

## **BAB III**

### **PELAKSANAAN KERJA MAGANG**

#### **3.1 Kedudukan dan Koordinasi**

Gugus Mitigasi Lebak Selatan (GMLS) membuka peluang program magang Track 2 bagi mahasiswa Universitas Multimedia Nusantara (UMN) semester tujuh. Program ini dirancang untuk memberikan mahasiswa kesempatan menerapkan teori yang telah dipelajari dalam situasi kerja nyata, terutama dalam konteks mitigasi bencana. Dengan pendekatan berbasis pengalaman langsung, program ini membantu mahasiswa memahami bagaimana ilmu pengetahuan dan teknologi dapat menyelesaikan masalah di lapangan.

Tugas utama mahasiswa dalam program ini adalah analisis data mitigasi bencana dan pengembangan dashboard berbasis web. Analisis data bertujuan membantu GMLS memahami pola dan tren terkait mitigasi bencana, seperti gempa bumi, tsunami, atau bencana lain di wilayah Lebak Selatan. Hasil analisis menjadi dasar perencanaan dan pengambilan keputusan berbasis data. Selain itu, pengembangan dashboard web memungkinkan mahasiswa menciptakan platform digital untuk komunitas dan kolaborator GMLS. Dashboard ini dirancang menyajikan informasi terkini tentang risiko bencana, rencana evakuasi, dan panduan darurat. Dengan mendorong partisipasi masyarakat, platform ini diharapkan memperkuat kesiapan komunitas terhadap bencana.

Mahasiswa magang mendapatkan bimbingan dari mentor berpengalaman untuk memastikan pekerjaan memenuhi standar profesional. Setiap peserta menerima arahan teknis, evaluasi hasil kerja, dan dukungan dalam mengatasi tantangan selama pelaksanaan tugas. Selain penguasaan teknis, mahasiswa juga mengembangkan keterampilan seperti komunikasi, analisis data, dan manajemen proyek. Program ini menjadi pengalaman berharga bagi peserta untuk memahami

penerapan teori di dunia nyata, menciptakan solusi berbasis teknologi yang bermanfaat bagi masyarakat.

Program magang ini menghubungkan dunia akademik dengan kebutuhan lapangan, membuat pengalaman belajar lebih kontekstual. Mahasiswa memperoleh pemahaman lebih baik tentang mitigasi bencana sekaligus membangun jaringan profesional yang dapat membantu karier mereka di masa depan. Dengan keterlibatan langsung dalam proyek berbasis teknologi, peserta tidak hanya memperkuat sistem informasi GMLS, tetapi juga mendukung upaya sosial melalui inovasi praktis.

Selain pengembangan teknis, peserta magang juga bertanggung jawab memastikan data pada platform GMLS selalu akurat dan relevan. Melalui pengelolaan dashboard mitigasi bencana, mereka menyajikan informasi penting seperti peta risiko, titik evakuasi, dan panduan tanggap darurat secara real-time. Mahasiswa juga berperan dalam pengelolaan media sosial untuk meningkatkan keterlibatan masyarakat. Konten berbasis data yang mereka hasilkan bertujuan menjangkau audiens yang lebih luas dan mendorong partisipasi aktif dalam upaya mitigasi.

Bimbingan dari anggota tim GMLS memberikan wawasan mendalam tentang standar kerja profesional. Peserta mendapatkan pemahaman tentang proses kerja, cara menangani tantangan teknis, serta adaptasi terhadap perubahan industri. Interaksi langsung dengan profesional GMLS dan mitra komunitas Humanity Project memperkaya pengalaman mereka dalam kerja tim dan pengembangan solusi bersama.

Program ini juga berkontribusi pada pengembangan kemampuan strategis mahasiswa, seperti perencanaan komunikasi dan manajemen proyek. Dengan mengintegrasikan teknologi ke aspek operasional GMLS, mahasiswa membantu menciptakan pendekatan berbasis data yang lebih efisien. Pengalaman ini tidak hanya memperkuat sistem informasi organisasi tetapi juga memberikan landasan kokoh bagi pengembangan karier mereka.

Kesempatan magang ini membuka peluang membangun jejaring profesional di bidang mitigasi bencana dan pengelolaan masyarakat. Relasi yang terbentuk sering kali menjadi aset penting untuk kemajuan karier mahasiswa. Setelah menyelesaikan program, peserta lebih percaya diri menghadapi tantangan dunia kerja dan mampu memberikan kontribusi signifikan di tempat kerja.

Melalui program magang ini, GMLS dan UMN berharap menciptakan generasi muda yang tidak hanya cerdas secara intelektual tetapi juga mampu membawa perubahan sosial yang signifikan. Mahasiswa dilatih untuk menjembatani teori akademik dengan kebutuhan lapangan, membangun hubungan yang kuat antara GMLS, masyarakat, dan mitra komunitas, serta menghasilkan solusi nyata yang bermanfaat. Pengalaman ini memberikan persiapan berharga bagi mahasiswa untuk berkembang sebagai individu yang adaptif, inovatif, dan berdampak positif bagi masyarakat.

### 3.2 Tugas dan Uraian Kerja Magang

4

Tabel 3.1 merupakan kegiatan magang yang dijalankan selama masa magang September – Desember 2024

Tabel 3. 1 Uraian Kerja Magang

Nomor	Kegiatan Magang	Tanggal Mulai	Tanggal Selesai
1	Mempelajari Project magang yang akan di buat	2 September	31 September
2	Memulai Trip ke dua pergi ke desa Bayah dan mendiskusikan project magang yang dibuat. Mempelajari github dan	1 Oktober	7 Oktober

Nomor	Kegiatan Magang	Tanggal Mulai	Tanggal Selesai
	next.js untuk bisa mengerjakan project tersebut		
3	Memulai Trip ke tiga pergi ke desa Bayah dan mendiskusikan progress project magang. Serta Memulai mengimplementasikan tableau yang sudah dibuat ke localhost GMLS melalui github repository dan VisCode	5 November	11 November
4	Memulai Trip ke empat pergi ke desa Bayah dan mengimplementasikan tableau yang sudah dibuat oleh teman-teman humanity project ke localhost website GMLS	7 Desember	13 Desember
5	Merapihkan dashboard yang sudah muncul di website localhost GMLS.	15 Desember	25 Desember

Tahap Kerja Magang:

1. Tahap Persiapan (2-31 September)

1. Mempelajari secara mendalam mengenai proyek pengembangan website Gugus Mitigasi Lebak Selatan.
  2. Memahami konsep dan penggunaan Tableau API dalam proyek ini.
  3. Menentukan tujuan dan scope dari proyek yang akan dikerjakan.
2. Kunjungan Awal ke GMLS dan Perencanaan Project(1-7 Oktober)
    1. Melakukan kunjungan ke desa Bayah untuk berdiskusi dengan pihak terkait mengenai proyek.
    2. Mengumpulkan data dan informasi yang relevan untuk pengembangan website.
    3. Memulai perencanaan teknis, termasuk belajar github dan next.js
3. Implementasi Tahap Awal(5-11 November)
    1. Melanjutkan diskusi mengenai progres proyek dengan pihak desa Bayah.
    2. Mulai mengimplementasikan Tableau dummy yang sudah dibuat ke dalam localhost website Gugus Mitigasi Lebak Selatan.
    3. Menggunakan GitHub dan Visual Code sebagai tools utama dalam pengembangan.
4. Integrasi Data dari Tim Lain (7-13 Desember)
    1. Mengimplementasikan Tableau yang dibuat oleh tim Humanity Project ke dalam localhost website.
    2. Melakukan pengujian untuk memastikan integrasi data berjalan dengan baik dan memperbaiki errornya di visual studio code.
5. Finalisasi dan Perapian (15-25 Desember)
    1. Merapikan dashboard yang sudah muncul di localhost website.
    2. Melakukan pengujian akhir untuk memastikan website berfungsi dengan baik.

GitHub adalah platform pengembangan software berbasis web di mana orang dapat bekerja sama, menyimpan, dan berbagi kode sumber. GitHub, yang didirikan

pada tahun 2008, telah menjadi salah satu alat yang paling banyak digunakan oleh pengembang software, tim kolaborasi teknologi, dan perusahaan besar yang ingin menjalankan proyek mereka dengan baik. Sistem kontrol versi berbasis Git, yang didirikan oleh Linus Torvalds, pencipta kernel Linux, digunakan oleh GitHub[7].



