL: "-"

NEXTAUTH_SECRET: "-",
# NEXTAUTH_URL: "-"

```

Lampiran 3. e Kode Pemrograman *Front-end*, src-api-auth-(signin.js) – Menyesuaikan data signin

```
"use client";

import { useMutation } from "@tanstack/react-query";
import axios from "axios";
import { setCookie } from "nookies";

const LoginHook = async (value) => {
  const params = new URLSearchParams();
  params.append("username", value.username);
  params.append("password", value.password);
  const request = await axios
    .post(process.env.API_URL + `/api/v1/auth/signin`, params)
    .then((responses) => {
      if (responses.status === 200) {
        setCookie(null, "accessToken", responses.data.accessToken, {
          maxAge: 43200,
        });
      }
    });
  return responses;
};

export function useLoginHooks() {
  return useMutation({
    mutationFn: LoginHook,
  });
}
```

Lampiran 3. f Kode Pemrograman *Front-end*, src-api-auth-(signup.js) – Menyesuaikan data signup

```
"use client";

import { useMutation } from "@tanstack/react-query";
import axios from "axios";

const SignUpAccount = async (value) => {
  const request = await axios
    .post(process.env.API_URL + `/api/v1/auth/signup`, value)
    .then((responses) => {
      return responses;
    });
};

export function useSignUpAccounts() {
  return useMutation({
    mutationFn: SignUpAccount,
  });
}
```

Lampiran 3. g Kode Pemrograman *Front-end*, src-api-auth-(signup.js) – Menyesuaikan data signup

```
'use client';
import { useQuery } from '@tanstack/react-query';
import axios from 'axios';
const GetListBus = (key) => {
  const asal = key.queryKey[1];
  const tujuan = key.queryKey[2];
  const departureTime = key.queryKey[3];
  const totalSeats = key.queryKey[4];
  const request = axios
    .get(
      process.env.API_URL +
        `/api/v1/bus?asal=${asal}&tujuan=${tujuan}&departureTime=${departureTime}&totalSeats=${totalSeats}`
    )
    .then((responses) => {
      return responses;
    });
export const useGetListBus = (asal, tujuan, departureTime, totalSeats) =>
  useQuery({
    queryKey: ['List Bus', asal, tujuan, departureTime, totalSeats],
    queryFn: GetListBus,
  });
```

Lampiran 3. h Kode Pemrograman *Front-end*, src-api-dashboard-(get-dashboard-statistic.js) - Memanggil data dashboard dari *Back-end*

```
"use client";
import { useQuery } from "@tanstack/react-query";
import axios from "axios";
const request = axios
  .get(process.env.API_URL + `/api/v1/dashboard`, {
    headers: {
      Authorization: `Bearer ${token}`,
    },
  })
  .then((responses) => {
    return responses;
  });
export const useGetDashboardStatistic = (token) =>
  useQuery({
    enabled: !!token,
    queryKey: ["Statistic Dashboard", token],
    queryFn: GetDashboardStatistic,
    refetchOnWindowFocus: false,
    refetchOnMount: false,
    refetchOnReconnect: false,
  });
```

Lampiran 3. i Kode Pemrograman *Front-end*, src-api-news-(get-news-statistic.js) - Memanggil data berita dari *Back-end*

```
'use client';
import { useQuery } from '@tanstack/react-query';
import axios from 'axios';
const GetAllNews = (key) => {
  const page = key.queryKey[1];
  const limit = key.queryKey[2];
  const request = axios
    .get(process.env.API_URL + `/api/v1/news?page=${page}&limit=${limit}`)
    .then((responses) => {
      return responses;
    })
};
export const useGetAllNews = (page, limit) =>
  useQuery({
    queryKey: ['List All News', page, limit],
    queryFn: GetAllNews,
    refetchOnWindowFocus: false,
    refetchOnMount: false,
    refetchOnReconnect: false,
  });
```

Lampiran 3. j Kode Pemrograman *Front-end*, src-api-route-(get-all-route.js) - Memanggil data rute dari *Back-end*

```
'use client';

import { useQuery } from '@tanstack/react-query';
import axios from 'axios';
const GetAllRoute = (key) => {
  const page = key.queryKey[1];
  const limit = key.queryKey[2];
  const request = axios
    .get(process.env.API_URL + `/api/v1/route?page=${page}&limit=${limit}`)
    .then((responses) => {
      return responses;
    })
};
return request;
};
export const useGetAllRoute = (page, limit) =>
  useQuery({
    queryKey: ['List All Route', page, limit],
    queryFn: GetAllRoute,
    refetchOnWindowFocus: false,
    refetchOnMount: false,
    refetchOnReconnect: false,
  });
```

Lampiran 3. k Kode Pemrograman *Front-end*, src-api-statistic-(get-all-statistic.js) - Memanggil data statistik dari *Back-end*

```
'use client';

import { useQuery } from '@tanstack/react-query';
import axios from 'axios';

const GetAllStatistic = () => {
  const request = axios
    .get(process.env.API_URL + `/api/v1/statistic`)
    .then((responses) => {
      return responses;
    })
    .catch((err) => {
      return err;
    });
  return request;
};

export const useGetAllStatistic = () =>
  useQuery({
    queryKey: ['List All Statistic'],
    queryFn: GetAllStatistic,
    refetchOnWindowFocus: false,
    refetchOnMount: false,
    refetchOnReconnect: false,
  });
```

Lampiran 3. l Kode Pemrograman *Front-end*, src-api-transaction-(get-list-transaction.js) - Memanggil data transaksi dari *Back-end*

```
'use client';

import { useQuery } from '@tanstack/react-query';
import axios from 'axios';

const GetListTransaction = (key) => {
  const token = key.queryKey[1];
  const page = key.queryKey[2];
  const limit = key.queryKey[3];
  const startdate = key.queryKey[4];
  const enddate = key.queryKey[5];
  const search = key.queryKey[6];
  const request = axios
    .get(
      process.env.API_URL +
        `/api/v1/transaction?page=${page}&limit=${limit}&startdate=${startdate}&enddate=${enddate}&search=${search}`
    ),
```



```

    {
      headers: {
        Authorization: `Bearer ${token}`,
      },
    }
  )
  .then((responses) => {
    return responses;
  })
  .catch((err) => {
    return err;
  });
  return request;
};

export const useGetListTransaction = (
  token,
  page,
  limit,
  startdate,
  enddate,
  search
) =>
  useQuery({
    enabled: !!token,
    queryKey: [
      'List All Transaction',
      token,
      page,
      limit,
      startdate,
      enddate,
      search,
    ],
    queryFn: GetListTransaction,
    refetchOnWindowFocus: false,
    refetchOnMount: false,
    refetchOnReconnect: false,
  });

```

Lampiran 3. m Kode Pemrograman *Front-end*, src-components-navbar-(navbar.js) – Struktur komponen navigation-bar

```

import {
  ArrowLeftStartOnRectangleIcon,
  CubeIcon,
  CubeTransparentIcon,
  XMarkIcon,

```

```

} from "@heroicons/react/24/solid";
import { MenuIcon } from "./icon";
import { AnimatePresence, motion } from "framer-motion";
import { signOut, useSession } from "next-auth/react";
import Image from "next/image";
import Link from "next/link";
import { useState } from "react";

const Navbar = () => {
  const { data: session, status } = useSession();

  const [isOpen, setIsOpen] = useState(false);
  const [isSidebarOpen, setIsSidebarOpen] = useState(false);
  const dropdownVariants = {
    hidden: { opacity: 0, Scale: 0.95 },
    visible: { opacity: 1, Scale: 1 },
  };
};
return (
  <nav className="fixed top-0 z-50 w-full">
    <div className="w-full py-6 bg-white border-b drop-shadow-sm border-Neutral-100">
      <div className="container">
        <div className="flex items-center justify-between w-full text-pl-mobile lg:text-pl-desktop">
          <div>
            <Link href={`/${`}>
              <Image
                src={`/borlindo_logo.png`}
                alt="Logo"
                width={240}
                height={34}
                className="w-[146px] md:w-[214px] object-contain"
              />
            </Link>
          </div>
          <div>
            href={`/${`about`}
          >
            TENTANG BORLINDO
          </Link>
          <Link
            className="font-medium text-Neutral-700 hover:text-Neutral-900"
            href={`/${`information`}
          >
            INFORMASI PERWAKILAN
          </Link>
        </div>
      </div>
    </div>
  </nav>
)

```

```

<div className="relative justify-end hidden md:flex">
  {session ? (
    <div className="relative">
      <button
        onClick={() => {
          setIsOpen(!isOpen);
        }}
        className="relative flex items-center gap-2"
      >
        <Image
          src={` /profile.png`}
          alt="Profile Image"
          width={34}
          height={34}
          className="object-cover rounded-full"
        />
        <p className="font-medium text-cap-2-mobile text-Neutral-
700">
          {session?.name}
        </p>
      </button>
      <AnimatePresence>
        <Link
          href={` /dashboard`}
          className="flex items-center justify-between gap-2
px-4 py-2 text-left transition-colors cursor-pointer text-cap-2-mobile
hover:bg-Neutral-50"
        >
          Dashboard
          <CubeTransparentIcon className="w-5 h-5 fill-
Neutral-900" />
        </Link>
        <button
          onClick={() => {
            signOut();
          }}
          className="flex items-center justify-between gap-2
px-4 py-2 text-left transition-colors cursor-pointer text-cap-2-mobile
hover:bg-Neutral-50"
        >
          Keluar
          <ArrowLeftStartOnRectangleIcon className="w-5 h-5
fill-Neutral-900" />
        </button>
      </motion.div>
    )}
  </AnimatePresence>
</div>

```

Lampiran 3. n Kode Pemrograman *Front-end*, src-app-about-(page.js) – Tampilan Halaman Tentang Borlindo. Source Code terlampir hanya bagian awal dan akhir saja.

```

import React from 'react';
import Navbar from '~/components/navbar/navbar';
import Footer from '~/components/footer/footer';
import Image from 'next/image';

export default function AboutPage() {
  return (
    <main>
      <Navbar />
      <div className="min-h-screen container pt-28 text-center pb-20">
        <h1>Company Profile</h1>
        <div>
          <Image
            src={'/logo.png'}
            className="mx-auto mt-10"
          />
          <Image
            src={'/logo-text.png'}
            width={458}
            height={69}
            className="mx-auto"
          />
          <p className="mt-10 text-sm md:text-base">
            PO Borlindo (Borneo Lintas Indonesia) yang bernaung dibawah{' '}
            <b>PT Borlindo Mandiri Jaya</b>, merupakan perusahaan yang
            bergerak di bidang jasa transportasi umum yang berdiri sejak 5 Juni 2017
            dengan modal 2 unit armada yaitu BMJ-01 dan BMJ-02. Armada Borlindo saat ini
            berjumlah 22 unit didukung oleh Scania, Volvo dan Mercedes Benz yang merupakan
            produsen bus terkemuka di dunia.
          </p>
          .
          .
          .
          <div className="mt-20">
            <h3 className="text-dark-blue font-bold text-3xl md:text-6xl mb-5 font-serif text-shadow">
              MOTTO
            </h3>
            <p className="font-bold text-md md:text-2xl">
              We may not be the First <br />
              but We Always Provide the Best Service
            </p>
          </div>
          <Footer />
        </main>
      </div>
    </main>
  )
}

```

Lampiran 3. o Kode Pemrograman *Front-end*, *src-components-atomic-molecules-representative-table-(index.js)* – Tampilan Halaman Informasi Perwakilan. Source Code terlampir hanya bagian awal dan akhir saja.

```
'use client';
import React from 'react';
import Image from 'next/image';
import listData from './information.JSON';

export default function RepresentativeTable() {
  const [searchList, setSearchList] = React.useState(listData || []);
  const [keyword, setKeyword] = React.useState('');

  return (
    <div className="w-full ">
      <div className="bg-white shadow-2xl pb-10 rounded-lg w-full">
        <div className="flex justify-between items-center bg-[#67A3D9] px-5 md:px-10 py-2 w-full rounded-t-lg relative z-10">
          <div>
            
          </div>
          <h6 className="text-dark-blue font-bold text-sm md:text-2xl">
            Informasi Perwakilan
          </h6>
        </div>
        <div className="flex justify-center my-10 relative z-10">
          <div className="relative w-full md:w-3/4 px-4">
            <input
              const filteredData = listData.filter((el) => {
                return (
                  el?.provinsi
                    .toLowerCase()
                    .includes(e.target.value.toLowerCase()) ||
                  el?.kota
                    .toLowerCase()
                    .includes(e.target.value.toLowerCase()) ||
                  el?.alamat
                    .toLowerCase()
                    .includes(e.target.value.toLowerCase())
                );
              });
              setSearchList(filteredData);
            } else {
              setSearchList(listData);
            }
          </div>
          <div className="w-full mx-2 border rounded-md py-2 px-2 focus:outline-none "
            placeholder="Cari Perwakilan..."
          </div>
        </div>
      </div>
    </div>
  );
}
```

```

        />
        .
        .
        .
        {searchList?.length <= 0 && keyword.length > 0 ? (
            <tr>
                <td colSpan={4} className="text-center">
                    Maaf, kata kunci yang anda masukkan tidak terdapat di
list
                </td>
            </tr>
        ) : null}
    </tbody>
  );
}

```

Lampiran 3. p Kode Pemrograman *Front-end*, src-components-atomic-molecules-landingpage-(section-two.js) – Tampilan Halaman Experience. Source Code terlampir hanya bagian awal dan akhir saja.