Gambar 3 1 Logo Github

Fungsi utama GitHub yaitu :

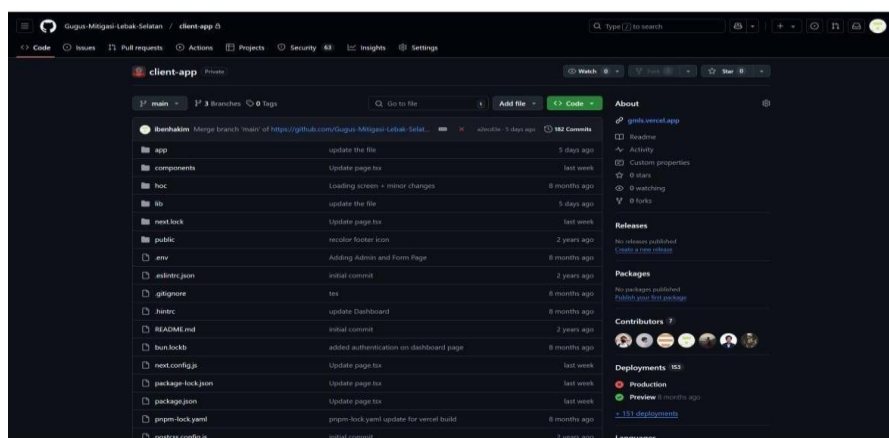
1. Kontrol Versi dan Manajemen Kode, memungkinkan pengguna melacak perubahan dalam kode sumber dan memberikan riwayat perkembangan proyek yang jelas. Pengguna dapat membuat commit untuk menyimpan perubahan, memanfaatkan cabang untuk mengembangkan fitur baru tanpa mengganggu kode utama, dan menggunakan merger untuk menggabungkan perubahan setelah fitur selesai.
2. Kolaborasi Tim

Kolaborasi Tim: Fitur utama GitHub adalah kemampuan untuk memfasilitasi kerja sama. Dengan alat seperti pull request, tim pengembang dapat bekerja pada proyek yang sama secara bersamaan. Ini memungkinkan anggota tim untuk mengusulkan

perubahan, membahasnya, dan kemudian memasukkannya ke dalam proyek utama setelah disetujui.

3. Penyimpanan dan Distribusi Proyek: GitHub menyediakan ruang penyimpanan cloud untuk repositori kode. Dapat mengatur proyek menjadi repositori pribadi yang hanya dapat diakses oleh anggota tertentu atau repositori publik yang dapat diakses oleh semua orang.
4. Manajemen Proyek: GitHub juga memiliki alat untuk manajemen proyek, seperti masalah untuk melacak bug atau fitur baru, dan board proyek untuk mengatur pekerjaan seperti Kanban.
5. Integrasi dan Otomasi GitHub memungkinkan integrasi dengan berbagai layanan lain. Ini termasuk alat pengujian, layanan cloud, platform CI/CD (Integrasi Kontinuitas/Deployment Kontinuitas), dan platform CI/CD. Pengguna dapat mengotomatiskan tugas seperti pengujian kode, penerapan aplikasi, atau pembaruan dokumen melalui fitur GitHub Actions.

GitHub adalah alat penting dalam ekosistem pengembangan perangkat lunak modern. Fiturnya membantu pengembang individu dan tim menciptakan, memelihara, dan berbagi perangkat lunak, dan memungkinkan kolaborasi, manajemen proyek, dan kontrol versi. Memahami dan menggunakan GitHub adalah langkah penting untuk menjadi bagian dari dunia teknologi yang terus berkembang



Gambar 3 2 Github Repository

Repository pengembangan aplikasi klien berbasis web yang digunakan oleh Gugus Mitigasi Lebak Selatan (GMLS) merupakan pusat aktivitas teknologi untuk mendukung tujuan mitigasi bencana. Repository ini dirancang dan dikelola dengan fokus pada inovasi dan efisiensi, menyediakan platform berbasis Next.js yang memungkinkan pengembangan aplikasi yang tangguh dan responsif. Dengan tujuan utama untuk menyediakan informasi bencana secara real-time, aplikasi ini dilengkapi dengan fitur-fitur canggih seperti dashboard berbasis data, peta risiko, serta panduan evakuasi yang relevan bagi masyarakat dan pemangku kepentingan.

Aktivitas pengelolaan repository ini mencerminkan kolaborasi yang dinamis antara para pengembang, sebagaimana terlihat dari banyaknya commit dan pembaruan kode yang dilakukan secara rutin. Dengan lebih dari 150 deploy di lingkungan produksi, repository ini tidak hanya menunjukkan stabilitas dalam pengembangan, tetapi juga integrasi yang matang dengan praktik DevOps modern. Struktur file yang terorganisasi, penggunaan teknologi seperti .env untuk konfigurasi, dan adanya dokumentasi melalui README.md menunjukkan bahwa repository ini dirancang untuk memastikan efisiensi pengembangan dan kemudahan kolaborasi bagi tim.

Selain itu, repository ini menjadi alat penting dalam mendukung keberlanjutan program mitigasi bencana oleh GMLS. Dengan antarmuka yang intuitif dan



backend yang kuat, aplikasi ini memungkinkan pengelolaan data bencana secara real-time dan penyebaran informasi secara luas melalui media sosial maupun dashboard utama. Melalui pengembangan ini, GMLS tidak hanya meningkatkan kesiapan masyarakat dalam menghadapi bencana tetapi juga menunjukkan bagaimana teknologi dapat dimanfaatkan secara strategis untuk mencapai dampak sosial yang signifikan.

Next.js Next.js adalah framework open-source yang memungkinkan pengembangan aplikasi web menggunakan bahasa pemrograman JavaScript. Secara sederhana, ia menawarkan sejumlah fitur dan fungsi yang mempercepat proses pembuatan dan pengembangan aplikasi web[8]. Dengan menggunakan Next.js, framework React, pengembang dapat membuat aplikasi web dengan menggunakan fitur rendering server-side (SSR), pembuatan situs statis (SSG), dan rute API. Dengan menggunakan Next.js, pengembang dapat mengoptimalkan aplikasi mereka untuk kinerja yang lebih baik, SEO yang lebih baik, dan membuat manajemen konten dan routing lebih mudah.



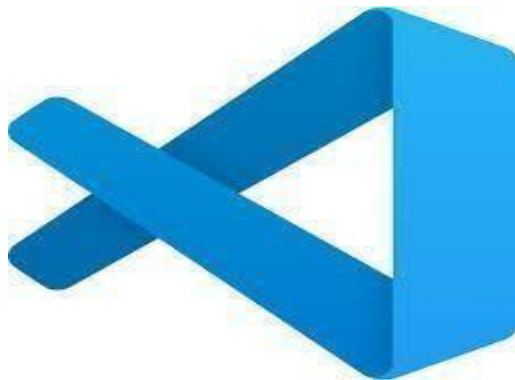
Gambar 3.3 Logo Next.js

Fitur Utama Next.js:

1. Rendering halaman server-side (SSR) adalah keunggulan utama Next.js. Ini menunjukkan bahwa setiap permintaan akan menghasilkan HTML yang telah dirender, yang kemudian dikirim ke browser. Karena konten halaman sudah ada di HTML saat halaman dimuat, halaman pertama dapat dimuat lebih cepat, sehingga lebih baik untuk SEO.

2. Static Site Generation (SSG) Next.js mendukung SSG, yang memungkinkan pembuatan halaman web dengan waktu pembuatan sebelumnya. Artinya, setiap halaman statis dibuat dan ditampilkan sebagai file HTML yang dapat dimuat oleh pengguna dengan cepat ketika aplikasi dijalankan. Ini sangat bermanfaat untuk situs web yang tidak memerlukan konten yang sering diubah.
3. Optimisasi Kinerja Otomatis: Next.js secara otomatis mengoptimalkan kinerja aplikasi sehingga dapat dimuat lebih cepat dengan menggunakan teknik seperti pemisahan kode dan pemuatan, yang meningkatkan pengalaman pengguna dan mengurangi ukuran file yang perlu dimuat pada awalnya.