```

import Image from "next/image";

const listExperience = [
  {
    id: 1,
    title: "KENYAMANAN DAN KEAMANAN ADALAH PRIORITAS KAMI",
    desc: "Kabin yang bersih, fasilitas yang steril, serta keamanan menjadi perhatian kami untuk menemani perjalanan anda.",
    url: "/experince1.png",
  },
  {
    id: 2,
    title: "ARMADA BUS BARU DAN TERBAIK",
    desc: "Dukungan perusahaan ternama dan terbaik di dunia, Scania dan Mercedes Benz menjadikan armada Borlindo menjadi yang terbaik.",
    url: "/experience2.png",
  },
  {
    id: 3,
    title: "KRU BUS YANG BERPENGALAMAN DAN RAMAH",
    desc: "Supir yang berpengalaman, serta pelayanan yang baik dan keramahan kru disetiap bus, akan menemani anda sepanjang perjalanan.",
    url: "/experience3.png",
  },
];

const LandingPageSectionTwo = () => {
  return (

```

```

<section>
  <div className="flex container items-center flex-col justify-center py-
16">
    <div className="text-center">
      <h1 className="text-h1-mobile lg:text-h1-desktop text-Neutral-900
font-bold">
        Experience
      </h1>
      <p className="italic text-pl-mobile lg:text-pl-desktop mt-2">
        Celebrate your trip with BORLINDO
      </p>
    </div>
    <div className="grid grid-cols-1 lg:grid-cols-3 gap-11 mt-10">
      {listExperience.map((item) => {
        return (
          <div key={item.id} className="relative">
            <div className="relative">
              <Image
                src={item.url}
                alt="Image"
                height={500}
                width={500}
                className="object-cover h-96 object-center w-full rounded-
xl drop-shadow-2xl shadow-xl"
              />
              <div className="absolute bottom-0 w-full flex items-end">
                <div className="bg-Neutral-900 bg-opacity-40 w-full p-4
rounded-b-xl">
                  <h1 className="relative z-10 text-white text-pl-mobile
italic lg:text-pl-desktop font-semibold">
                    {item.title}
                  </h1>
                </div>
              </div>
            </div>
            <p className="text-pr-mobile font-medium text-center lg:text-
pr-desktop text-Neutral-700 mt-6">
              {item.desc}
            </p>
          </div>
        );
      })}
    </div>
  </div>
</section>
);
};
export default LandingPageSectionTwo;

```

Lampiran 3. q Kode Pemrograman *Front-end*, src-components-atomic-molecules-landingpage-(section-three.js) – Tampilan Halaman Statistik Perusahaan

```
'use client';

import Image from 'next/image';
import { useEffect, useState } from 'react';
import { getStatistic } from '~/lib/services/statistic';

const LandingPageSectionThree = () => {
  const [data, setData] = useState(null);
  const handleGetData = async () => {
    try {
      const res = await getStatistic();
      if (res?.data?.length) {
        setData(res.data[0]);
      }
      // console.log(res);
      // console.log('data', res.data);
    } catch (err) {
      console.log(err);
    }
  };

  useEffect(() => {
    handleGetData();
  }, []);
  return (
    <section className="relative">
      <Image
        src={`~/Borlindo2.png`}
        alt="Jumbotron"
        fill
        priority={true}
        className="w-full object-cover"
      />
      <div className="flex items-center relative container z-10 justify-center min-h-screen">
        <div className="rounded-xl flex flex-col items-center w-full p-4 lg:p-8 text-white bg-Neutral-900 bg-opacity-70 backdrop-blur-xl">
          <h1 className="text-h1-mobile lg:text-h1-desktop font-semibold">
            PT. Borlindo Mandiri Jaya
          </h1>
          <p className="text-p1-mobile text-center lg:text-p1-desktop max-w-xl italic mt-2">
            Perusahaan Otobus dengan rute terlengkap dalam lintas pulau Sulawesi yang menawarkan kenyamanan serta kuliatas terbaik kepada masyarakat.{' '}

```



```

    </p>
    <div className="grid mt-10 grid-cols-2 md:grid-cols-3 gap-8">
      <div className="text-center">
        <p className="text-h6-desktop italic">Armada Bus</p>
        <h4 className="text-6xl mt-2 font-bold">{data?.armada || 0}</h4>
      </div>
      <div className="text-center">
        <p className="text-h6-desktop italic">Rute Inti</p>
        <h4 className="text-6xl mt-2 font-bold">{data?.route || 0}</h4>
      </div>
      <div className="text-center">
        <p className="text-h6-desktop italic">Perwakilan</p>
        <h4 className="text-6xl mt-2 font-bold">
          {data?.perwakilan || 0}
        </h4>
      </div>
    </div>
  </div>
</div>
</section>
);
};
export default LandingPageSectionThree;

```

Lampiran 3. r Kode Pemrograman *Front-end*, src-components-atomic-molecules-news section-(index.js) – Tampilan Halaman Berita Perusahaan. Source Code terlampir hanya bagian awal dan akhir saja.

```

'use client';

import React from 'react';
import newsList from './news.JSON';
import Image from 'next/image';
import { useGetAllNews } from '~/api/news/get-all-news';

export default function NewsSection() {
  const { data = [], isLoading, refetch } = useGetAllNews(1, 1000);

  return (
    <div className="relative overflow-hidden">
      <div className="absolute bg-acsen-1 w-[183px] h-[277px] right-0 top-0 "
      />
      <div className="absolute bg-acsen-2 w-[180px] h-[180px] left-0 bottom-0
      " />
      <div className="absolute bg-acsen-3 w-[200px] h-[200px] right-0 bottom-0
      " />
      <div className="overflow-hidden">

```

```

    <div className="absolute bg-acsen-4 w-[300px] h-[300px] top-0 left-28
" />
    <div className="absolute bg-acsen-4 w-[300px] h-[300px] top-0 left-0 "
/>
    <div className="absolute bg-acsen-4 bg-no-repeat w-[300px] h-[300px]
top-0 left-[-8rem] " />
    </div>
    <div className="container py-10 relative">
      <h1 className="text-[#666666] font-bold text-3xl md:text-5xl mb-4
uppercase text-shadow">
        Berita Terbaru
      </h1>
      .
      .
      .
    <div className="grid grid-cols-2 md:grid-cols-3 gap-5">
      {data?.data?.data?.map((news, index) => (
        <div key={index}>
          <div className="min-h-[140px] md:min-h-[202px] relative mb-2">
            <Image
              className="rounded-lg shadow-2xl mb-4 bg-cover bg-center"
              src={process.env.API_URL + '/' + news.image}
              alt={news?.title}
              objectFit="cover"
              fill
            </div>
            <div className="flex items-center gap-2">
              <div className="w-3 h-3 rounded-full bg-red-500" />
              <p className="text-red-500 font-semibold text-xs md:text-sm">
                {news?.date?.split('T')[0]}
              </p>
            </div>
            <h3 className="text-sm md:text-xl font-bold text-[#666666] text-
shadow mb-2">
              {news?.title}
            </h3>
            <a
              target="_blank"
              href={news?.url}
              rel="noopener noreferrer"
              className="font-bold text-sm md:text-lg text-[#E88E24]"
            >
              Baca Selengkapnya &gt;
            </a>
          </div>
        </div>
      )})
    </div>
  );
}

```

Lampiran 3. s Kode Pemrograman *Front-end*, src-components-atomic-dashboard-(home.js) – Tampilan Halaman Informasi. Source Code terlampir hanya bagian awal dan akhir saja.

```

"use client";

import { CubeIcon, ShoppingBagIcon } from "@heroicons/react/24/solid";
import { AnimatePresence, motion } from "framer-motion";
import moment from "moment";
import { useSession } from "next-auth/react";
import Image from "next/image";
import { useState } from "react";
import { useGetDashboardStatistic } from "~/api/dashboard/get-dashboard-
statistic";
import { useGetDetailTransaction } from "~/api/transaction/detail-
transaction";
import HomeLayout from "~/app/dashboard-layout";

const HomeDashboard = () => {
  const { data: session } = useSession();

  const [modalEtiket, setModalEtiket] = useState({
    id: null,
    modal: false,
  });

  //data fetching
  const { data, isLoading } = useGetDashboardStatistic(session?.jwt);
  const { data: resDetailTransaction, isLoading: isLoadingDetailTransaction }
=
  useGetDetailTransaction(session?.jwt, modalEtiket?.id);

  return (
    <>
      <HomeLayout>
        <div className=" md:flex hidden px-6 lg:px-8 py-3.5 border-b border-
Neutral-200">
          <h6 className="font-bold text-Neutral-900 text-h6-desktop">
            Dashboard
          </h6>
        </div>
        <div className="px-6 mt-12 lg:px-8 md:mt-0">
          <div className="grid grid-cols-1 gap-6 mt-10 lg:grid-cols-3">
            <div className="w-full p-6 border border-Neutral-200 rounded-2xl">
              <div className="p-2.5 rounded-lg flex items-center justify-
center w-fit bg-Success-50">
                <ShoppingBagIcon className="w-8 h-8 fill-Success-500" />
              </div>
              <h6 className="mt-4 font-bold text-h6-desktop text-Neutral-900">

```

```

        Total Pemesanan
    </h6>

    <div className="flex flex-col items-end justify-end mt-10">
        <h2 className="font-bold text-Neutral-900 text-h2-desktop">
            {data?.status === 200 ? data?.data?.totalTransactions : "0"}
        </h2>
        <div className="w-fit max-w-40">
            <p className="text-pr-mobile text-end lg:text-pr-desktop
text-Neutral-700">
                Total pemesanan dalam keseluruhan
            </p>
        </div>
    </div>
</div>
<div className="w-full p-6 border border-Neutral-200 rounded-2xl">
    <div className="p-2.5 rounded-lg flex items-center justify-
center w-fit bg-purple-50">
        <CubeIcon className="w-8 h-8 fill-purple-500" />
    </div>
    <h6 className="mt-4 font-bold text-h6-desktop text-Neutral-900">
        Total Bus
    </h6>

    <div className="flex flex-col items-end justify-end mt-10">
        <h2 className="font-bold text-Neutral-900 text-h2-desktop">
            {data?.status === 200 ? data?.data?.totalBus : "0"}
        </h2>
        <div className="w-fit max-w-40">
            <p className="text-pr-mobile text-end lg:text-pr-desktop
text-Neutral-700">
                Total bus yang tersedia saat ini
            </p>
        </div>
    </div>
</div>
.
.
.
<h4 className="font-bold text-h5-desktop text-Neutral-900">
    Your History Transaction
</h4>

    <div className="relative border border-Neutral-100 mt-10 max-h-
[540px] w-full overflow-x-auto font-poppins scrollbar-hide">
        <table className="relative w-full -mt-6 text-left border-
separate border-spacing-y-6 text-Secondary-900">
            <thead>

```

```

        <tr>
            <th className="sticky top-0 z-10 px-6 py-4 font-semibold
whitespace-nowrap bg-Shades-100 text-pr-mobile text-Neutral-900 xl:text-pr-
desktop">
                Transaction ID
            </th>
            <th className="sticky top-0 z-10 px-6 py-4 font-semibold
whitespace-nowrap bg-Shades-100 text-pr-mobile text-Neutral-900 xl:text-pr-
desktop">
                Seat Information
            </th>
            <th className="sticky top-0 z-10 px-6 py-4 font-semibold
whitespace-nowrap bg-Shades-100 text-pr-mobile text-Neutral-900 xl:text-pr-
desktop">
                Order Data
            </th>
            <th className="sticky top-0 z-10 px-6 py-4 font-semibold
whitespace-nowrap bg-Shades-100 text-pr-mobile text-Neutral-900 xl:text-pr-
desktop">
                Total Passenger
            </th>
            <th className="sticky top-0 z-10 px-6 py-4 font-semibold
whitespace-nowrap bg-Shades-100 text-pr-mobile text-Neutral-900 xl:text-pr-
desktop">
                Total Transaction
            </th>
            <th className="sticky top-0 z-10 px-6 py-4 font-semibold
whitespace-nowrap bg-Shades-100 text-pr-mobile text-Neutral-900 xl:text-pr-
desktop">
                Status
            </th>
            <th className="sticky top-0 z-10 px-6 py-4 font-semibold
whitespace-nowrap bg-Shades-100 text-pr-mobile text-Neutral-900 xl:text-pr-
desktop">
                Transaction Time
            </th>
            <th className="sticky top-0 z-10 px-6 py-4 font-semibold
whitespace-nowrap bg-Shades-100 text-pr-mobile text-Neutral-900 xl:text-pr-
desktop">
                Payment Status
            </th>
            <th>
                Action
            </th>
        </tr>
    </thead>

```

```

.
.
.
      <div className="flex items-center justify-between px-4 py-6
border-b lg:px-6 border-Neutral-100">
        <h5 className="font-bold text-Neutral-900 text-h5-desktop">
          E-Ticket
        </h5>
        <div>
          <p className="font-medium text-pr-mobile lg:text-pr-
desktop text-Neutral-700">
            Kode E Tiket
          </p>
          <p className="font-bold text-Neutral-900 text-h6-desktop">
            {resDetailTransaction?.status === 200
              ? resDetailTransaction?.data?.data?.transaction_id
              : "-"}
          </p>
        </div>
      </div>
      <div>
        Nama Pemesan
      </p>
      <p className="mt-2 text-Neutral-900 text-pr-mobile
lg:text-pr-desktop">
        {resDetailTransaction?.status === 200
          ?
          `${resDetailTransaction?.data?.data?.order_data?.first_name}
${resDetailTransaction?.data?.data?.order_data?.last_name}`
          : "-"}
      </p>
    </div>
    <div>
      <p className="font-semibold text-pr-mobile lg:text-pr-
desktop text-Neutral-700">
        Nomor Pemesan
      </p>
      <p className="mt-2 text-Neutral-900 text-pr-mobile
lg:text-pr-desktop">
        {resDetailTransaction?.status === 200
          ? resDetailTransaction?.data?.data?.order_data
            ?.phone_number
          : "-"}
      </p>
    </div>
    <div>
      <p className="font-semibold text-pr-mobile lg:text-pr-
desktop text-Neutral-700">
        Total Penumpang

```

```

        </p>
        <p className="mt-2 text-Neutral-900 text-pr-mobile
lg:text-pr-desktop">
            {resDetailTransaction?.status === 200
            ?
            `${resDetailTransaction?.data?.data?.passenger_data.length} Orang`
            : "-"}
            Bus Information
        </p>
        <div className="mt-2">
            <p className="text-Neutral-900 text-pr-mobile lg:text-
pr-desktop">
                {resDetailTransaction?.status === 200
                ?
                resDetailTransaction?.data?.data?.bus_information
                ?.bus_id?.busName
                : "-"}
            </p>
            <div className="flex items-center gap-2">
                <p className="text-Neutral-900 text-pr-mobile
lg:text-pr-desktop">
                    {resDetailTransaction?.status === 200
                    ? resDetailTransaction?.data?.data
                    ?.bus_information?.bus_id?.from?.name
                    : "-"}
                </p>
                {" - "}
                <p className="text-Neutral-900 text-pr-mobile
lg:text-pr-desktop">
                    {resDetailTransaction?.status === 200
                    ? resDetailTransaction?.data?.data
                    ?.bus_information?.bus_id?.to?.name
                    : "-"}
                </p>
            </div>
            .
            .
            .
        >
        <div>
            <button>
                Close
            </button>
        </div>
    </motion.div>
    )}
    );
};

export default HomeDashboard;

```

Lampiran 3. t Kode Pemrograman *Front-end*, *src-components-atomic-dashboard-bus control - (bus-control.js)* – Tampilan Halaman Kontrol Bus

```
'use client';

import { AnimatePresence, motion } from 'framer-motion';
import moment from 'moment';
import { useSession } from 'next-auth/react';
import Image from 'next/image';
import { useEffect, useState } from 'react';
import { useAddSeats } from '~/api/bus/add-seat';
import { useCreateBus } from '~/api/bus/create-bus';
import { useGetAllListBus } from '~/api/bus/get-all-list-bus';
import { useGetDetailBus } from '~/api/bus/get-detail-bus';
import SelectInput from '~/components/atomic/select';
import { deleteBus } from '~/lib/services/bus';
import { getActiveRoute } from '~/lib/services/route';
import ModalAddBus from './modal-bus';
import ModalKursi from './modal-kursi';

const BusControl = () => {
  //get data session
  const { data: session, status } = useSession();
  const [listRoute, setListRoute] = useState([]);
  const [selectedTujuan, setSelectedTujuan] = useState(null);
  const [selectedAsal, setSelectedAsal] = useState(null);
  //Modal
  const [modal, setModal] = useState({
    modalAddBus: false,
    modalAddSeat: false,
  });

  //alert
  const [alert, setAlert] = useState(false);
  //pagination
  const [page, setPage] = useState(1);
  const [limit, setLimit] = useState(20);

  //Search state
  const [searchState, setSearchState] = useState({
    province: '',
    regencie: '',
    district: '',
  });

  //tujuan
  provinceTujuan: '',
  regencieTujuan: '',
  districtTujuan: '',

```



```

    departureTime: '',
  });

  //form state
  const [formCreateBus, setFormCreateBus] = useState({
    busName: '', //String
    totalSeats: 1, //Number
    departureTime: '',
    arrivalTime: '',
    rangePrice: {
      min: 0,
      max: 0,
    },
  });

  //form add seats
  const [formAddSeats, setFormAddSeats] = useState({
    busId: '',
    seatDetails: {
      seatNumber: '',
      classType: '',
      floor: '',
      price: '',
    },
    classType: '',
  });

  //fetching create bus
  const { data: resAddBus, mutate, isPending } = useCreateBus();
  //fetching add seats
  const {
    data: resAddSeat,
    mutate: mutateAddSeats,
    isPending: isPendingAddSeats,
  } = useAddSeats();
  //fetching detail bus
  const {
    data: resDetailBus,
    isLoading: isLoadingDetailBus,
    refetch: refetchDetailBus,
  } = useGetDetailBus(formAddSeats?.busId);
  .
  .
  .
  </>
  );
};
export default BusControl;

```

Lampiran 3. u Kode Pemrograman *Front-end*, src-components-atomic-dashboard-transaction control - (transaction-control.js) – Tampilan Halaman Kontrol Transaksi

```

"use client";
import { Menu } from "@headlessui/react";
import { CheckCircleIcon } from "@heroicons/react/24/solid";
import { PDFDownloadLink } from "@react-pdf/renderer";
import { AnimatePresence, motion } from "framer-motion";
import moment from "moment";
import { useSession } from "next-auth/react";
import Image from "next/image";
import { useEffect, useState } from "react";
import { useGetDetailTransaction } from "~/api/transaction/detail-transaction";
import { useGetListTransaction } from "~/api/transaction/get-list-transaction";
import { useUpdateTransactions } from "~/api/transaction/update-transaction";
import Input from "~/components/atomic/input";
import { deleteTransaction } from "~/lib/services/transaction";
import PdfTemplate from "../../pdf-template";

const listStatus = [
  {
    id: 1,
    status: "Pending",
  },
  {
    id: 2,
    status: "Success",
  },
  {
    id: 3,
    status: "Cancelled",
  },
];

export const TransactionControl = () => {
  const { data: session, status } = useSession();
  const [startDate, setStartDate] = useState("");
  const [endDate, setEndDate] = useState("");
  const [search, setSearch] = useState("");

  //pagination
  const [page, setPage] = useState(1);
  const [limit, setLimit] = useState(20);

  //modal
  const [modal, setModal] = useState({
    id: "",
  });

```

```

    modalDetailTransaction: false,
  });
  //option
  const [updateStatusOptions, setUpdateStatusOptions] = useState(false);

  //data fetching
  const { data, isLoading, refetch } = useGetListTransaction(
    session?.jwt,
    page,
    limit,
    startDate,
    endDate,
    search
  );
  .
  .
  .
  })

      <div className="flex gap-4 px-4 pb-4 mt-8">
        <button
          className="w-full px-4 py-3 border rounded-lg text-pr-mobile
lg:text-pr-desktop border-Neutral-200 text-Neutral-700"
          onClick={() => {
            setModal({ ...modal, modalDetailTransaction: false });
          }}
        >
          Tutup
        </button>
      </div>
    </motion.div>
    <div className="fixed inset-0 bg-black opacity-50"></div>
  </motion.div>
  })
</AnimatePresence>
</>
);
};

export default TransactionControl;

```

Lampiran 3. v Kode Pemrograman *Front-end*, *src-components-atomic-dashboard-rute-control* - (*rute-control.js*) – Tampilan Halaman Kontrol Rute

```

'use client';

import { useSession } from 'next-auth/react';
import { useState } from 'react';
import ModalRute from './modal-rute';

```

```
import { useFormik } from 'formik';
import { useCreateRoute } from '~/api/route/create-route';
import { useGetAllRoute } from '~/api/route/get-all-route';
import { toast } from 'react-toastify';
import { deleteRoute, updateRoute } from '~/lib/services/route';

export const RouteControlTable = () => {
  const { data: session, status } = useSession();
  const [isOpen, setIsOpen] = useState(false);
  // const [isLoading, setIsLoading] = useState(false);

  //pagination
  const [page, setPage] = useState(1);
  const [limit, setLimit] = useState(20);

  const { data, isLoading, refetch } = useGetAllRoute(page, limit);
  const { data: resAddBus, mutate, isPending } = useCreateRoute();

  // console.log(data);

  const handleSubmit = async (values) => {
    mutate(
      {
        ...values,
        token: session?.jwt,
      },
      {
        onSuccess: (res) => {
          if (res?.data?.status === 200) {
            toast.success('Success add route');
            setIsOpen(false);
            formik.resetForm();
            refetch();
          } else {
            toast.error(res?.response?.data?.message || 'Failed add route');
          }
        },
      },
      {}
    );
  };

  const formik = useFormik({
    initialValues: {
      id: '',
      code: '',
      name: '',
      isActive: true,
    },
  });
};
```

```

    },
    onSubmit: handleSubmit,
  });
.
.
.
</div>
  <ModalRute
    formik={formik}
    isOpen={isOpen}
    handleClose={() => {
      setIsOpen(false);
    }}
    onSubmit={() => {
      if (formik.values.id) {
        handleUpdate(formik.values);
      } else {
        handleSubmit(formik.values);
      }
    }}
  />
</>
);
};

export default RuteControlTable;

```

Lampiran 3. w Kode Pemrograman *Front-end*, src-components-atomic-dashboard-news control - (news-control.js) – Tampilan Halaman Kontrol Berita

```

'use client';

import { useSession } from 'next-auth/react';
import { useState } from 'react';
import { useFormik } from 'formik';
import { useCreateRoute } from '~/api/route/create-route';
import { useGetAllRoute } from '~/api/route/get-all-route';
import { toast } from 'react-toastify';
import { deleteRoute, updateRoute } from '~/lib/services/route';
import ModalNews from './modal-news';
import { useGetAllNews } from '~/api/news/get-all-news';
import Image from 'next/image';
import { useCreateNews } from '~/api/news/create-news';
import { deleteNews, updateNews } from '~/lib/services/news';

export const NewsControlTable = () => {
  const { data: session, status } = useSession();

```

```
const [isOpen, setIsOpen] = useState(false);
// const [isLoading, setIsLoading] = useState(false);

//pagination
const [page, setPage] = useState(1);
const [limit, setLimit] = useState(20);

const { data, isLoading, refetch } = useGetAllNews(page, limit);
const { data: resAddBus, mutate, isPending } = useCreateNews();

// console.log(data);

const handleSubmit = async (values) => {
  mutate(
    {
      ...values,
      token: session?.jwt,
    },
    {
      onSuccess: (res) => {
        console.log(res);
        if (res?.data?.status === 200) {
          toast.success('Success add news');
          setIsOpen(false);
          formik.resetForm();
          refetch();
        } else {
          toast.error('Failed add news');
        }
      },
      onError: (err) => {
        console.log(err);
        toast.error('Failed add news');
      },
    },
    {}
  );
};

const formik = useFormik({
  initialValues: {
    id: '',
    title: '',
    date: '',
    image: '',
    url: '',
  },
});
```

```

    onSubmit: handleSubmit,
  });
.
.
.
  <div className="relative mt-4 w-full overflow-x-auto rounded-t-lg border
border-Neutral-200">
    <thead>
      <tr>
        <th className="p-4 text-left text-pr-mobile font-medium
text-Neutral-900 xl:text-pr-desktop">
          No
        </th>
        <th className="p-4 text-left text-pr-mobile font-medium
text-Neutral-900 xl:text-pr-desktop">
          Judul Berita
        </th>
        <th className="p-4 text-left text-pr-mobile font-medium
text-Neutral-900 xl:text-pr-desktop">
          Gambar
        </th>
        <th className="p-4 text-center text-pr-mobile font-medium
text-Neutral-900 xl:text-pr-desktop">
          Url
        </th>
        <th className="p-4 text-center text-pr-mobile font-medium
text-Neutral-900 xl:text-pr-desktop">
          Action
        </th>
        <td
          colspan={5}
          className="text-center py-4 text-Neutral-900"
        >
          loading...
        </td>
      </tr>
    ) : data?.data?.data?.length === 0 ? (
      <tr>
        <td
          colspan={5}
          className="text-center py-4 text-Neutral-900"
        >
          Data not found
        </td>
      </tr>
    ) : (
      <tbody>
        <tr>
          <td>No</td>
          <td>Judul Berita</td>
          <td>Gambar</td>
          <td>Url</td>
          <td>Action</td>
        </tr>
      </tbody>
    )
  </div>
}

export default NewsControlTable;

```

Lampiran 3. x Kode Pemrograman *Front-end*, src-components-atomic-dashboard-statistic control - (statistic-control.js) – Tampilan Halaman Kontrol Statistik

```
'use client';

import { useSession } from 'next-auth/react';
import { useState } from 'react';
import { useFormik } from 'formik';
import { toast } from 'react-toastify';
import { deleteStatistic, updateStatistic } from '~/lib/services/statistic';
import ModalStatistic from './modal-statistic';
import { useGetAllStatistic } from '~/api/statistic/get-all-statistic';
import { useCreateStatistic } from '~/api/statistic/create-statistic';

export const StatisticControlTable = () => {
  const { data: session, status } = useSession();
  const [isOpen, setIsOpen] = useState(false);
  // const [isLoading, setIsLoading] = useState(false);

  const { data, isLoading, refetch } = useGetAllStatistic();
  const { data: resAddBus, mutate, isPending } = useCreateStatistic();

  // console.log(data);

  const handleSubmit = async (values) => {
    mutate(
      {
        ...values,
        token: session?.jwt,
      },
      {
        onSuccess: (res) => {
          console.log(res);
          if (res?.data?.status === 200) {
            toast.success('Success add statistic');
            setIsOpen(false);
          }
        }
      }
    );
  };

  const formik = useFormik({
    initialValues: {
      id: '',
      armada: '',
      route: '',
      perwakilan: '',
    },
    onSubmit: handleSubmit,
  });
}

<div>
  <th className="p-4 text-center text-pr-mobile font-medium
text-Neutral-900 xl:text-pr-desktop">
```



```

        Jumlah Armada
      </th>
      <th className="p-4 text-center text-pr-mobile font-medium
text-Neutral-900 xl:text-pr-desktop">
        Jumlah Rute
      </th>
      <th className="p-4 text-center text-pr-mobile font-medium
text-Neutral-900 xl:text-pr-desktop">
        Jumlah Perwakilan
      </th>
      <th className="p-4 text-center text-pr-mobile font-medium
text-Neutral-900 xl:text-pr-desktop">
        Action
      </th>
    </tr>
    <td className="text-center py-4 text-Neutral-900"
    >
      loading...
    </td>
  </tr>
) : (data?.data?.data || [])?.length === 0 ? (
  <tr>
    <td
      colspan={5}
      className="text-center py-4 text-Neutral-900"
    >
      Data not found
    </td>
  </tr>
)
export default StatisticControlTable;

```

Lampiran 3. y Kode Pemrograman *Front-end*, src-components-footer-(footer.js) – Struktur komponen footer

```

import Image from 'next/image';

const partner = [
  {
    id: 1,
    url: '/partner/image 15.png',
  },
];

const Footer = () => {
  return (
    <>
      <footer className="bg-[#CFCFCF]">
        <div className="container w-full relative pt-20 pb-8">
          <Image

```

```

    src={` /borlindo_logo.png`}
    alt="Logo"
    width={240}
    height={34}
    className="w-[146px] md:w-[214px] object-contain"
  />
  <p className="font-bold">PT. Borlindo Mandiri Jaya</p>
  <div className="mt-2">
    <p className="font-bold">Head Office :</p>
    <p className="">
      Jl. Jalur, Lingkar Barat, Tamalanrea, Makassar, Sulawesi
      Selatan, 90245
    </p>
  </div>
  <div className="mt-2">
    <p className="font-bold">Contact Us </p>
    <div className="">
      <div className="flex ">
        <p className="min-w-[8rem]">WhatsApp</p>
        <p className="">: +62 813-5444-0081</p>
      </div>
      <div className="flex ">
        <p className="min-w-[8rem]">Email</p>
        <p className="">: borlindomj@gmail.com</p>
      </div>
    <div className="col-span-1 lg:col-span-5">
      <h5 className="text-h3-mobile lg:text-h3-desktop font-bold text-
Neutral-900">
        OUR PARTNER{' '}
      </h5>
      <div className="grid grid-cols-3 md:grid-cols-5 gap-5 mt-10">
        {partner.map((item) => {
          return (
            <Image
              className="my-auto"
              src={item.url}
              key={item.id}
              alt="partner"
              width={100}
              height={100}
            />
          )
        })}
      </div>
      <div className="p-6 bg-[#373737]">
        © 2024 PT. Borlindo Mandiri Jaya. All rights reserved.
      </div>
    </footer>
  </>
  );
};
export default Footer;

```

Lampiran 4. Kode Pemrograman *Back-end* Website

Lampiran 4. a Kode Pemrograman *Back-end*, (Package.JSON) – Inisiasi Library

```
{
  "name": "borlindo_be",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "type": "module",
  "scripts": {
    "start": "node index.js",
    "dev": "nodemon --watch . --exec \"node\" index.js"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "bcryptjs": "^2.4.3",
    "body-parser": "^1.20.2",
    "cloudinary": "^2.0.3",
    "compression": "^1.7.4",
    "cookie-session": "^2.1.0",
    "cors": "^2.8.5",
    "crypto-js": "^4.2.0",
    "dotenv": "^16.4.4",
    "express": "^4.18.2",
    "JSONwebtoken": "^9.0.2",
    "moment": "^2.30.1",
    "mongoose": "^8.4.1",
    "morgan": "^1.10.0",
    "multer": "^1.4.5-lts.1",
    "node-cron": "^3.0.3",
    "rotating-file-stream": "^3.2.1",
    "vercel": "^34.2.7"
  }
}
```

Lampiran 4. b Kode Pemrograman *Back-end*, (.env) – API Endpoints *ENV* tersamarkan.