Visual Studio Code (VS Code) adalah salah satu editor kode sumber terbuka (open-source) dan lintas platform yang dikembangkan oleh Microsoft. Editor ini dirancang untuk mendukung pengembangan perangkat lunak secara efisien dengan berbagai fitur modern yang sangat membantu para developer, baik pemula maupun profesional. VS Code pertama kali dirilis pada April 2015, dan sejak itu, menjadi salah satu editor kode yang paling populer di komunitas pengembang software.



Gambar 3 4 Logo Visual Studio Code

Fitur utama dari VS Code yang populer adalah:

1. Lintas Platform Dukungan untuk Windows, macOS, dan Linux VS Code dapat dijalankan di semua sistem operasi utama, memungkinkan pengembang menggunakan alat yang sama terlepas dari perangkat atau lingkungan kerja mereka.
2. Sinkronisasi Pengaturan Antar Perangkat: Dengan menggunakan akun Microsoft atau GitHub, pengguna dapat menyinkronkan ekstensi, tema, pengaturan, dan shortcut antar perangkat melalui fitur Pengaturan Sinkronisasi.

Keunggulan Visual Studio Code

1. Gratis dan Open Source, tidak ada biaya, dan dapat digunakan oleh individu dan kelompok, termasuk proyek komersial. Komunitas pengembang dapat memeriksa, mengubah, dan berkontribusi pada pengembangan editor dengan kode sumbernya yang tersedia di GitHub.
2. Ringan namun cepat, VS Code adalah IDE yang kuat dan cepat. Ini berbeda dengan IDE berat seperti Visual Studio atau Eclipse. Bahkan pada perangkat dengan spesifikasi rendah, penggunaan ini tidak membutuhkan perangkat keras khusus. Kecepatan Memuat VS Code dibuat untuk memuat proyek besar dengan cepat tanpa lag seperti IDE lain.

Tableau adalah platform Business Intelligence (BI) yang mendukung berbagai sumber data, seperti file Excel, database, layanan cloud, dan platform big data seperti Hadoop, yang membantu organisasi mengubah data mentah menjadi wawasan yang bermanfaat[9].



Gambar 3 5 Logo Tableau

Tableau memiliki banyak fitur yang membuatnya unggul sebagai perangkat lunak visualisasi data. Berikut adalah beberapa fitur utamanya:

1. Antarmuka Drag-and-Drop: Tableau memiliki antarmuka yang mudah digunakan dengan fitur drag-and-drop untuk membuat visualisasi yang mudah digunakan bahkan bagi pengguna yang tidak *familiar* sebelumnya
2. Konektivitas Multi-Sumber Data: Tableau memiliki konektivitas ke lebih dari 70 sumber data, seperti: file lokal seperti Excel dan CSV; database seperti MySQL, PostgreSQL, SQL Server, dan Oracle; platform cloud seperti Google Analytics, Salesforce, dan AWS; dan big data platform seperti Hadoop dan Spark.
3. Dashboard Tableau Interaktif memungkinkan pengguna membuat dashboard yang dinamis dan interaktif. Pengguna dapat melakukan hal-hal berikut: menambahkan filter untuk mempermudah eksplorasi data;
4. Visualisasi Data yang Beragam: Tableau menawarkan berbagai jenis visualisasi data, termasuk diagram batang, garis, dan pie; peta geografis; diagram scatter, heatmap, dan tree map; dan visualisasi kompleks seperti garis dan diagram gantt.

5. API dan ekstensi Tableau dapat diperluas untuk memenuhi kebutuhan tertentu, seperti menggabungkannya dengan aplikasi lain atau menambah fitur unik.

Jenis Produk Tableau: Tableau memiliki banyak produk penting yang dapat digunakan untuk berbagai tujuan:

1. Tableau Desktop Dashboard dibuat dan diposting menggunakan Tableau Desktop. Ideal untuk analisis profesional atau individu.
2. Tableau Server memungkinkan organisasi untuk berbagi dashboard Tableau Desktop dengan tim atau organisasi lainnya.
3. Tableau Cloud (sebelumnya Tableau Online), yang sama seperti Tableau Server, tetapi dikelola di cloud, sehingga tidak memerlukan infrastruktur server internal
4. Tableau Prep, yang digunakan untuk menggabungkan, membersihkan, dan mempersiapkan data sebelum digunakan di Tableau Desktop; dan
5. Tableau Public, versi gratis Tableau yang memungkinkan pengguna membuat visualisasi data dan membagikannya ke publik. Semua ini termasuk dalam kelompok yang mencakup data yang disimpan di server perusahaan.

Implementasi Tableau untuk Gugus Mitigasi Lebak Selatan (GMLS) Tableau tersedia untuk digunakan oleh Gugus Mitigasi Lebak Selatan (GMLS) untuk berbagai tujuan analisis dan pengambilan keputusan strategis yang berkaitan dengan mitigasi bencana. Ini adalah beberapa aplikasinya[10]:

1. Mitigasi Bencana: Memvisualisasikan data historis gempa bumi dan tsunami untuk mengidentifikasi pola dan risiko bencana di wilayah Lebak Selatan. Menganalisis daerah yang paling rentan terhadap bencana menggunakan data geografis dan kepadatan penduduk.

Keunggulan Tableau Dibandingkan Dengan software BI Lain:

1. Tableau dianggap lebih mudah digunakan daripada alat BI lain seperti Power BI atau QlikView.
2. Visualisasi yang Lebih Baik: Tableau memungkinkan visualisasi data yang sederhana dan kompleks
3. Komunitas Besar: Tableau memiliki banyak sumber daya pelatihan, forum, dan pembaruan karena komunitas penggunanya yang tersebar di seluruh dunia.

```

components > Dashboard > TableauEmbed.js > tableauEmbed > useEffect() callback
1  "use client";
2
3  /*
4  ***
5  Tableau can only connect to JavaScript with React (must be client components, thats why we use "use client" despite the reduced speed off
6  Need to wait for future development of Tableau Embedding API
7  ***
8  */
9
10 import { useEffect, useState } from "react";
11
12 // Tableau Embedding API source
13 const TABLEAU_EMBEDDING_API = "https://public.tableau.com/javascripts/api/tableau.embedding.3.latest.min.js";
14

```

Gambar 3 6 Tableau Embedding API

Dengan Tableau Embedding API v3 seperti di gambar 3.6, pengembang dapat menanamkan dasbor Tableau, atau dashboard, ke dalam aplikasi web atau portal pihak ketiga, sehingga pengguna dapat melihat dan berinteraksi dengan data tanpa membuka Tableau secara terpisah

#### Fitur Utama Embedding Api:

1. Tableau dapat dengan cepat dimasukkan ke dalam aplikasi berbasis web karena mendukung JavaScript dan teknologi kontemporer seperti React, Angular, dan Vue.js untuk membuat aplikasi interaktif. Kontrol yang Berinteraksi
2. Fleksibilitas desain dan kemampuan untuk menyaring data langsung dari aplikasi memungkinkan pengguna mengubah parameter visualisasi tanpa kembali ke Tableau.

3. Beri pengembang kontrol penuh untuk menyesuaikan tampilan visualisasi sesuai dengan aplikasi. API mendukung mode responsif, yang berarti dasbor terlihat bagus di berbagai perangkat, seperti ponsel, tablet, dan desktop.

Pengalaman pengguna yang lebih baik dengan Tableau Embedding API v3 Pengguna dapat mengakses visualisasi langsung tanpa berpindah dari Tableau berkat visualisasi yang tertanam, yang membuat pengalaman menggunakan Tableau lebih mudah dan efektif.

```
const loadTableauScript = async () => {
  if (!customElements.get("tableau-viz")) {
    await import(TABLEAU_EMBEDDING_API);
  }

  const tableauViz = document.getElementById("tableauViz");

  // Add resize event listener
  window.addEventListener("resize", () => tableauViz?.resize());

  return () => {
    // Cleanup the resize listener
    window.removeEventListener("resize", () => tableauViz?.resize());
  };
};
```

Gambar 3.7 Code Tableau Embedding

Pada gambar 3.7 Kode JavaScript yang ditampilkan memiliki tujuan utama untuk memuat Tableau Embedding API dan memastikan integrasi visualisasi Tableau dalam aplikasi berbasis web berjalan dengan lancar. Pada bagian awal kode, dilakukan pengecekan untuk memastikan bahwa elemen khusus bernama "tableau-viz" belum ada dalam dokumen. Jika elemen ini belum tersedia, kode akan secara dinamis mengimpor modul Tableau Embedding API. Pendekatan ini tidak hanya meningkatkan efisiensi tetapi juga memastikan bahwa sumber daya hanya dimuat ketika benar-benar diperlukan, mengoptimalkan performa aplikasi.