```
PORT = 4000
APP_URL = - '
MONGOURL=-
SECRET_KEY=-3BvjtcXwNerRyhAFnYCseCAxtEiPA
CLOUD_NAME=-
API_KEY=xy2oq3j7FKfmTbETpjCa3NQq7DzRiomISXiLqZ6IH6BDaJ700w7NBn4BdByncTNB
API_SECRET=8IATdp9Gsi9ze9mBKq7VxQ8q
```

Lampiran 4. c Kode Pemrograman *Back-end*, controllers-Auth-(auth.controllers.js) - Validasi Data User

```

import bcrypt from "bcryptjs";
import jwt from "JSONwebtoken";
import moment from "moment";
import roleModel from "../../models/Auth/Roles.models.js";
import userModel from "../../models/Auth/Users.models.js";

export const authSignUp = async (req, res) => {
  const { username, name, email, password, roles } = req.body;

  try {
    if (roles === "Super Admin") {
      return res.status(401).send({ message: "You can't create super admin"
});
    }
    // Buat objek pengguna
    const userDetails = new userModel({
      username,
      name,
      email,
      password: bcrypt.hashSync(password, 8),
      createdAt: moment().format("YYYY MMMM DD hh:mm a"),
      roles: null, // Kami akan mengaturnya nanti
    });
    // Cari peran (role) yang sesuai
    const role = await roleModel.findOne({ name: roles });
    if (!role) {
      return res.status(401).send({ message: "Role Not Found" });
    }
    // Set peran pada pengguna
    userDetails.roles = role._id;
    // Simpan pengguna
    const userSaved = await userDetails.save();
    if (!userSaved) {
      return res.status(401).send({ message: "Pengguna Gagal Dibuat" });
    }
    return res.status(200).send({
      message: "Pengguna Berhasil Dibuat",
      data: userDetails,
    });
  } catch (error) {
    return res.status(500).send({ message: error.message });
  }
};

export const authSignIn = async (req, res) => {
  let userExists;

```

```

try {
  userExists = await userModel
    .findOne({ username: req.body.username })
    .populate("roles");
} catch (error) {
  console.log(error);
  // res.status(500).send({ message: error });
}

if (!userExists) {
  res.status(404).send({ message: "User Not Found" });
} else {
  let passwordIsValid = bcrypt.compareSync(
    req.body.password,
    userExists.password
  );

  if (!passwordIsValid) {
    res.status(404).send({ message: "Password Is Invalid" });
  } else {
    let token = jwt.sign({ id: userExists._id }, process.env.SECRET_KEY, {
      expiresIn: 86400, // 24 hours
    });

    res.status(200).send({
      id: userExists._id,
      name: userExists.name,
      username: userExists.username,
      email: userExists.email,
      roles: userExists.roles,
      accessToken: token,
    });
  }
}
};

```

Lampiran 4. d Kode Pemrograman *Back-end*, controllers-Bus-(bus.controllers.js) - Validasi Data Bus

```

import BusModels from '../models/main_models/Bus.models.js';

// Controller to create a new bus
const createBusControllers = async (req, res) => {
  try {
    const { from, to, departureTime } = req.body;
    //Chek apakah ada bus dengan keberangkatan yang sama dari lokasi awal dan
    lokasi tujuan serta jam yang sama

```

```

const existingBus = await BusModels.findOne({
  'from.name': from.name,
  'to.name': to.name,
  departureTime,
});
if (existingBus) {
  return res.status(400).JSON({
    message:
      'Bus with the same from, to, and departure time already exists',
  });
}

const { busName, arrivalTime, classType } = req.body;
const newBus = new BusModels({
  busName,
  from,
  to,
  departureTime,
  arrivalTime,
  classType,
});

await newBus.save();

if (req.query.asal) {
  query['from.id'] = req.query.asal;
}
if (req.query.tujuan) {
  query['to.id'] = req.query.tujuan;
}
if (req.query.departureTime) {
  query.departureTime = { $gte: new Date(req.query.departureTime) };
}
.
.
.

export {
  createBusControllers,
  getListBusControllers,
  addSeatToBusControllers,
  getDetailBusController,
  deleteBusControllers,
  addMultipleSeatToBusControllers,
};

```

Lampiran 4. e Kode Pemrograman *Back-end*, controllers-dashboard-(dashboard.controllers.js) ↯ - Validasi Data dashboard

```
import userModels from "../../models/Auth/Users.models.js";
import BusModels from "../../models/main_models/Bus.models.js";
import TransactionModels from
"../../models/main_models/Transaction.models.js";

export const DashboardStatistic = async (req, res) => {
  try {
    //Total bus yang tersedia
    const totalBus = await BusModels.countDocuments();
    //find user from id req.userId
    const user = await userModels.findById(req.userId).populate("roles");
    let totalTransactions;
    if (user?.roles?.name === "admin") {
      totalTransactions = await TransactionModels.countDocuments();
    } else {
      totalTransactions = await TransactionModels.countDocuments({
        order_by: req.userId,
      });
    }

    //List History Transaction
    const historyTranasction = await TransactionModels.find({
      order_by: req.userId,
    })
      .populate("order_by")
      .populate({
        path: "bus_information.bus_id",
        model: "Bus",
      });

    return res.status(200).JSON({
      totalBus,
      totalTransactions,
      historyTranasction,
    });
  }
};
```

Lampiran 4. f Kode Pemrograman *Back-end*, controllers-news-(news.controllers.js) - Validasi Data Berita. Source Code terlampir hanya bagian awal dan akhir saja.

```
import NewsModels from '../../models/main_models/News.models.js';
import { uploadMulter } from '../multer.js';
import mongoose from 'mongoose';

// Get all news
```

```

export const getNews = async (req, res) => {
  try {
    let query = {};
    // Menambahkan pagination
    let page = parseInt(req.query.page) || 1; // Default ke halaman 1 jika
    tidak ditentukan
    let limit = parseInt(req.query.limit) || 10; // Default limit per halaman
    adalah 10 jika tidak ditentukan
    let skip = (page - 1) * limit; // Menghitung jumlah dokumen yang harus
    dilewati

    // Menghitung total dokumen
    const totalDocuments = await NewsModels.countDocuments(query);

    // Menghitung total halaman
    const totalPages = Math.ceil(totalDocuments / limit);

    // Mendapatkan dokumen dengan limit dan skip
    const news = await NewsModels.find(query).skip(skip).limit(limit);

    // Menentukan apakah ada halaman selanjutnya atau sebelumnya
    const hasNextPage = page < totalPages;
    const hasPrevPage = page > 1;

    res.status(200).JSON({
      message: 'List of news',
      data: news,
      currentPage: page,
      totalPages: totalPages,
      totalDocuments: totalDocuments,
      nextPage: hasNextPage,
      prevPage: hasPrevPage,
    });
  }
  .
  .
  .
  export const updateNews = async (req, res) => {
    //update news based on create news function
    try {
      uploadMulter.single('image')(req, res, async (err) => {
        const { id } = req.params;
        const news = await NewsModels.findById(id);
        if (!news) {
          return res.status(404).JSON({ message: 'News not found' });
        }
        news.title = req.body.title;
        news.date = req.body.date;
        news.url = req.body.url;
      });
    }
  }
}

```



```

    if (req.file) {
      news.image = req.file.path;
    }
    await news.save();
    res.status(200).JSON({
      message: 'News updated successfully',
      status: 200,
    });
  });
} catch (error) {
  res.status(409).JSON({ message: error.message });
}
};

// Delete news
export const deleteNews = async (req, res) => {
  try {
    //find routecode and delete
    const news = await NewsModels.findOneAndDelete({
      _id: req.params.id,
    });
    if (!news) {
      return res.status(404).JSON({ message: 'News not found' });
    }
    res.status(200).JSON({ message: 'News deleted successfully', status: 200 });
  } catch (error) {
    res
      .status(500)
      .JSON({ message: 'Failed to delete news', error: error.message });
  }
};

```

Lampiran 4. g Kode Pemrograman *Back-end*, controllers-route control-(route.controllers.js) - Validasi Data Rute

```

import RouteModels from '../models/main_models/Route.models.js';

// Controller to create a new route
const createRouteControllers = async (req, res) => {
  try {
    const { code, name, isActive } = req.body;
    //periksa apakah kode rute sudah ada
    const existingRoute = await RouteModels.findOne({ routeCode: code });
    if (existingRoute) {
      return res.status(400).JSON({ message: 'Route code already exists' });
    }
    const newRoute = new RouteModels({

```

```

    routeCode: code,
    routeName: name,
    isActive: isActive,
  });
  await newRoute.save();
  res
    .status(200)
    .JSON({ message: 'Route created successfully', status: 200 });
} catch (error) {
  res
    .status(500)
    .JSON({ message: 'Failed to create route', error: error.message });
}
};
.
.
.
//get all active routes
const getActiveRoutes = async (req, res) => {
  try {
    const activeRoutes = await RouteModels.find({ isActive: true });
    res
      .status(200)
      .JSON({ message: 'Success', data: activeRoutes, status: 200 });
  } catch (error) {
    res
      .status(500)
      .JSON({ message: 'Failed to get active routes', error: error.message });
  }
};

export {
  createRouteControllers,
  getRouteControllers,
  getRouteByIdControllers,
  updateRouteControllers,
  deleteRouteControllers,
  getActiveRoutes,
};

```

Lampiran 4. h Kode Pemrograman *Back-end*, controllers-statistic-(statistic.controllers.js) - Validasi Data Statistik

```

import StatisticModels from '../models/main_models/Statistic.models.js';

export const getHomeStatistic = async (req, res) => {
  try {
    const statistic = await StatisticModels.find();

```

```
    res.status(200).JSON({ data: statistic, status: 200 });
  } catch (error) {
    res.status(404).JSON({ message: error.message });
  }
};

export const createStatistic = async (req, res) => {
  try {
    const newstatistic = new StatisticModels(req.body);
    await newstatistic.save();
    res
      .status(200)
      .JSON({ message: 'Statistic created successfully', status: 200 });
  } catch (error) {
    res
      .status(500)
      .JSON({ message: 'Failed to create statistic', error: error.message });
  }
};

export const updateStatistic = async (req, res) => {
  try {
    const { armada, route, perwakilan } = req.body;

    const statistic = await StatisticModels.findOneAndUpdate({
      _id: req.params.id,
    });
    if (!statistic) {
      return res.status(404).JSON({ message: 'Statistic not found' });
    }
    statistic.armada = armada;
    statistic.route = route;
    statistic.perwakilan = perwakilan;
    await statistic.save();
    res
      .status(200)
      .JSON({ message: 'Statistic updated successfully', status: 200 });
  } catch (error) {
    res
  }
};

export const deleteStatistic = async (req, res) => {
  try {
    //find routecode and delete
    const data = await StatisticModels.findOneAndDelete({
      _id: req.params.id,
    });
  }
};
```

Lampiran 4. i Kode Pemrograman Back-end, controllers-transaction-(transaction.controllers.js) - Validasi Data Transaksi. Source Code terlampir hanya bagian awal dan akhir saja.

```
import BusModels from "../../models/main_models/Bus.models.js";
import TransactionModels from
"../../models/main_models/Transaction.models.js";
import { uploadMulter } from "../multer.js";

// Controller untuk membuat transaksi
const createTransaction = async (req, res) => {
  const { order_data, passenger_data, bus_information } = req.body;

  try {
    // Fungsi untuk menghasilkan angka acak dengan panjang 6-7 digit
    const generateRandomTransactionId = () => {
      const min = 100000; // Angka terkecil dengan 6 digit
      const max = 9999999; // Angka terbesar dengan 7 digit
      return Math.floor(Math.random() * (max - min + 1)) + min;
    };

    // Generate unique 6-7 digit transaction_id
    let transactionId = generateRandomTransactionId(); // Fungsi untuk
menghasilkan transaction_id acak
    const existingTransaction = await TransactionModels.findOne({
      transaction_id: transactionId,
    });

    // Membuat transaction_id unik dengan panjang 6-7 digit
    while (existingTransaction) {
      transactionId = generateRandomTransactionId();
      existingTransaction = await TransactionModels.findOne({
        transaction_id: transactionId,
      });
    }

    const newTransaction = new TransactionModels({
      transaction_id: transactionId,
      order_data,
      passenger_data,
      bus_information,
      total_transaction: req.body.total_transaction,
      order_by: req.userId,
    });

    await newTransaction.save();

    //Dapatkan informasi bus dan rubah seat yang di booking menjadi true agar
tidak ada duplicate pemesanan
```

```

const bus = await BusModels.findById(bus_information.bus_id);
bus_information.seat.map((seat) => {
  const index = bus.seats.findIndex(
    (busSeat) => busSeat.seatNumber === seat?.seatNumber
  );
  bus.seats[index].isBooked = true;
});
await bus.save();
.
.
.
const deleteTransaction = async (req, res) => {
  try {
    const { id } = req.params;
    const transaction = await TransactionModels.findByIdAndDelete(id);
    if (!transaction) {
      return res.status(404).JSON({ message: "Transaction not found" });
    }
    res
      .status(200)
      .JSON({ message: "Transaction deleted successfully", status: 200 });
  } catch (err) {
    res.status(500).JSON({
      message: "Failed to delete transaction",
      error: err.message,
    });
  }
};

export {
  createTransaction,
  getListTransaction,
  updateController,
  uploadAttachmentPayment,
  deleteTransaction,
};

```

Lampiran 4. j Kode Pemrograman *Back-end*, controllers-(multer.js) - Validasi Data Upload File

```

import multer from "multer";

});

// Konfigurasi multer
const uploadMulter = multer({ storage: storage });

export { uploadMulter };

```

Lampiran 4. k Kode Pemrograman *Back-end*, routes-Auth-(auth.routes.js) - Navigasi / URL Mapping Informasi signin / signup

```
import express from "express";
import {
  authSignIn,
  authSignUp,
  changePassword,
  changePasswordAdmin,
  deleteUser,
  detailUser,
  loggedInfoUser,
  signout,
} from "../../controllers/Auth/auth.controllers.js";
import { isAdmin, verifyToken } from "../../middlewares/Auth/authJwt.js";
import {
  checkDuplicateUsernameAndEmail,
  checkRolesExist,
} from "../../middlewares/Auth/verifySignup.js";

const router = express.Router();

// Signup Endpoint
router.post(
  "/api/v1/auth/signup",
  [checkDuplicateUsernameAndEmail, checkRolesExist],
  authSignUp
);
// Signin Endpoint
router.post("/api/v1/auth/signin", authSignIn);
// Signout Endpoint
router.post("/api/v1/auth/signout", signout);

export { router as authRoutes };
```

Lampiran 4. l Kode Pemrograman *Back-end*, routes-Bus-(bus.routes.js) - Navigasi / URL Mapping Bus

```
import express from 'express';
import {
  addMultipleSeatToBusControllers,
  addSeatToBusControllers,
  createBusControllers,
  deleteBusControllers,
  getDetailBusController,
  getListBusControllers,
} from '../../controllers/Bus/bus.controllers.js';
```

```

import { IsAdmin, verifyToken } from '../../middlewares/Auth/authJwt.js';

const router = express.Router();

// Create Bus
router.post('/api/v1/bus', [verifyToken, IsAdmin], createBusControllers);

//Detail Bus
router.get('/api/v1/bus/:id', getDetailBusController);

// Get List Bus
router.get('/api/v1/bus', getListBusControllers);

// Add Seat to Bus
router.post(
  '/api/v1/bus/add-seat',
  [verifyToken, IsAdmin],
  addSeatToBusControllers
);

//add multiple seat to bus
router.post(
  '/api/v1/bus/add-multiple-seat',
  [verifyToken, IsAdmin],
  addMultipleSeatToBusControllers
);

//delete bus
router.delete('/api/v1/bus/:id', [verifyToken, IsAdmin],
deleteBusControllers);

export { router as busRoutes };

```

**Lampiran 4. m Kode Pemrograman *Back-end*, routes-dashboard-(dashboard. routes.js)
- Navigasi / URL Mapping dashboard**

```

import express from "express";
import { DashboardStatistic } from
"../../controllers/dashboard/dashboard.controllers.js";
import { verifyToken } from "../../middlewares/Auth/authJwt.js";

const router = express.Router();

// Dashboard Statistic
router.get("/api/v1/dashboard", [verifyToken], DashboardStatistic);

export { router as dashboardRouter };

```

Lampiran 4. n Kode Pemrograman *Back-end*, routes-news-(news.routes.js) - Navigasi / URL Mapping Berita

```
//create like route.routes.js
import express from 'express';
import { verifyToken } from '../middlewares/Auth/authJwt.js';
import {
  createNews,
  deleteNews,
  getNews,
  getNewsById,
  updateNews,
} from '../controllers/news/news.controllers.js';

const router = express.Router();

// Get all news
router.get('/api/v1/news', getNews);

// Get news by id
router.get('/api/v1/news/:id', getNewsById);

// Create news
router.post('/api/v1/news', [verifyToken], createNews);

// Update news
router.put('/api/v1/news/:id', [verifyToken], updateNews);

// Delete news
router.delete('/api/v1/news/:id', [verifyToken], deleteNews);

export { router as newsRouter };
```

Lampiran 4. o Kode Pemrograman *Back-end*, routes-route control-(route.routes. js) - Navigasi / URL Mapping Rute

```
import express from 'express';
import { isAdmin, verifyToken } from '../middlewares/Auth/authJwt.js';
import {
  createRouteControllers,
  deleteRouteControllers,
  getActiveRoutes,
  getRouteByIdControllers,
  getRouteControllers,
  updateRouteControllers,
} from '../controllers/route-control/route.controllers.js';

const router = express.Router();
```



```

// Route for get all active routes
router.get('/api/v1/route/active', getActiveRoutes);

// Create route
router.post('/api/v1/route', [verifyToken, IsAdmin], createRouteControllers);

// Get List Route
router.get('/api/v1/route', getRouteControllers);

// Get Route by Id
router.get('/api/v1/route/:id', getRouteByIdControllers);

// Update Route
router.put('/api/v1/route/:id', [verifyToken, IsAdmin],
updateRouteControllers);

// Delete Route
router.delete(
  '/api/v1/route/:id',
  [verifyToken, IsAdmin],
  deleteRouteControllers
);

export { router as routeRoutes };

```

Lampiran 4. p Kode Pemrograman *Back-end*, routes-statistic-(statistic.routes.js) - Navigasi / URL Mapping Statistik

```

import express from 'express';
import {
  createStatistic,
  deleteStatistic,
  getHomeStatistic,
  updateStatistic,
} from '../controllers/statistic/statistic.controllers.js';
import { verifyToken } from '../middlewares/Auth/authJwt.js';

const router = express.Router();

// Update Statistic
router.put('/api/v1/statistic/:id', [verifyToken], updateStatistic);
// Delete Statistic
router.delete('/api/v1/statistic/:id', [verifyToken], deleteStatistic);

export { router as statisticRouter };

```

Lampiran 4. q Kode Pemrograman *Back-end*, routes-transaction-(transaction. routes.js) - Navigasi / URL Mapping Transaksi

```
import express from 'express';
import {
  createTransaction,
  deleteTransaction,
  detailTransactionControllers,
  getListTransaction,
  updateController,
  uploadAttachmentPayment,
} from '../../controllers/transaction/transaction.controllers.js';
import { isAdmin, verifyToken } from '../../middlewares/Auth/authJwt.js';

const router = express.Router();

// Create Transaction
router.post('/api/v1/transaction', [verifyToken], createTransaction);
//Get List Transaction
router.get('/api/v1/transaction', [verifyToken, isAdmin], getListTransaction);
//update transaction
router.put('/api/v1/transaction/:id', [verifyToken, isAdmin],
updateController);
//detail transaction
router.get(
  '/api/v1/transaction/:id',
  [verifyToken],
  detailTransactionControllers
);
//attachpayment with multer
router.post(
  '/api/v1/transaction/:id/attachment',
  [verifyToken],
  uploadAttachmentPayment
);
//delete transaction
router.delete('/api/v1/transaction/:id', [verifyToken], deleteTransaction);

export { router as transactionRouter };
```

Lampiran 4. r Kode Pemrograman *Back-end*, (index.js) – Entrypoint Utama

```
import cookieSession from "cookie-session";
import cors from "cors";
import dotenv from "dotenv";
import express from "express";
import mongoose from "mongoose";
```

```

import cron from "node-cron";
import bodyParser from "body-parser";
import compression from "compression";
import bcrypt from "bcryptjs";
import path from "path";
import { fileURLToPath } from "url";

//cluster
import cluster from "cluster";
import morgan from "morgan";
import os from "os";
// import rfs from "rotating-file-stream"; // version 2.x
import { authRoutes } from "../routes/Auth/auth.routes.js";
import roleModel from "../models/Auth/Roles.models.js";
import userModels from "../models/Auth/Users.models.js";
import { busRoutes } from "../routes/bus/bus.routes.js";
import { transactionRouter } from
"./routes/transaction/transaction.routes.js";
import { dashboardRouter } from "../routes/dashboard/dashboard.routes.js";
import { routeRoutes } from "../routes/route/route.routes.js";
import { statisticRouter } from "../routes/statistics/statistics.routes.js";
import { newsRouter } from "../routes/news/news.routes.js";

dotenv.config();
async function connectToDatabase() {
  try {
    const options = {
      // useNewUrlParser: true,
    };
    // Membuat koneksi mongoose
    mongoose
      .connect(process.env.MONGOURL, options)
      .then(() => {
        console.log("Koneksi MongoDB berhasil.");
      })
      .catch((error) => {
        console.error("Koneksi MongoDB gagal:", error);
      });
  } catch (error) {
    console.error("Koneksi MongoDB gagal:", error);
  }
}

// Panggil fungsi untuk menghubungkan ke database
await connectToDatabase();
const app = express();
app.use(
  compression({

```

```

    level: 6,
    threshold: 1 * 1000,
    //filter
    filter: (req, res) => {
      if (req.headers["x-no-compression"]) {
        return false;
      }
      return compression.filter(req, res);
    },
  },
)
);
// Initite Session
app.use(
  cookieSession({
    name: "session",
    secret: process.env.SECRET_KEY || "Secret Token", // should use as secret
environment variable
    httpOnly: true,
  })
);
.
.
.
// Simpan data user ke dalam database
  try {
    for (const userData of usersData) {
      const userRole = await roleModel.findOne({ name: userData.role });





      const newUser = new userModel({
        username: userData.username,
        name: userData.name,
        email: userData.email,
        password: bcrypt.hashSync(userData.password, 8),
        roles: userRole._id,
      });