Selanjutnya, element Tableau yang tertanam diidentifikasi dengan menggunakan metode getElementById pada dokumen, yang mencari elemen dengan ID tertentu, yaitu "tableauViz". Elemen ini merupakan wadah utama untuk

menampilkan visualisasi Tableau dalam aplikasi. Dengan demikian, kode ini memastikan bahwa elemen Tableau terhubung langsung ke antarmuka aplikasi web dan siap untuk merespons berbagai tindakan pengguna. Pendekatan ini memungkinkan pengembang untuk mengintegrasikan Tableau secara mulus ke dalam struktur aplikasi yang ada tanpa hambatan teknis yang signifikan.

```
return (  
  <div id="tableauContainer" className="flex flex-col w-full h-full px-4">  
    <tableau-viz  
      id="tableauViz"  
      src={src}  
      iframe-attr-loading="lazy"  
      className="flex w-full h-full"  
    >  
  </tableau-viz>  
</div>  
);  
  
export default TableauEmbed;
```

Gambar 3 8 Code Tableau Embedding

Gambar 3.8 adalah Komponen React yang ditampilkan dalam kode ini dirancang untuk mengintegrasikan visualisasi data dari Tableau langsung ke dalam antarmuka aplikasi web berbasis React. Komponen ini menggunakan elemen div dengan ID `tableauContainer` untuk membungkus elemen Tableau, memberikan struktur dasar yang fleksibel melalui penggunaan kelas-kelas Tailwind CSS seperti `flex`, `flex-col`, `w-full`, dan `h-full`. Pendekatan ini memastikan bahwa visualisasi Tableau yang dimasukkan dapat secara dinamis menyesuaikan diri dengan tata letak aplikasi, sehingga menciptakan antarmuka yang responsif dan menarik.

Pada inti komponen ini terdapat elemen kustom `<tableau-viz>`, yang merupakan elemen yang digunakan untuk menampilkan visualisasi Tableau. Elemen ini dilengkapi dengan properti `id="tableauViz"` untuk identifikasi, serta atribut `src` yang diisi dengan sumber data Tableau yang akan divisualisasikan. Atribut `iframe-attr-loading="lazy"` juga ditambahkan untuk meningkatkan performa aplikasi dengan menunda pemuatan elemen Tableau hingga benar-benar



diperlukan. Pendekatan ini tidak hanya mengurangi waktu pemuatan halaman secara keseluruhan, tetapi juga memberikan pengalaman pengguna yang lebih mulus, terutama pada aplikasi yang kompleks.

Untuk memastikan visualisasi Tableau terlihat optimal di berbagai perangkat, elemen ini juga menggunakan kelas CSS seperti `w-full` dan `h-full`, yang memungkinkan elemen tersebut untuk memanfaatkan ruang yang tersedia secara maksimal. Hal ini penting untuk menciptakan pengalaman pengguna yang konsisten di berbagai ukuran layar, dari perangkat desktop hingga ponsel. Dengan integrasi yang mulus ini, aplikasi tidak hanya berfungsi sebagai alat analisis data tetapi juga memberikan pengalaman visual yang menarik dan intuitif.

```
1 export const menuDasbor = [  
2   {  
3     name: "Dasbor Titik Evakuasi Situregen",  
4     src: 'https://public.tableau.com/views/HumanityProject_Situregen_TitikEvakuasi/Dashboard1',  
5   },  
6   {  
7     name: "Dasbor Sensus Penduduk Situregen",  
8     src: 'https://public.tableau.com/views/SensusPendudukDesaSituregen/Overview',  
9   },  
10  {  
11    name: "Dasbor Dampak Kejadian Bencana",  
12    src: 'https://public.tableau.com/shared/YGDK734WX',  
13  },  
14 ]
```

Gambar 3 9 Code Tableau

Gambar 3.9 menyajikan sebuah kode yang menjadi fondasi penting dalam aplikasi yang memanfaatkan kekuatan visualisasi data dari Tableau. Kode ini secara khusus merancang struktur data yang akan digunakan untuk membangun sebuah menu navigasi interaktif. Menu ini memungkinkan pengguna untuk dengan mudah memilih tampilan visualisasi Tableau mana yang ingin mereka eksplorasi lebih lanjut.

Fungsi utama dari kode ini adalah untuk:

1. Mendefinisikan Struktur Menu Dasbor: Kode ini secara eksplisit menentukan bagaimana setiap item dalam menu dasbor akan ditampilkan kepada pengguna. Setiap item memiliki dua properti utama: `name` yang

berisi label atau nama yang akan ditampilkan pada menu, dan src yang menyimpan URL unik dari visualisasi Tableau yang terkait. Dengan kata lain, kode ini bertindak sebagai semacam peta yang menghubungkan antara tampilan visual di menu dengan data yang sebenarnya akan ditampilkan.

2. Menyimpan Informasi Visualisasi Tableau: Setiap objek dalam array menuDasbor berfungsi sebagai sebuah entri yang merepresentasikan satu item menu. Properti src pada setiap entri inilah yang menyimpan informasi penting, yaitu URL yang menunjuk langsung ke visualisasi Tableau yang spesifik. Ketika pengguna memilih sebuah item menu, aplikasi akan diarahkan ke URL tersebut dan menampilkan visualisasi yang sesuai.

Secara keseluruhan, kode ini merupakan bagian integral dari mekanisme navigasi dalam aplikasi. Dengan adanya struktur data yang jelas ini, pengembang dapat dengan mudah menambahkan, menghapus, atau mengubah item menu tanpa perlu melakukan perubahan yang signifikan pada bagian lain dari aplikasi. Selain itu, pengguna juga akan mendapatkan pengalaman yang lebih baik karena mereka dapat dengan cepat menemukan informasi yang mereka butuhkan melalui menu yang terorganisir dengan baik.

Implementasi kode ini memungkinkan aplikasi untuk menyajikan berbagai macam visualisasi Tableau kepada pengguna dalam satu antarmuka yang terpadu. Pengguna dapat dengan bebas menjelajahi data yang berbeda-beda, seperti data titik evakuasi, sensus penduduk, atau dampak bencana, hanya dengan mengklik pilihan yang tersedia pada menu. Hal ini sangat berguna dalam berbagai konteks, mulai dari analisis bisnis hingga pemantauan kondisi lingkungan.

```
return (  
  <div id="tableauContainer" className="flex flex-col w-full h-full px-4">  
    <tableau-viz  
      id="tableauViz"  
      src={src}  
      iframe-attr-loading="lazy"  
      className="flex w-full h-full"  
    >  
    </tableau-viz>  
  </div>  
);  
  
export default TableauEmbed;
```

Gambar 3 10 Code Embedding API

Kode pada Gambar 3.10 merupakan komponen inti dalam sebuah aplikasi yang memanfaatkan Next.js dan visualisasi data dari Tableau. Komponen ini berperan sebagai fondasi atau kerangka utama bagi seluruh halaman dalam aplikasi, mengatur struktur, tampilan, dan fungsionalitas dasar.

Secara keseluruhan, kode ini merupakan komponen kunci dalam pengembangan aplikasi berbasis Next.js yang menggunakan Tableau. Dengan memahami fungsi dari setiap bagian kode, kita dapat lebih mudah memodifikasi dan memperluas aplikasi sesuai dengan kebutuhan.

```
1 "use client";  
2  
3 import { menuDasbor } from '@lib/menuDasbor';  
4 import { useSelectedLayoutSegment } from 'next/navigation';  
5 import Link from 'next/link';  
6  
7 const DashboardSidebar = () => {  
8   return (  
9     <div className='flex flex-col w-auto gap-4 focus:ring-1 p-4'>  
10      {menuDasbor?.map((menu, i) => (  
11        <Link  
12          key={i}  
13          className='font-medium text-left px-2 focus:ring-1 rounded-xl'  
14          href={`./dashboard/${i}`}  
15        >  
16          {`${menu?.name}`}  
17        </Link>  
18      )  
19    )  
20  );  
21 };  
22  
23 export default DashboardSidebar  
24
```

Gambar 3 11 Code Menu Dashboard

Kode di atas adalah sebuah komponen React yang dinamakan DashboardSidebar, ditulis dengan JavaScript menggunakan framework Next.js.