      await newUser.save();
      console.log(
        `User ${userData.username} berhasil dibuat dengan role
${userData.role}.`
      )
    } catch (error) {
      console.error("Gagal menginisialisasi roles dan users:", error);
    }
  }
}
function startCron() {
  cron.schedule("*/10 * * * *", async () => {});
}
export default app;


```

Lampiran 5. Bukti Responden Pengujian System Usability Scale**Lampiran 5. a Bukti Responden Secara Tatap Muka**


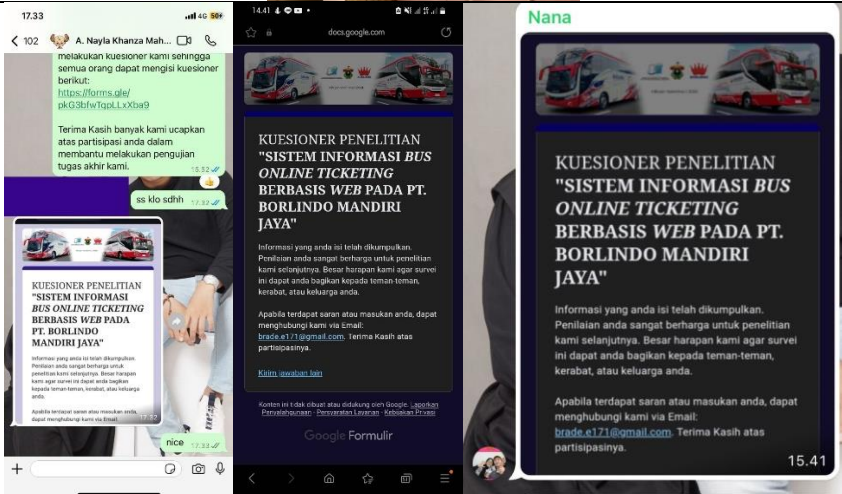
RESPONDEN	BUKTI DOKUMENTASI
Admin Borlindo	 
Security Borlindo	

RESPONDEN	BUKTI DOKUMENTASI
<p data-bbox="268 1104 411 1142"><i>User Trial</i></p>	   

RESPONDEN	BUKTI DOKUMENTASI
	  

RESPONDEN	BUKTI DOKUMENTASI
	

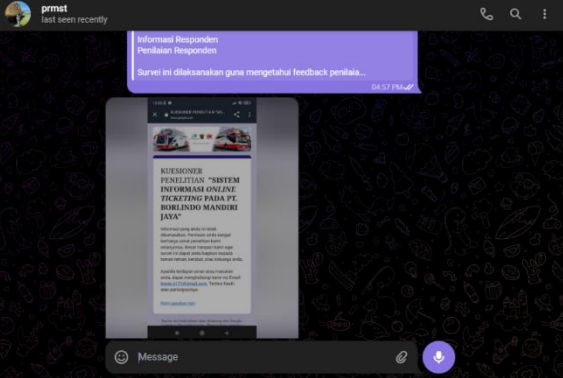
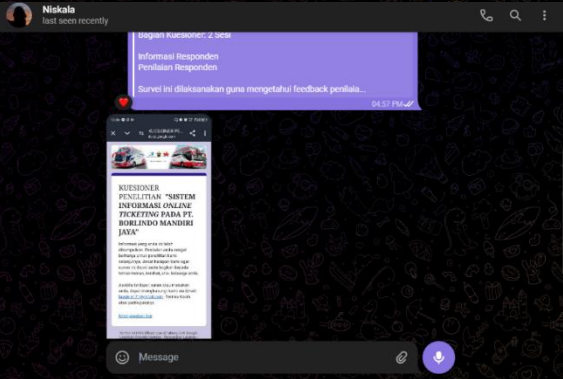
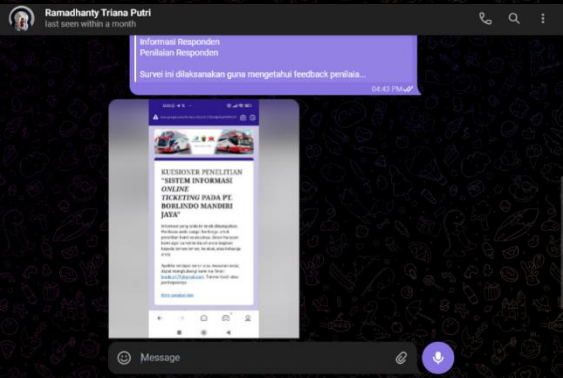
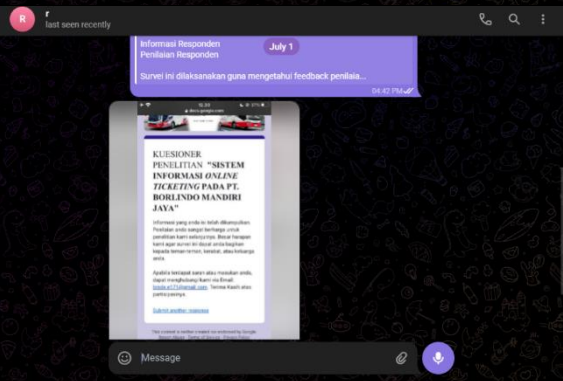
Lampiran 5. b Bukti Responden Via Sosial Media WhatsApp

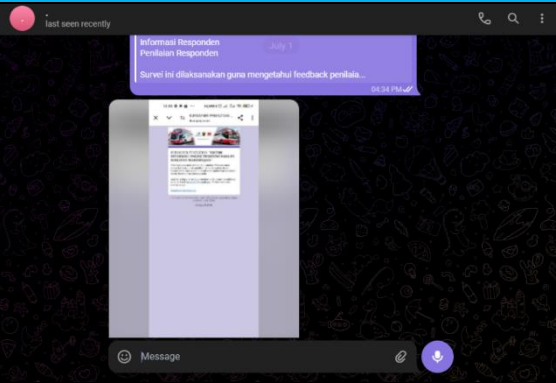
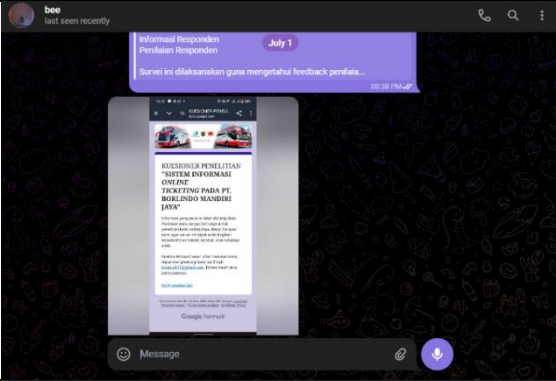
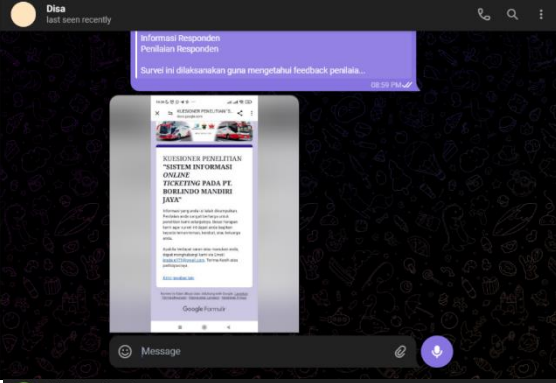
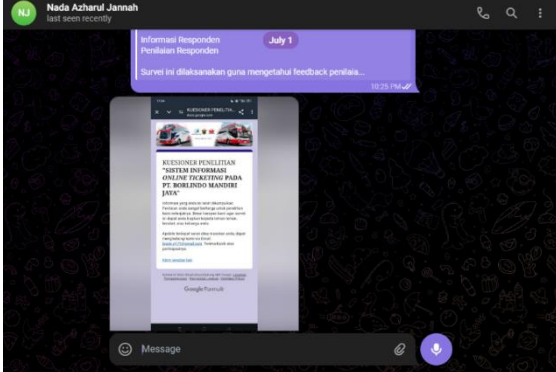
RESPONDEN	BUKTI DOKUMENTASI
	
	

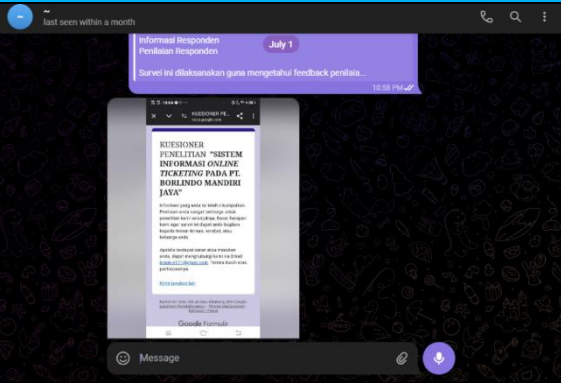
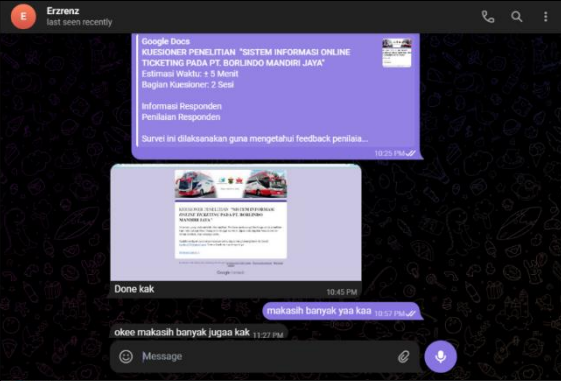
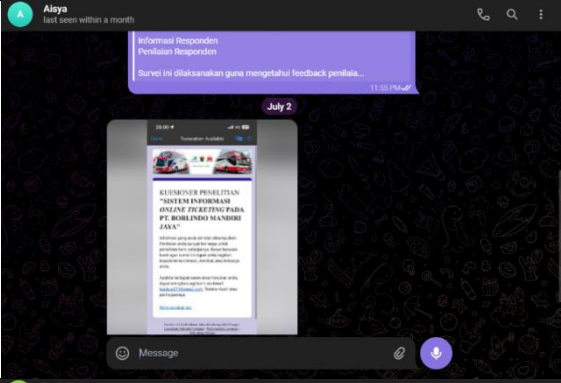
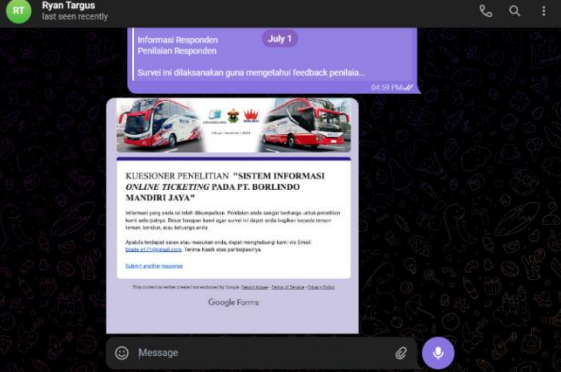
Lampiran 5. c Bukti Responden Via Sosial Media TikTok

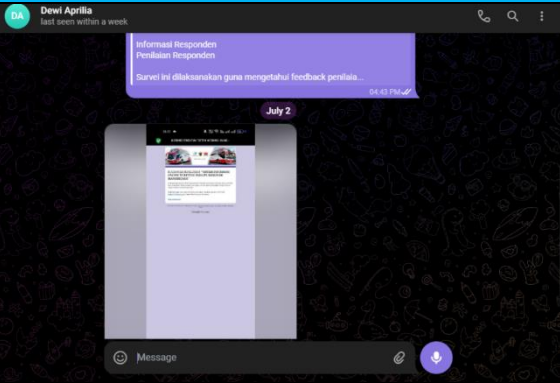
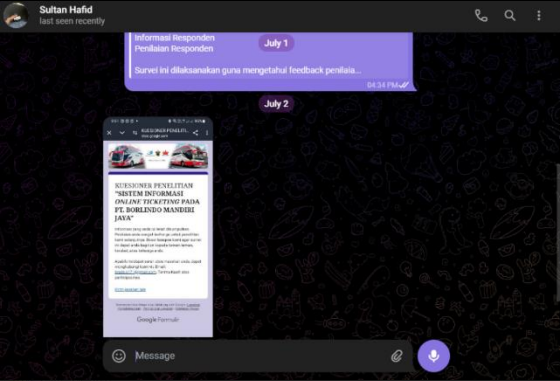
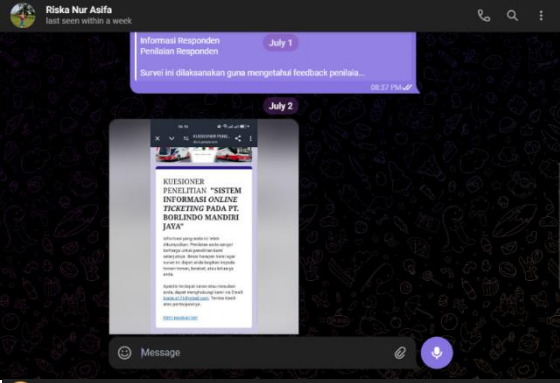
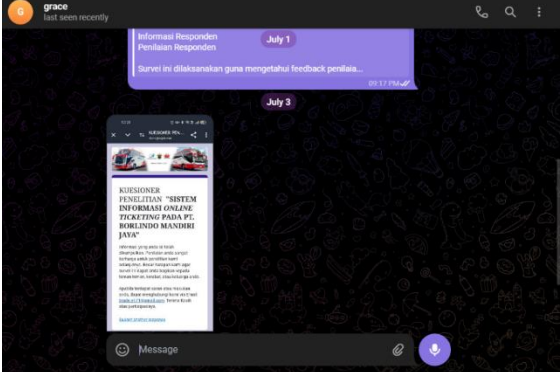
<p>USERNAME RESPONDEN</p>	<p>BUKTI DOKUMENTASI</p>
<p>@rinjaniip</p>	
<p>@sukakerudunghitam</p>	
<p>@hillomei</p>	

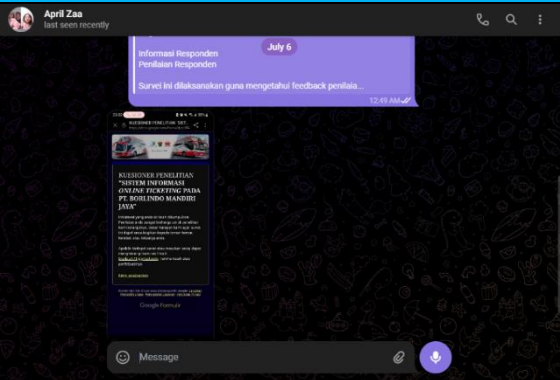
Lampiran 5. d Bukti Responden Via Sosial Media Telegram

<p>USERNAME RESPONDEN</p>	<p>BUKTI DOKUMENTASI</p>
<p>@prmst123</p>	
<p>@akuades</p>	
<p>@ramadhantytp</p>	
<p>@pencintaakucinggg</p>	

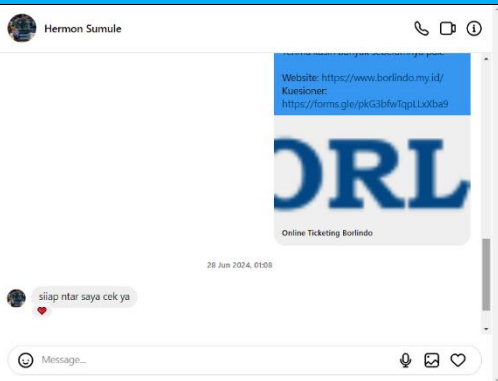
USERNAME RESPONDEN	BUKTI DOKUMENTASI
<p>@Rzk031</p>	
<p>@bowiebe</p>	
<p>@Disa_14</p>	
<p>Nada Azharul Jannah</p>	

<p>USERNAME RESPONDEN</p>	<p>BUKTI DOKUMENTASI</p>
<p>@eviprmtasari</p>	
<p>@erikadwarm</p>	
<p>Aisya</p>	
<p>@ryantargus</p>	

<p>USERNAME RESPONDEN</p>	<p>BUKTI DOKUMENTASI</p>
<p>@ApriliadewiN</p>	
<p>Sultan Hafid</p>	
<p>Riska Nur Asifa</p>	
<p>grace</p>	

<i>USERNAME RESPONDEN</i>	BUKTI DOKUMENTASI
April Zaa	

Lampiran 5. e Bukti Responden Via Sosial Media Instagram

<i>USERNAME RESPONDEN</i>	BUKTI DOKUMENTASI
@hermonsumule	

Lampiran 6. Hasil Pengujian Black-box

ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
HP-NBTB	(Home Page) Nav-Bar Tentang Borlindo	Menekan tombol “Tentang Borlindo”	Halaman Tentang Borlindo dapat dimuat dan menampilkan profil, informasi, visi misi, serta motto dari perusahaan PT. Borlindo Mandiri Jaya	Sistem masuk ke halaman Tentang Borlindo dan menampilkan profil, informasi, visi misi, serta motto dari perusahaan PT. Borlindo Mandiri Jaya	Sesuai dengan hasil yang diharapkan
HP-NBIP	(Home Page) Nav-Bar Informasi Perwakilan	Menekan tombol “Informasi Perwakilan”	Halaman Informasi Perwakilan dapat dimuat dan menampilkan kontak dan lokasi perwakilan di beberapa daerah	Sistem masuk ke halaman Informasi Perwakilan dan menampilkan kontak dan lokasi perwakilan di beberapa daerah	Sesuai dengan hasil yang diharapkan
HP-NBSI	(Home Page) Nav-Bar Masuk	Menekan tombol “Masuk”	Halaman Masuk (<i>Sign In</i>) akan dimuat dan terdapat kolom untuk mengisi <i>Username</i> dan <i>Password</i> , tombol <i>SignIn</i> dan tombol <i>register</i>	Sistem beralih ke halaman Masuk (<i>Sign In</i>) dan terdapat kolom untuk mengisi <i>Username</i> dan <i>Password</i> tombol <i>SignIn</i> dan tombol untuk <i>register</i>	Sesuai dengan hasil yang diharapkan
HP-NBSU	(Home Page) Nav-Bar Daftar	Menekan tombol “Daftar”	Halaman Daftar (<i>Sign Up</i>) akan dimuat dan terdapat kolom untuk mengisi Nama, <i>Username</i> , <i>Password</i> , dan <i>Email</i> serta tombol <i>SignUp</i> dan tombol untuk Login	Sistem beralih ke halaman Daftar (<i>Sign Up</i>) dan terdapat kolom untuk mengisi Nama, <i>Username</i> , <i>Password</i> , dan <i>Email</i> serta tombol <i>SignUp</i> dan tombol untuk Login	Sesuai dengan hasil yang diharapkan
HP-NP	(Home Page) News Page	Menekan tombol “Baca Selengkapnya” pada laman Berita Terbaru	Tab baru pada <i>search engine</i> akan menampilkan berita lengkap sesuai link yang telah diinput oleh Admin	<i>Search engine</i> membuka tab baru dan menampilkan berita lengkap sesuai link yang telah diinput oleh Admin	Sesuai dengan hasil yang diharapkan
HP-NBNP	(Home Page) Nav-Bar Nama Profil	Menekan tombol “BRYAN ADMIN” saat setelah <i>Login Admin role</i>	Akan muncul pop up dengan pilihan <i>Dashboard</i> atau Keluar	Sistem memunculkan pop up dengan pilihan <i>Dashboard</i> atau Keluar	Sesuai dengan hasil yang diharapkan

ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
HP-NBDB	(Home Page) Nav-Bar Dashboard	Menekan tombol "Dashboard"	Admin page akan dimuat dan menampilkan beberapa <i>control bar</i> yang bisa diatur oleh admin. Untuk tampilan pada <i>role user</i> hanya menampilkan <i>history</i> dan pemesanan saja	Sistem mengalihkan ke <i>admin page</i> dan menampilkan beberapa <i>control bar</i> yang bisa diatur oleh admin. Untuk tampilan pada <i>role user</i> hanya menampilkan <i>history</i> dan pemesanan saja	Sesuai dengan hasil yang diharapkan
HP-NBLO	(Home Page) Nav-Bar Keluar	Menekan tombol "Keluar"	Menu utama dimuat dan tidak dapat melakukan pemesanan dan melihat <i>dashboard</i>	Sistem akan kembali ke menu utama dan tidak dapat melakukan pemesanan dan melihat <i>dashboard</i>	Sesuai dengan hasil yang diharapkan

ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
IPP-SB01	(Informasi Perwakilan Page) Search Bar- Rute Tersedia	Kolom Cari Perwakilan diisi dengan "Palopo"	Sistem akan menampilkan informasi mengenai rute yang diisikan	Sistem menampilkan informasi kontak dan lokasi dari rute Palopo	Sesuai dengan hasil yang diharapkan
IPP-SB02	(Informasi Perwakilan Page) Search Bar-Rute Tidak Tersedia	Kolom Cari Perwakilan diisi dengan "Papolo"	Sistem tidak akan menampilkan informasi mengenai rute yang diisikan	Sistem menampilkan pop up berisikan pesan "Maaf, kata kunci yang anda masukkan tidak terdapat di list"	Sesuai dengan hasil yang diharapkan
ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
SUP-01	(Sign Up Page) Tanpa Diisi	Tidak mengisi data dan langsung menekan tombol "SignUp"	Sistem akan memunculkan pop up dengan pesan "Please fill out this field"	Sistem tidak dapat lanjut dan memunculkan pop up dengan pesan "Please fill out this field"	Sesuai dengan hasil yang diharapkan
SUP-02	(Sign Up Page) Isi Email tanpa @	Tidak menambahkan @ pada kolom email dan menekan tombol "SignUp"	Sistem akan memunculkan pop up dengan pesan "Please include an '@' in the email address. 'you're email' is missing an '@'"	Sistem tidak dapat lanjut dan memunculkan pop up dengan pesan "Please include an '@' in the email address. 'you're email' is missing an '@'"	Sesuai dengan hasil yang diharapkan

ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
SUP-03	(Sign Up Page) Diisi Lengkap	Mengisi data dengan benar dan menekan tombol "SignUp"	Sistem akan lanjut dan kita dapat melihat nama profil melakukan pemesanan dan masuk ke menu <i>dashboard</i>	Sistem memuat halaman utama dengan Nama Profil, dapat melakukan pemesanan serta masuk ke menu <i>dashboard</i>	Sesuai dengan hasil yang diharapkan
SUP-04	(Sign Up Page) Tombol Login	Menekan tombol "Allready have account? Login"	Sistem akan memuat halaman masuk (Login/SignIn)	Halaman teralihkan dari halaman daftar (<i>signup</i>) menjadi halaman masuk (<i>signin</i>)	Sesuai dengan hasil yang diharapkan
ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
SIP-01	(Sign In Page) Tanpa Diisi	Tidak mengisi data dan langsung menekan tombol "SignIn"	Sistem akan memunculkan pop up dengan pesan "Please fill out this field"	Sistem tidak dapat lanjut dan memunculkan pop up dengan pesan "Please fill out this field"	Sesuai dengan hasil yang diharapkan
SIP-02	(Sign In Page) Isi Data tidak sesuai	Mengisi data yang tidak sesuai dan langsung menekan tombol "SignIn"	Sistem akan memunculkan pop up dengan pesan "Data not found. Try again"	Sistem tidak dapat lanjut dan memunculkan pop up dengan pesan "Data not found. Try again"	Sesuai dengan hasil yang diharapkan
SIP-03	(Sign In Page) Isi Data Benar	Mengisi data dengan benar dan menekan tombol "SignIn"	Sistem akan lanjut dan kita dapat melihat nama profil melakukan pemesanan dan masuk ke menu <i>dashboard</i>	Sistem memuat halaman utama dengan Nama Profil, dapat melakukan pemesanan serta masuk ke menu <i>dashboard</i>	Sesuai dengan hasil yang diharapkan