Komponen ini memiliki fungsi utama untuk membuat sidebar navigasi pada halaman dashboard. Berikut adalah fungsi utama dari kode:

1. Komponen `DashboardSidebar` didefinisikan menggunakan pendekatan arrow function (`const DashboardSidebar = () => {}`). Pada bagian atas kode, beberapa modul dan fungsi diimpor, seperti `menuDasbor` dari library lokal (`@/lib/menuDasbor`), `useSelectedLayoutSegment` dari `Next.js` (`next/navigation`), dan komponen `Link` dari `next/link`. Fungsi `menuDasbor` kemungkinan berisi data atau daftar menu yang akan ditampilkan di sidebar. Komponen `Link` digunakan untuk navigasi antar halaman di aplikasi `Next.js`.

2. Struktur JSX Utama:

Dalam return, kode menggunakan elemen `<div>` sebagai container utama untuk sidebar. Elemen ini diberikan beberapa kelas CSS yang didefinisikan dalam properti `className`. Properti ini memanfaatkan utility-first CSS framework (`Tailwind CSS`). Beberapa kelas yang digunakan seperti `flex`, `flex-col`, `gap-4`, `w-auto`, dan `p-4` bertujuan untuk membuat layout sidebar dalam bentuk kolom vertikal dengan jarak antar elemen yang teratur.

3. Iterasi Data Menu:

Sidebar dibuat dinamis dengan menggunakan metode `.map()` untuk iterasi melalui data `menuDasbor`. Setiap item dalam data menu dirender menjadi sebuah elemen `<Link>`. Properti `key` diberikan untuk setiap elemen yang dihasilkan iterasi, menggunakan indeks `i` dari metode `.map()`. Hal ini penting dalam `React` untuk menghindari peringatan dan meningkatkan efisiensi rendering.

4. Komponen Link Navigasi:

Elemen `<Link>` di setiap iterasi digunakan untuk membuat tautan navigasi ke halaman tertentu. Properti `href` menentukan tujuan tautan, yang dihasilkan secara dinamis berdasarkan nilai indeks `i` (`/dashboard/${i}`). Teks tautan yang ditampilkan adalah properti `name` dari setiap item `menuDasbor`.

Properti `className` dalam `<Link>` menambahkan styling untuk teks navigasi, seperti membuatnya rata kiri (`text-left`) dan memberikan padding (`px-2`).

#### 5. Ekspor Komponen:

Di bagian akhir, komponen `DashboardSidebar` diekspor menggunakan `export default`, sehingga dapat digunakan di file lain dalam proyek. Komponen ini cocok untuk digunakan dalam aplikasi dashboard berbasis Next.js yang memerlukan sidebar dinamis. Dengan menggunakan data menu dari `menuDasbor`, kode ini memungkinkan pengembang untuk mengubah menu navigasi hanya dengan memodifikasi data yang dipakai.

```
import Navbar from '@components/Navbar';
import dynamic from 'next/dynamic';
import { menuDasbor } from '@lib/menuDasbor';

// import Dashboard from '@components/FirebaseCrud/Dashboard';

const DashboardSidebar = dynamic(() => import("@/components/Dashboard/DashboardSidebar"), { ssr: true });
const TableauEmbed = dynamic(() => import("@/components/Dashboard/TableauEmbed"), { ssr: true });

type DashboardPageProps = {
  params: DashboardPageParam;
};

type DashboardPageParam = {
  id: number;
};

const Page = (props: DashboardPageProps) => {
  const { id = 0 } = props.params; // default 0
```

Gambar 3 12 Dashboard SideBar

Kode di atas merupakan bagian dari aplikasi Next.js yang bertujuan untuk menampilkan dashboard dengan navigasi dan visualisasi data dari Tableau. Berikut fungsi kodenya:

1. **Import Komponen dan Library:** Di bagian awal, kode mengimpor berbagai komponen dan library yang dibutuhkan. `Navbar` adalah komponen navigasi utama yang diimpor dari folder lokal, kemungkinan digunakan untuk menampilkan menu di bagian atas aplikasi. Selain itu, fungsi `dynamic`

dari Next.js digunakan untuk memuat komponen secara dinamis. Komponen menuDasbor diimpor untuk mendapatkan daftar menu yang digunakan dalam sidebar dashboard. Adapun bagian kode yang mengimpor komponen Dashboard dari FirebaseCrud/Dashboard tampaknya dikomentari, mungkin karena tidak digunakan dalam kode ini.

2. **Dynamic Import untuk Lazy Loading:** Komponen DashboardSidebar dan TableauEmbed dideklarasikan menggunakan dynamic() dari Next.js. Fitur ini memungkinkan lazy loading, di mana komponen hanya dimuat ketika diperlukan. Hal ini berguna untuk meningkatkan performa aplikasi dengan mengurangi ukuran awal bundle yang dimuat di browser. Kedua komponen ini juga diatur agar mendukung server-side rendering (SSR) melalui opsi { ssr: true }. SSR memungkinkan komponen untuk dirender di server sebelum dikirim ke klien, memberikan waktu loading awal yang lebih cepat dan meningkatkan SEO.
3. **Definisi Tipe dengan TypeScript:** Kode memanfaatkan kemampuan TypeScript untuk mendefinisikan tipe data, sehingga membantu mencegah kesalahan. DashboardPageProps adalah tipe yang merepresentasikan properti yang diterima oleh komponen Page, dan memiliki struktur params yang bertipe DashboardPageParam. Tipe DashboardPageParam mendefinisikan parameter ID sebagai bilangan (id: number). Dengan pendekatan ini, pengembang dapat dengan mudah memahami struktur data yang digunakan dan mendapatkan manfaat dari validasi tipe pada saat pengembangan.

```
return [  
  <div className="block">  
    <Navbar /> You, 3 weeks ago + Update page.tsx  
    <div className='flex flex-col max-w-screen py-28 gap-4'>  
      <h2 className="text-2xl font-bold text-center">  
        {menuDasbor?.[id]?.name}  
      </h2>  
      <div className='flex flex-col lg:flex-row gap-2'>  
        <TableauEmbed  
          src={menuDasbor?.[id]?.src} />  
        <DashboardSidebar />  
      </div>  
    </div>  
  </div >  
]  
export default Page
```

Gambar 3 13 MenuDasboard

Kode pada gambar ini adalah bagian dari komponen halaman utama dalam aplikasi Next.js yang menampilkan berbagai dashboard. Dashboard yang ditampilkan bergantung pada identitas (id) yang diterima sebagai props. Berikut penjelasan kode tersebut:

1. Struktur Dasar dan Komponen Navbar: Komponen ini dimulai dengan elemen `<div>` dengan `className="block"` yang menjadi pembungkus utama untuk keseluruhan halaman. Di dalamnya, komponen Navbar dipanggil di bagian atas untuk menampilkan navigasi global aplikasi. Navigasi ini biasanya mencakup menu atau tautan yang membantu pengguna berpindah antara berbagai bagian aplikasi. Dengan penempatan di awal halaman, Navbar berfungsi sebagai elemen konstan yang muncul di setiap halaman dashboard.
2. Penggunaan `menuDasbor` untuk Dinamisasi: Di bawah elemen Navbar, terdapat elemen `<div>` lain dengan kelas CSS yang menggunakan utility Tailwind CSS seperti `flex`, `flex-col`, dan `gap-4` untuk mengatur layout. Elemen `<h2>` di dalamnya memanfaatkan data dari `menuDasbor` untuk menampilkan judul dashboard berdasarkan ID yang diterima. Properti `menuDasbor?.[id]?.name` memastikan bahwa hanya nama dashboard yang sesuai dengan ID yang diterima akan ditampilkan. Hal ini membuat komponen ini fleksibel untuk menampilkan berbagai dashboard dengan judul yang berbeda sesuai data.
3. Visualisasi Tableau dengan `TableauEmbed`: Komponen ini juga menyertakan `TableauEmbed`, yang merupakan komponen dinamis untuk menampilkan visualisasi Tableau. Properti `src` dari `TableauEmbed` diisi dengan nilai `menuDasbor?.[id]?.src`, yang menentukan sumber data Tableau yang akan digunakan untuk visualisasi. Dengan demikian, visualisasi Tableau yang spesifik untuk setiap dashboard dapat ditampilkan berdasarkan data

menuDasbor dan ID yang diterima. Pendekatan ini memudahkan integrasi visualisasi Tableau secara dinamis dan efisien.

4. Integrasi Sidebar Navigasi: Di bawah komponen Tableau, terdapat komponen DashboardSidebar. Sidebar ini memberikan navigasi tambahan bagi pengguna untuk berpindah antar halaman dashboard atau fitur lain dalam aplikasi. Dengan penggunaan layout fleksibel melalui Tailwind CSS (`lg:flex-row` dan `gap-2`), sidebar dan visualisasi Tableau diatur untuk tampil berdampingan pada layar lebar, namun tersusun secara vertikal pada layar kecil. Hal ini memastikan responsivitas antarmuka pengguna.

Mengembangkan aplikasi berbasis Node.js telah menjadi salah satu praktik yang sangat populer, terutama dengan hadirnya berbagai framework seperti Next.js, Vue.js, Nuxt.js, dan React.js. Salah satu perintah yang paling sering digunakan oleh pengembang dalam proses ini adalah `npm run dev`. Perintah ini berfungsi untuk memulai server pengembangan (`development server`) yang memungkinkan pengembang menjalankan aplikasi dalam mode pengembangan. Mode ini memberikan fleksibilitas untuk melakukan iterasi cepat dan melihat perubahan secara real-time, sehingga sangat mendukung produktivitas selama pengembangan aplikasi.

```
D:\client-app>npm run dev
> gmls-app@0.1.0 dev
> next dev

▲ Next.js 14.2.14
- Local:      http://localhost:3000
- Environments: .env

✓ Starting...
✓ Ready in 2.3s
```

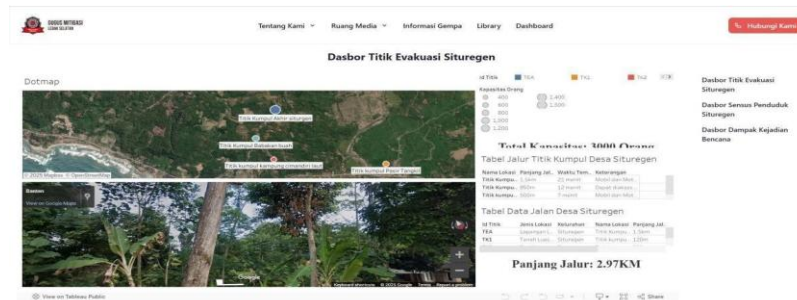
Gambar 3 14 NPM RUN DEV



Saat command `npm run dev` dijalankan, seperti di gambar 3.14 ini, Node Package Manager (npm) akan membaca konfigurasi yang ada di dalam file `package.json` proyek tersebut. File `package.json` biasanya berisi berbagai skrip yang telah ditentukan sebelumnya untuk mempermudah eksekusi perintah tertentu. Dalam konteks server pengembangan, `npm run dev` sering dikaitkan dengan skrip yang menjalankan perintah bawaan framework, seperti `next dev` untuk Next.js atau `nuxt` untuk Nuxt.js. Skrip ini secara otomatis mengaktifkan server lokal, yang biasanya dapat diakses melalui alamat seperti <http://localhost:3000>.

Framework-framework modern seperti Next.js dirancang untuk mempermudah proses pengembangan dengan menyediakan fitur bawaan seperti hot reloading. Hot reloading memungkinkan pengembang melihat perubahan kode tanpa perlu me-refresh halaman secara manual. Hal ini sangat penting dalam pengembangan aplikasi yang kompleks, di mana setiap detik yang dihemat bisa meningkatkan efisiensi secara keseluruhan. Dengan menjalankan server pengembangan menggunakan `npm run dev`, pengembang bisa fokus pada logika aplikasi tanpa harus khawatir tentang konfigurasi server.

Dashboard titik evakuasi Situregen adalah alat visual yang dirancang untuk menyajikan informasi penting terkait titik-titik evakuasi di wilayah Situregen secara terintegrasi dan mudah dipahami. Dashboard ini berperan sebagai pusat informasi yang dapat diakses oleh berbagai pihak, mulai dari masyarakat umum hingga pihak berwenang, dalam rangka meningkatkan kesiapsiagaan menghadapi bencana.



Gambar 3 15 Dasboard Titik Evakuasi

### Fungsi Utama Dashboard titik evakuasi:

1. **Visualisasi Lokasi:** Melalui peta interaktif yang terintegrasi, dashboard ini memungkinkan pengguna untuk dengan mudah mengidentifikasi lokasi-lokasi titik evakuasi di wilayah Situregen. Peta ini memberikan gambaran yang jelas mengenai sebaran titik evakuasi dan jaraknya dari pemukiman penduduk.
2. **Informasi Kapasitas:** Dashboard juga menyajikan informasi mengenai kapasitas maksimal setiap titik evakuasi. Data ini sangat penting untuk perencanaan evakuasi, terutama dalam situasi darurat ketika jumlah pengungsi dapat meningkat secara signifikan.
3. **Data Demografi:** Selain informasi tentang titik evakuasi, dashboard ini juga menyajikan data demografi seperti jumlah penduduk. Informasi ini berguna untuk mengestimasi kebutuhan logistik dan sumber daya lainnya dalam situasi darurat.
4. **Analisis Jalur Evakuasi:** Dashboard mungkin juga menampilkan informasi mengenai jalur evakuasi yang menghubungkan titik-titik kumpul dengan titik evakuasi utama. Informasi ini sangat penting untuk memastikan kelancaran proses evakuasi.

5. Simulasi Skenario Bencana: Beberapa dashboard yang lebih canggih mungkin dilengkapi dengan fitur simulasi yang memungkinkan pengguna untuk memodelkan berbagai skenario bencana dan memprediksi dampaknya terhadap masyarakat.
6. Integrasi dengan Sistem Informasi Lain: Dashboard ini dapat diintegrasikan dengan sistem informasi lainnya, seperti sistem peringatan dini, untuk memberikan informasi yang lebih komprehensif dan up-to-date.

#### Manfaat Dashboard Titik Evakuasi

1. Meningkatkan Kesiapsiagaan: Dengan akses mudah ke informasi yang relevan, masyarakat dapat lebih siap menghadapi bencana dan mengetahui tindakan yang harus dilakukan.
2. Mempercepat Respon Darurat: Pihak berwenang dapat memanfaatkan dashboard ini untuk mengkoordinasikan upaya penyelamatan dan penanggulangan bencana secara lebih efektif.
3. Meningkatkan Efisiensi Penggunaan Sumber Daya: Informasi yang akurat mengenai kapasitas titik evakuasi dan jumlah penduduk dapat membantu dalam mengalokasikan sumber daya secara efisien.
4. Fokus pada Pencegahan: Dashboard ini juga dapat digunakan sebagai alat untuk mengidentifikasi area yang rawan bencana dan merumuskan langkah-langkah pencegahan.

Berikut beberapa fitur dari dashboard titik evakuasi yang diklik:

##### 1. **Kapasitas Orang**

Filter berbentuk lingkaran untuk mengatur tampilan berdasarkan kapasitas titik kumpul (400, 600, 800, 1000, 1200, 1400, 1500). Kemungkinan dengan mengklik salah satu opsi, peta akan menampilkan titik kumpul dengan kapasitas tersebut.

Cara Kerja:

Pengguna memilih salah satu opsi kapasitas (400, 600, 800, dst.). Dashboard memfilter titik kumpul pada peta untuk menampilkan hanya lokasi dengan kapasitas sesuai pilihan. Peta diperbarui secara dinamis untuk memvisualisasikan data yang difilter.

## 2. **Label Data dan Jalur Titik Kumpul Desa Situregen**

Kolom dalam table interaktif, misalnya untuk melihat informasi lebih detail mengenai jalur, jenis lokasi, atau waktu tempuh.