SIP-04	(Sign In Page) Tombol Register	Menekan tombol "Dont have an account? Register"	Sistem akan memuat halaman daftar (Register/SignUp)	Halaman teralihkan dari halaman masuk (<i>signin</i>) menjadi halaman daftar (<i>signup</i>)	Sesuai dengan hasil yang diharapkan
ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
HP-RTPR	(Home Page) Reservasi Tiket - Pilih Rute & Tanggal	Tanpa mengisi data dan menekan tombol "Cari"	Sistem tidak dapat dilanjutkan	Tombol lanjut menjadi <i>disable</i> dan proses pemesanan tidak dapat dilanjutkan	Sesuai dengan hasil yang diharapkan

ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
HP-RTPP	(Home Page) Reservasi Tiket – Pulang Pergi	Menekan tombol "Pulang Pergi"	Sistem akan memunculkan pilihan jadwal untuk pulang pergi	Tombol jadwal menjadi <i>available</i> dan proses pemesanan tiket pulang dapat dilanjutkan	Sesuai dengan hasil yang diharapkan
HP-RTPB	(Home Page) Reservasi Tiket - Pilih Bus	Menekan tombol "Pilih"	Sistem akan lanjut ke halaman pilih kursi	Sistem lanjut ke halaman pilih kursi	Sesuai dengan hasil yang diharapkan
HP-RTPS1	(Home Page) Reservasi Tiket - Pilih Seat Test 1	Tidak menentukan pilihan "Seat" dan menekan tombol "Lanjut"	Sistem tidak dapat dilanjutkan	Tombol lanjut menjadi <i>disable</i> dan proses pemesanan tidak dapat dilanjutkan	Sesuai dengan hasil yang diharapkan
HP-RTPS2	(Home Page) Reservasi Tiket - Pilih Seat Test 2	Menentukan pilihan "Seat" dan menekan tombol "Lanjut"	Sistem akan lanjut ke halaman pengisian data pemesanan	Tombol lanjut menjadi <i>available</i> dan proses pemesanan dapat dilanjutkan ke halaman pengisian data pemesanan	Sesuai dengan hasil yang diharapkan
HP-RTDP1	(Home Page) Reservasi Tiket - Isi Data Pemesanan Test 1	Tanpa mengisi data dan menekan tombol "Lanjut"	Sistem tidak dapat dilanjutkan	Tombol lanjut menjadi <i>disable</i> dan proses pemesanan tidak dapat dilanjutkan	Sesuai dengan hasil yang diharapkan
HP-RTDP2	(Home Page) Reservasi Tiket - Isi Data Pemesanan Test 2	Mengisi data lengkap dan menekan tombol "Lanjut"	Sistem akan lanjut ke halaman pembayaran	Tombol lanjut menjadi <i>available</i> dan proses pemesanan dapat dilanjutkan ke halaman pembayaran	Sesuai dengan hasil yang diharapkan
HP-RTP1	(Home Page) Reservasi Tiket - Pembayaran Test 1	Tidak memilih bukti pembayaran dan menekan tombol "Lanjut"	Sistem tidak dapat dilanjutkan	Tombol lanjut menjadi <i>disable</i> dan proses pemesanan tidak dapat dilanjutkan	Sesuai dengan hasil yang diharapkan
HP-RTP2	(Home Page) Reservasi Tiket - Pembayaran Test 2	Tidak mengunggah bukti pembayaran dan menekan tombol "Lanjut"	Sistem tidak dapat dilanjutkan	Tombol lanjut menjadi <i>disable</i> dan proses pemesanan tidak dapat dilanjutkan	Sesuai dengan hasil yang diharapkan

ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
HP-RTP3	(Home Page) Reservasi Tiket - Pembayaran Test 3	Telah memilih dan mengunggah bukti pembayaran dan menekan tombol "Lanjut"	Sistem akan lanjut ke halaman menunggu verifikasi pembayaran	Tombol lanjut menjadi <i>available</i> dan proses pemesanan dapat dilanjutkan ke halaman menunggu verifikasi pembayaran	Sesuai dengan hasil yang diharapkan
HP-RTVP	(Home Page) Reservasi Tiket - Verifikasi Pembayaran	Menekan tombol "Lanjut"	Halaman <i>dashboard</i> dimuat untuk melihat ID Transaksi dan <i>e-ticket</i> setelah pemesanan disetujui oleh admin	Sistem akan <i>direct</i> ke halaman dashboard dan dapat melihat ID transaksi dan e-ticket setelah pemesanan disetujui oleh admin	Sesuai dengan hasil yang diharapkan
ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
AP-I	(Admin Page) Informasi	Menekan tombol "Informasi"	Halaman informasi dimuat dan menampilkan informasi pemesanan, bus tersedia, dan histori transaksi	Sistem menampilkan informasi pemesanan, bus tersedia, dan histori transaksi	Sesuai dengan hasil yang diharapkan
AP-IA1	(Admin Page) Informasi - Action Test 1	Menekan tombol "Open Etiket"	Sistem akan memunculkan pop up berisi ID transaksi, data penumpang, syarat perjalanan, dan informasi kantor pusat	Sistem memunculkan pop up <i>E-Ticket</i> berisi ID transaksi, data penumpang, syarat perjalanan, dan informasi kantor pusat	Sesuai dengan hasil yang diharapkan
AP-IA2	(Admin Page) Informasi - Action Test 2	Menekan tombol "Close"	Sistem akan kembali ke halaman informasi	Sistem kembali ke halaman informasi	Sesuai dengan hasil yang diharapkan
ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
AP-P	(Admin Page) Pemesanan	Menekan tombol "Pemesanan"	Halaman utama dimuat agar dapat melakukan pemesanan tiket	Sistem akan mengalihkan tampilan ke halaman utama agar dapat melakukan pemesanan tiket	Sesuai dengan hasil yang diharapkan
ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
AP-KB	(Admin Page) Kontrol Bus	Menekan tombol "Bus Kontrol"	Halaman Bus Management dimuat dan menampilkan daftar list bus yang telah diatur oleh admin dan	Sistem masuk ke halaman bus management dan menampilkan daftar list bus yang telah diatur oleh admin dan terdapat pilihan	Sesuai dengan hasil yang diharapkan

ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
			terdapat pilihan tambah atau hapus bus dan atur harga kursi bus	tambah atau hapus bus dan atur harga kursi bus	
AP-KBTB 1	(Admin Page) Kontrol Bus - Tambah Bus Test 1	Menekan tombol "Tambah Bus" dan melanjutkan tanpa mengisi data	Halaman dalam posisi tetap dan memunculkan pop up berisi pesan "failed to create bus"	Sistem tidak lanjut dan muncul pop up berisi pesan "failed to create bus"	Sesuai dengan hasil yang diharapkan
AP-KBTB 2	(Admin Page) Kontrol Bus - Tambah Bus Test 2	Menekan tombol "Tambah Bus" dan melanjutkan dengan mengisi data lengkap	Sistem akan kembali ke halaman Bus Management dan bus yang baru ditambahkan akan muncul	Sistem berhasil lanjut dan kembali ke halaman Bus Management dan bus yang baru ditambahkan telah muncul	Sesuai dengan hasil yang diharapkan
AP-KBTB 3	(Admin Page) Kontrol Bus - Tambah Bus Test 3	Menekan pilihan "Kelas Bus"	Sistem akan memunculkan 3 pilihan kelas bus, yaitu tronton, reguler, dan sleeper	Sistem memunculkan 3 pilihan kelas bus, yaitu tronton, reguler, dan sleeper	Sesuai dengan hasil yang diharapkan
AP-KBTB 4	(Admin Page) Kontrol Bus - Tambah Bus Test 4	Menekan pilihan "Keberangkatan"	Sistem akan memunculkan beberapa pilihan rute yang telah di atur di halaman kontrol rute	Sistem memunculkan beberapa pilihan rute yang telah di atur di halaman kontrol rute	Sesuai dengan hasil yang diharapkan
AP-KBTB 5	(Admin Page) Kontrol Bus - Tambah Bus Test 5	Menekan pilihan "Tujuan"	Sistem akan memunculkan beberapa pilihan rute yang telah di atur di halaman kontrol rute	Sistem memunculkan beberapa pilihan rute yang telah di atur di halaman kontrol rute	Sesuai dengan hasil yang diharapkan
AP-KBTAK 1	(Admin Page) Kontrol Bus - Atur Kursi Test 1	Menekan tombol "Atur Kursi" dan melanjutkan tanpa mengisi data	Sistem tidak dapat dilanjutkan	Tombol lanjut menjadi <i>disable</i> dan proses tidak dapat dilanjutkan	Sesuai dengan hasil yang diharapkan
AP-KBTAK 2	(Admin Page) Kontrol Bus - Atur Kursi Test 2	Menekan tombol "Atur Kursi" dan melanjutkan	Kursi akan memiliki harga masing-masing saat <i>user</i> melakukan reservasi tiket	Kursi menjadi warna merah yang artinya harga telah diatur dan dapat digunakan saat <i>user</i> melakukan reservasi tiket	Sesuai dengan hasil yang diharapkan

ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
		dengan mengisi data lengkap			
AP-KBAK 3	(Admin Page) Kontrol Bus - Atur Kursi Test 3	Menekan pilihan "Kelas Kursi"	Sistem akan memunculkan 3 pilihan kelas kursi, yaitu sleeper, VIP, dan executive	Sistem memunculkan 3 pilihan kelas kursi, yaitu sleeper, VIP, dan executive	Sesuai dengan hasil yang diharapkan
AP-KBHB 9	(Admin Page) Kontrol Bus - Hapus Bus	Menekan tombol "Hapus"	Daftar bus yang dihapus akan hilang dari sistem dan tidak akan muncul di menu reservasi tiket	Daftar bus yang dihapus menjadi hilang dari sistem dan tidak muncul di menu reservasi tiket	Sesuai dengan hasil yang diharapkan
ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
AP-KT	(Admin Page) Kontrol Transaksi	Menekan tombol "Kontrol Transaksi"	Halaman Transaction Management dimuat dan menampilkan daftar list transaksi yang telah diatur oleh admin dan terdapat pilihan Cetak dan Status Transaksi	Sistem akan masuk ke halaman Transaction Management dan menampilkan daftar list transaksi yang telah diatur oleh admin dan terdapat pilihan Cetak dan Status Transaksi	Sesuai dengan hasil yang diharapkan
AP-KTC1	(Admin Page) Kontrol Transaksi - Cetak Test 1	Tidak memilih tanggal dan Menekan tombol "Cetak"	Rekap Transaksi akan dicetak semua yang terdapat pada list transaksi	Sistem melakukan pengunduhan pada semua rekap transaksi yang terdapat pada list transaksi	Sesuai dengan hasil yang diharapkan