Cara Kerja: Pengguna mengklik salah satu kolom atau baris pada tabel. Dashboard akan menampilkan informasi lebih rinci, seperti peta jalur, waktu perjalanan, atau kondisi lokasi. Data tambahan dapat muncul dalam bentuk tooltip, dan pop-up

## 3. **Titik Kumpul di Peta**

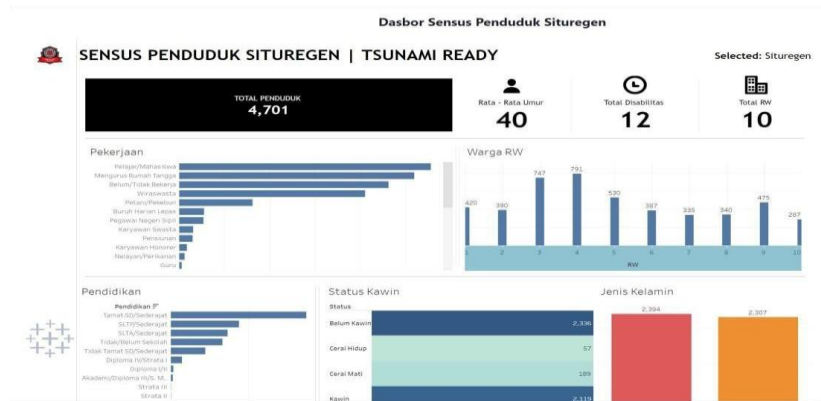
Titik pada peta, seperti "Titik Kumpul Babakan Buah" dan "Titik Kumpul Pasir Tangkil," kemungkinan dapat diklik untuk menampilkan informasi lebih detail tentang titik tersebut.

Cara Kerja: Pengguna mengklik salah satu ikon titik kumpul di peta. Dashboard akan menampilkan informasi tambahan, seperti nama lokasi, kapasitas, atau waktu tempuh. Informasi ini dapat muncul dalam bentuk tooltip atau panel samping di peta.

## 4. **Google Street View**

Pada bagian bawah, terdapat tampilan "Google Street View" yang menunjukkan lokasi lapangan atau area di sekitar.

Cara Kerja: Pengguna menggunakan fitur navigasi di tampilan Street View. Dashboard memuat pemandangan lokasi secara interaktif, memungkinkan pengguna untuk memutar, memperbesar, atau menjelajahi area di sekitar titik kumpul. Lokasi spesifik dapat dipilih dengan klik pada peta atau melalui tabel jalur.



Gambar 3 16 Dashboard Sensus Penduduk desa Situregen

Dashboard sensus penduduk Situregen merupakan sebuah visual yang dirancang untuk memberikan gambaran yang komprehensif mengenai karakteristik demografi masyarakat di wilayah Situregen. Informasi yang disajikan dalam dashboard ini memiliki peran krusial dalam perencanaan dan pelaksanaan mitigasi bencana, khususnya bencana tsunami. Menyajikan data demografi melalui visualisasi:

1. Pekerjaan: Menampilkan distribusi penduduk berdasarkan jenis pekerjaan yang ditawarkan. Memahami struktur ekonomi masyarakat dan potensi dampak bencana terhadap mata pencaharian menjadi lebih mudah dengan data ini.
2. Warga per RW: Menunjukkan berapa banyak orang yang hidup di setiap RW (Rukun Warga). Distribusi bantuan dan perencanaan evakuasi bergantung pada data ini.
3. Pendidikan: Menunjukkan tingkat pendidikan yang dimiliki penduduk. Dengan data ini, dapat mengetahui seberapa sadar masyarakat terhadap risiko bencana dan seberapa baik mereka mengikuti petunjuk evakuasi.
4. Status Pernikahan: Menunjukkan status perkawinan seseorang. Kelompok yang rentan, seperti orang tua atau ibu hamil, dapat diidentifikasi dengan data ini.

5. Jenis Kelamin: Menampilkan perbedaan antara laki-laki dan perempuan dalam jumlah penduduk. Perencanaan logistik dan medis bergantung pada informasi ini.
6. Identifikasi kelompok rentan: Kelompok yang lebih rentan terhadap dampak bencana dapat diidentifikasi dengan melihat data seperti pekerjaan, pendidikan, status kawin, dan jenis kelamin.
7. Perencanaan Evakuasi: Jumlah penduduk per RW dapat digunakan untuk menentukan rute evakuasi dan titik kumpul yang efektif.
8. Distribusi bantuan: Menggunakan data demografi dapat membantu menentukan jenis bantuan apa yang dibutuhkan oleh masyarakat setelah bencana.

Berikut adalah filter dashboard sensus penduduk situregen:

#### 1. **Pekerjaan**

Barchat pada kategori pekerjaan, seperti "Pelajar/Mahasiswa," "Mengurus Rumah Tangga," atau "Petani," dapat diklik untuk memperinci data pekerjaan lebih lanjut atau memfilter data terkait kategori tersebut.

Cara Kerja: Pengguna mengklik salah satu batang (misalnya, "Pelajar/Mahasiswa"). Dashboard memfilter seluruh data pada dasbor untuk menampilkan statistik hanya dari kategori pekerjaan yang dipilih. Data terkait, seperti distribusi RW, tingkat pendidikan, atau status kawin, diperbarui sesuai dengan pilihan.

#### 2. **Warga RW**

Barchat berdasarkan RW (1-10) diklik untuk memfilter data berdasarkan RW tertentu. Misalnya, mengklik batang untuk RW 1 akan menampilkan data khusus untuk RW tersebut.

Cara Kerja: Pengguna mengklik salah satu barchat pada data RW (misalnya, RW3). Dashboard akan memfilter seluruh data untuk

menampilkan statistik yang hanya terkait dengan RW yang dipilih. Informasi seperti pekerjaan, pendidikan, status kawin, dan jenis kelamin diperbarui untuk mencerminkan data RW tersebut.

### 3. **Pendidikan**

Barchat yang menunjukkan tingkat pendidikan, seperti "Tamat SD/Sederajat," "SLTP/Sederajat," dan "Diploma/D-IV/S1," dapat diklik untuk memperinci atau memfilter data berdasarkan tingkat pendidikan tertentu.

Cara Kerja: Pengguna mengklik salah satu batang (misalnya, "Diploma/D-IV/S1"). Dashboard menampilkan data yang hanya relevan dengan tingkat pendidikan yang dipilih. Data lain di dasbor, seperti pekerjaan atau status kawin, diperbarui untuk mencerminkan data pendidikan tersebut.

### 4. **Kawin**

Barchat untuk status kawin (Belum Kawin, Cerai Hidup, Cerai Mati, dsb.) dapat diklik untuk menampilkan informasi lebih detail atau memfilter data.

Cara Kerja: Pengguna mengklik salah satu batang (misalnya, "Belum Kawin"). Dashboard memfilter seluruh data pada dasbor untuk menampilkan statistik yang hanya berkaitan dengan status kawin yang dipilih. Visualisasi lain, seperti pekerjaan, pendidikan, atau distribusi RW, akan diperbarui sesuai dengan pilihan.

### 5. **Kelamin**

Barchat yang menunjukkan jumlah penduduk berdasarkan jenis kelamin (laki-laki dan perempuan) dapat diklik untuk memperinci data terkait jenis kelamin tertentu.

Cara Kerja: Pengguna mengklik salah satu barchat (misalnya, "Perempuan"). Dashboard akan memfilter seluruh data untuk hanya menampilkan statistik penduduk perempuan. Semua data lain, seperti

pekerjaan, pendidikan, atau status kawin, diperbarui untuk mencerminkan data sesuai dengan jenis kelamin yang dipilih.



Gambar 3 17 Dashboard Dampak Kejadian Bencana(total kejadian)

Dashboard dampak kejadian bencana ini berfungsi untuk memahami pola dan tren kejadian bencana di suatu wilayah, khususnya di Indonesia yang dikenal sebagai negara dengan tingkat kerentanan bencana yang tinggi. Dashboard ini berfungsi sebagai jendela untuk melihat sejarah bencana, mengidentifikasi daerah-daerah yang rawan bencana, serta memberikan gambaran mengenai jenis bencana yang paling sering terjadi.

#### Fungsi Utama Dashboard

1. Visualisasi Sebaran Bencana: Melalui peta interaktif, dashboard ini menyajikan informasi mengenai lokasi-lokasi di mana bencana pernah terjadi. Hal ini memungkinkan kita untuk mengidentifikasi daerah-daerah yang sering dilanda bencana dan memiliki risiko yang lebih tinggi.
2. Analisis Frekuensi Kejadian: Grafik batang pada dashboard menunjukkan jumlah kejadian bencana dalam kurun waktu tertentu. Dengan melihat tren peningkatan atau penurunan jumlah kejadian, kita dapat mengidentifikasi pola-pola tertentu dan mengantisipasi potensi bencana di masa depan.