AP-KTC2	(Admin Page) Kontrol Transaksi - Cetak Test 2	Memilih tanggal 1 Juni - 1 Juli dan Menekan tombol "Cetak"	Rekap Transaksi mulai tanggal 1 Juni hingga 1 Juli akan dicetak	Sistem melakukan pengunduhan pada Rekap Transaksi mulai tanggal 1 Juni hingga 1 Juli	Sesuai dengan hasil yang diharapkan
AP-KTST1	(Admin Page) Kontrol Transaksi - Status Transaction Test 1	Menekan pilihan "Status Transaksi"	Sistem akan memunculkan 3 pilihan status, yaitu pending, cancelled, dan success	Sistem memunculkan 3 pilihan status, yaitu pending, cancelled, dan success	Sesuai dengan hasil yang diharapkan
AP-KTST2	(Admin Page) Kontrol Transaksi - Status	Menekan Tombol "Success"	Pemesanan berhasil dan Sistem akan memunculkan E-Ticket di sisi pemesan tiket	Sistem memunculkan E-Ticket di sisi pemesan tiket	Sesuai dengan hasil yang diharapkan

ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
	Transaction Test 2				
AP-KTST3	(Admin Page) Kontrol Transaksi - Status Transaction Test 3	Menekan Tombol "Cancelled"	Pemesanan gagal dan Sistem akan tidak memunculkan E-Ticket di sisi pemesan tiket	Sistem tidak memunculkan E-Ticket di sisi pemesan tiket	Sesuai dengan hasil yang diharapkan
AP-KTA1	(Admin Page) Kontrol Transaksi - Action Test 1	Menekan pilihan "Action"	Sistem akan memunculkan 3 pilihan action, yaitu detail dan hapus	Sistem memunculkan 3 pilihan action, yaitu detail dan hapus	Sesuai dengan hasil yang diharapkan
AP-KTA2	(Admin Page) Kontrol Transaksi - Action Test 2	Menekan Tombol "Hapus"	Daftar transaksi yang dihapus akan hilang dari transaction management	Daftar transaksi yang dihapus hilang dari transaction management	Sesuai dengan hasil yang diharapkan
AP-KTA3	(Admin Page) Kontrol Transaksi - Action Test 3	Menekan Tombol "Detail"	Sistem akan menampilkan data pemesanan <i>user</i>	Sistem menampilkan data pemesanan <i>user</i>	Sesuai dengan hasil yang diharapkan
ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
AP-KR	(Admin Page) Kontrol Rute	Menekan tombol "Kontrol Rute"	Halaman Kelola Rute dimuat dan menampilkan daftar list rute yang telah diatur oleh admin dan terdapat pilihan tambah, edit atau hapus rute	Sistem masuk ke Halaman Kelola Rute dan menampilkan daftar list rute yang telah diatur oleh admin dan terdapat pilihan tambah, edit atau hapus rute	Sesuai dengan hasil yang diharapkan
AP-KRTR1	(Admin Page) Kontrol Rute - Tambah Rute Test 1	Menekan tombol "Tambah Rute" dilanjutkan tidak mengisi data dan menekan tombol "Submit"	Halaman dalam posisi tetap dan memunculkan pop up berisi pesan "Failed to create route"	Sistem tidak lanjut dan muncul pop up berisi pesan "Failed to create route"	Sesuai dengan hasil yang diharapkan
AP-KRTR2	(Admin Page) Kontrol Rute - Tambah Rute Test 2	Menekan tombol "Tambah Rute" dilanjutkan mengisi data dan	Sistem akan kembali ke halaman Kelola Rute dan rute yang baru ditambahkan akan muncul	Sistem berhasil lanjut dan kembali ke halaman Kelola Rute dan rute yang baru ditambahkan telah muncul	Sesuai dengan hasil yang diharapkan

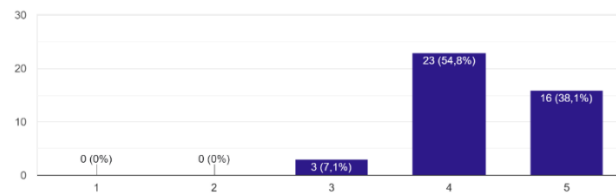
ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
		menekan tombol "Submit"			
AP-KRTR3	(Admin Page) Kontrol Rute - Tambah Rute Test 3	Menekan pilihan "Status"	Sistem akan memunculkan 2 pilihan status, yaitu aktif (kanan) dan tidak aktif (kiri)	Sistem akan memunculkan 2 pilihan status, yaitu aktif (kanan) dan tidak aktif (kiri)	Sesuai dengan hasil yang diharapkan
AP-KRA1	(Admin Page) Kontrol Rute - Action Test 1	Menekan Tombol "Edit"	Sistem akan memunculkan kembali data rute dan terdapat pilihan untuk aktifkan atau tidak aktifkan rute	Sistem memunculkan kembali data rute dan terdapat pilihan untuk aktifkan atau tidak aktifkan rute	Sesuai dengan hasil yang diharapkan
AP-KRA2	(Admin Page) Kontrol Rute - Action Test 2	Menekan Tombol "Hapus"	Daftar Rute yang dihapus akan hilang dari Rute Kontrol	Daftar Rute yang dihapus hilang dari Rute Kontrol	Sesuai dengan hasil yang diharapkan
ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
AP-KBT	(Admin Page) Kontrol Berita	Menekan tombol "Kontrol Berita"	Halaman Kelola Berita dimuat dan menampilkan daftar list Berita yang telah diatur oleh admin dan terdapat pilihan tambah, edit atau hapus Berita	Sistem masuk ke Halaman Kelola Berita dan menampilkan daftar list Berita yang telah diatur oleh admin dan terdapat pilihan tambah, edit atau hapus Berita	Sesuai dengan hasil yang diharapkan
AP-KBTTB1	(Admin Page) Kontrol Berita - Tambah Berita Test 1	Menekan tombol "Tambah Berita" dan melanjutkan tanpa mengisi data	Halaman dalam posisi tetap dan memunculkan pop up berisi pesan "Failed add news"	Sistem tidak lanjut dan muncul pop up berisi pesan "Failed add news"	Sesuai dengan hasil yang diharapkan
AP-KBTTB2	(Admin Page) Kontrol Berita - Tambah Berita Test 2	Menekan tombol "Tambah Berita" dan melanjutkan dengan mengisi data lengkap	Sistem akan kembali ke halaman Kelola Berita dan Berita yang baru ditambahkan akan muncul	Sistem berhasil lanjut dan kembali ke halaman Kelola Berita dan Berita yang baru ditambahkan telah muncul	Sesuai dengan hasil yang diharapkan
AP-KBA1	(Admin Page) Kontrol Berita - Action Test 1	Menekan Tombol "Edit"	Sistem akan memunculkan kembali data Berita dan admin dapat melakukan perubahan data	Sistem memunculkan kembali data Berita dan admin dapat melakukan perubahan data	Sesuai dengan hasil yang diharapkan

ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
AP-KBA2	(Admin Page) Kontrol Berita - Action Test 2	Menekan Tombol "Hapus"	Daftar Berita yang dihapus akan hilang dari Berita Kontrol	Daftar Berita yang dihapus hilang dari Berita Kontrol	Sesuai dengan hasil yang diharapkan
ID	Deskripsi Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
AP-KS	(Admin Page) Kontrol Statistik	Menekan tombol "Kontrol Statistik"	Halaman Kelola Statistik dimuat dan menampilkan daftar list Statistik yang telah diatur oleh admin dan terdapat pilihan tambah, edit atau hapus Statistik	Sistem masuk ke Halaman Kelola Statistik dan menampilkan daftar list Statistik yang telah diatur oleh admin dan terdapat pilihan tambah, edit atau hapus Statistik	Sesuai dengan hasil yang diharapkan
AP-KSTS1	(Admin Page) Kontrol Rute - Tambah Statistik Test 1	Menekan tombol "Tambah Statistik" dan melanjutkan tanpa mengisi data	Halaman dalam posisi tetap dan memunculkan pop up berisi pesan "Failed add statistic"	Sistem tidak lanjut dan muncul pop up berisi pesan "Failed add statistic"	Sesuai dengan hasil yang diharapkan
AP-KSTS2	(Admin Page) Kontrol Rute - Tambah Statistik Test 2	Menekan tombol "Tambah Statistik" dan melanjutkan dengan mengisi data lengkap	Sistem akan kembali ke halaman Kelola Statistik dan Statistik yang baru ditambahkan akan muncul	Sistem berhasil lanjut dan kembali ke halaman Kelola Statistik dan Statistik yang baru ditambahkan telah muncul	Sesuai dengan hasil yang diharapkan
AP-KSA1	(Admin Page) Kontrol Rute - Action Test 1	Menekan Tombol "Edit"	Sistem akan memunculkan kembali data Statistik dan admin dapat melakukan perubahan data	Sistem memunculkan kembali data Statistik dan admin dapat melakukan perubahan data	Sesuai dengan hasil yang diharapkan
AP-KSA2	(Admin Page) Kontrol Rute - Action Test 2	Menekan Tombol "Hapus"	Daftar Statistik yang dihapus akan hilang dari Statistik Kontrol	Daftar Statistik yang dihapus hilang dari Statistik Kontrol	Sesuai dengan hasil yang diharapkan

Lampiran 7. Persentase Nilai Setiap Pertanyaan SUS

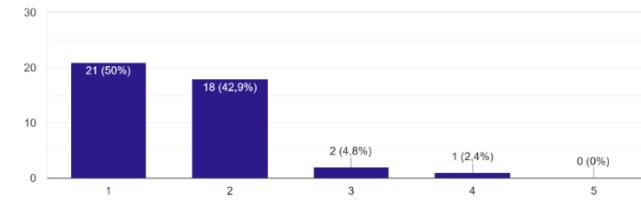
1. Saya berpikir akan sering menggunakan website seperti <https://www.borlindo.my.id/> untuk memesan tiket

42 jawaban



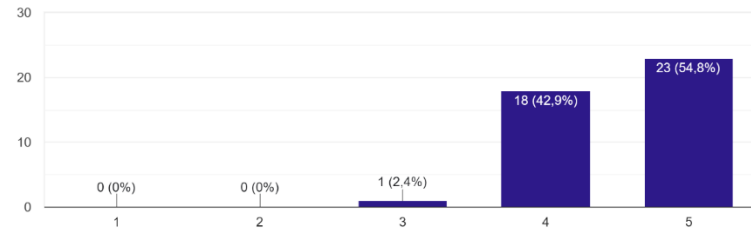
2. Saya merasa website <https://www.borlindo.my.id/> ini rumit untuk digunakan

42 jawaban



3. Saya merasa website <https://www.borlindo.my.id/> ini mudah digunakan

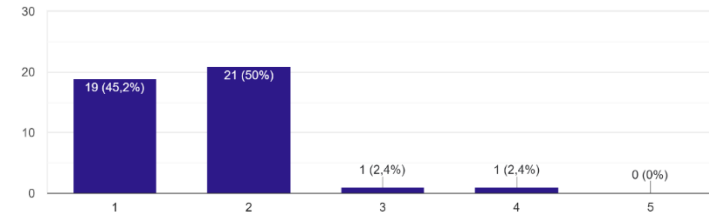
42 jawaban



4. Saya membutuhkan bantuan orang lain untuk mengakses website

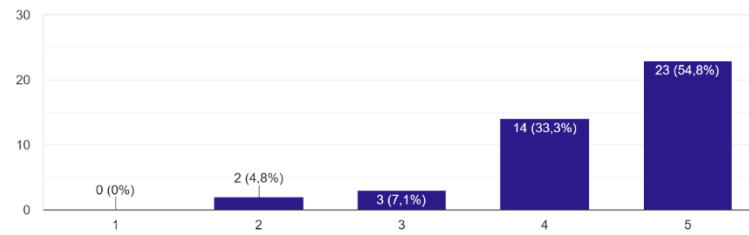
<https://www.borlindo.my.id/> ini

42 jawaban



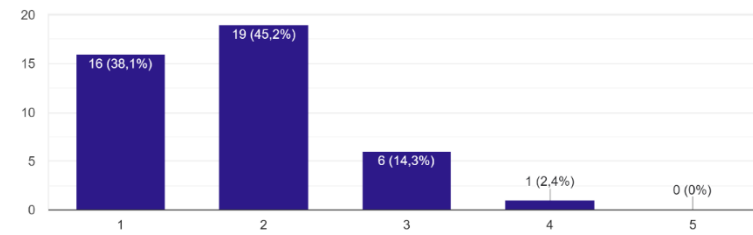
5. Saya merasa semua fitur website <https://www.borlindo.my.id/> ini berjalan dengan semestinya

42 jawaban

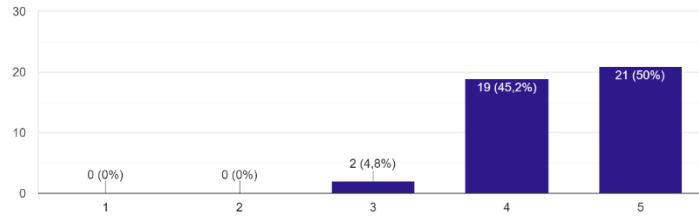


6. Saya merasa ada banyak hal yang tidak serasi pada website <https://www.borlindo.my.id/> ini

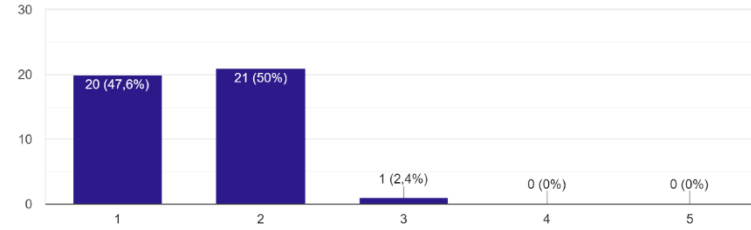
42 jawaban



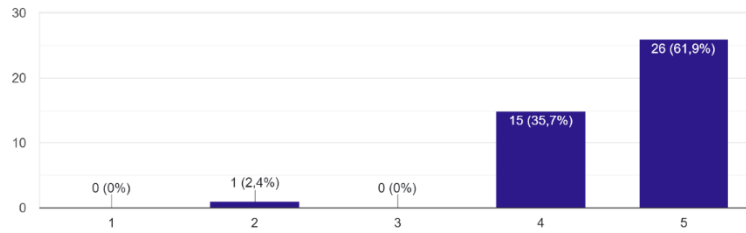
7. Saya merasa orang lain juga dapat memahami website <https://www.borlindo.my.id/> ini dengan cepat
42 jawaban



8. Saya merasa website <https://www.borlindo.my.id/> ini membingungkan
42 jawaban



9. Saya merasa tidak terdapat hambatan dalam mengakses website <https://www.borlindo.my.id/> ini
42 jawaban



10. Saya perlu banyak belajar terlebih dahulu sebelum mengakses website <https://www.borlindo.my.id/> ini
42 jawaban