3. **Klasifikasi Jenis Bencana:** Dashboard biasanya mengklasifikasikan jenis bencana, seperti gempa bumi, banjir, tanah longsor, dan sebagainya. Hal ini memungkinkan kita untuk menganalisis jenis bencana yang dominan di suatu wilayah dan mengembangkan strategi mitigasi yang sesuai.
4. **Dukungan Pengambilan Keputusan:** Informasi yang disajikan dalam dashboard ini sangat berguna untuk pengambilan keputusan dalam berbagai aspek penanggulangan bencana, mulai dari perencanaan mitigasi hingga respons darurat.
5. **Integrasi Tableau dengan website** memberikan manfaat signifikan, terutama dalam konteks mitigasi bencana seperti di GMLS, karena memungkinkan penyajian informasi yang real-time, interaktif, dan mudah diakses oleh masyarakat. Dengan menggunakan Tableau API, website dapat menjadi pusat informasi yang handal dan mendukung pengambilan keputusan berbasis data.

#### Penerapan dalam Mitigasi Bencana

1. **Perencanaan Tata Ruang:** Data dari dashboard ini dapat digunakan sebagai dasar dalam perencanaan tata ruang wilayah, dengan menghindari pembangunan di daerah-daerah yang rawan bencana.
2. **Pengembangan Infrastruktur Tahan Bencana:** Informasi mengenai jenis bencana yang sering terjadi dapat digunakan untuk mendesain infrastruktur yang tahan terhadap bencana.
3. **Sosialisasi dan Edukasi:** Data mengenai risiko bencana dapat digunakan untuk meningkatkan kesadaran masyarakat akan pentingnya kesiapsiagaan.
4. **Pengembangan Kebijakan:** Informasi dari dashboard ini dapat menjadi dasar dalam pengembangan kebijakan terkait penanggulangan bencana.

Beberapa filter yang bisa digunakan dalam dashboard ini, yaitu:

1. Filter Jenis Kejadian

Fitur ini memungkinkan pengguna untuk memilih jenis bencana tertentu, seperti gempa atau tsunami, sehingga hanya data yang relevan akan ditampilkan di dashboard. Saat pengguna mengklik dropdown atau daftar opsi filter ini, visualisasi pada grafik dan peta akan diperbarui secara otomatis untuk hanya menampilkan data sesuai jenis kejadian yang dipilih..

2. Filter Waktu (Tahun)

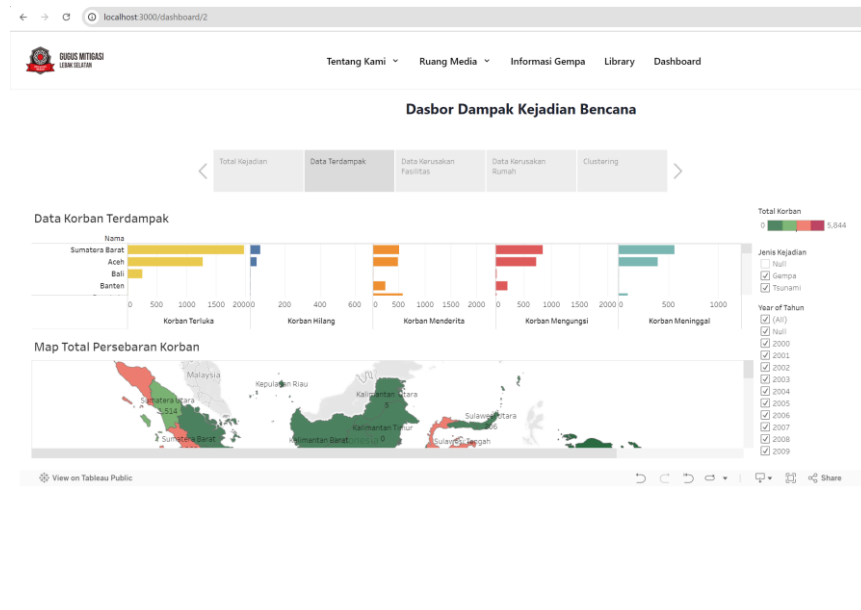
Pengguna dapat memanfaatkan fitur ini untuk memfilter data berdasarkan tahun tertentu atau rentang waktu tertentu. Dengan memilih periode waktu yang diinginkan, tren kejadian bencana dari tahun ke tahun akan terlihat pada barchart di bagian atas, yang secara dinamis akan diperbarui untuk mencerminkan data dari periode tersebut.

3. Filter Jumlah Kejadian

Filter ini menggunakan skala warna untuk menunjukkan intensitas atau jumlah kejadian bencana, membantu pengguna memahami distribusi kejadian secara geografis pada peta. Dengan fitur ini, pengguna dapat dengan mudah mengidentifikasi area dengan intensitas kejadian yang lebih tinggi atau lebih rendah.

4. Interaksi dengan Peta (Map Total Persebaran Kejadian)

Peta interaktif ini memberikan kemampuan kepada pengguna untuk mengklik wilayah tertentu, seperti provinsi atau kabupaten, guna melihat data kejadian bencana yang lebih rinci di lokasi tersebut. Setelah wilayah diklik, dashboard akan menampilkan informasi tambahan, seperti rincian jumlah kejadian yang terjadi di wilayah tersebut dalam periode yang dipilih.



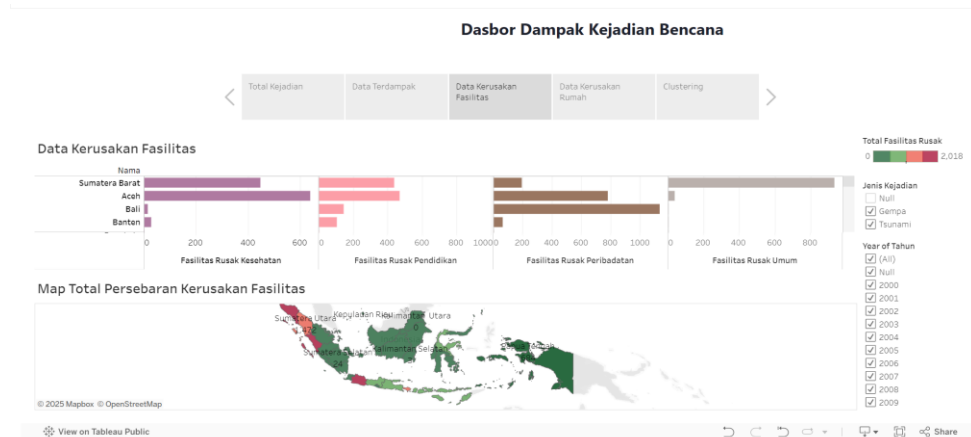
Gambar 3 18 Dashboard Dampak Kejadian Bencana(Data terdampak)

Dashboard diatas menyajikan informasi terkait dampak kejadian bencana dengan fokus pada data korban. Visualisasi data mencakup grafik barchat yang menunjukkan jumlah korban terluka, hilang, menderita, mengungsi, dan meninggal di berbagai wilayah, serta peta persebaran korban di Indonesia. Filter "Jenis Kejadian" dan "Year of Tahun" memungkinkan pengguna untuk menyaring data berdasarkan jenis bencana dan tahun kejadian. Dashboard ini bertujuan untuk memantau dampak bencana, menganalisis tren, dan membantu pengambilan keputusan terkait penanganan bencana dan alokasi bantuan.

Berikut beberapa filter yang bisa digunakan di dashboard tersebut:

1. **Jenis Kejadian:** Memungkinkan pengguna untuk memfilter data berdasarkan jenis bencana (Gempa, Tsunami, dll.).
2. **Year of Tahun:** Memungkinkan pengguna untuk menampilkan data berdasarkan tahun kejadian.

3. **Map total Persebaran Kerusakan Fasilitas:** Pengguna mengklik provinsi mana yang akan dilihat maka visualisasi lain seperti data korban terdampak akan menyesuaikan terhadap provinsi mana yang akan dipilih.

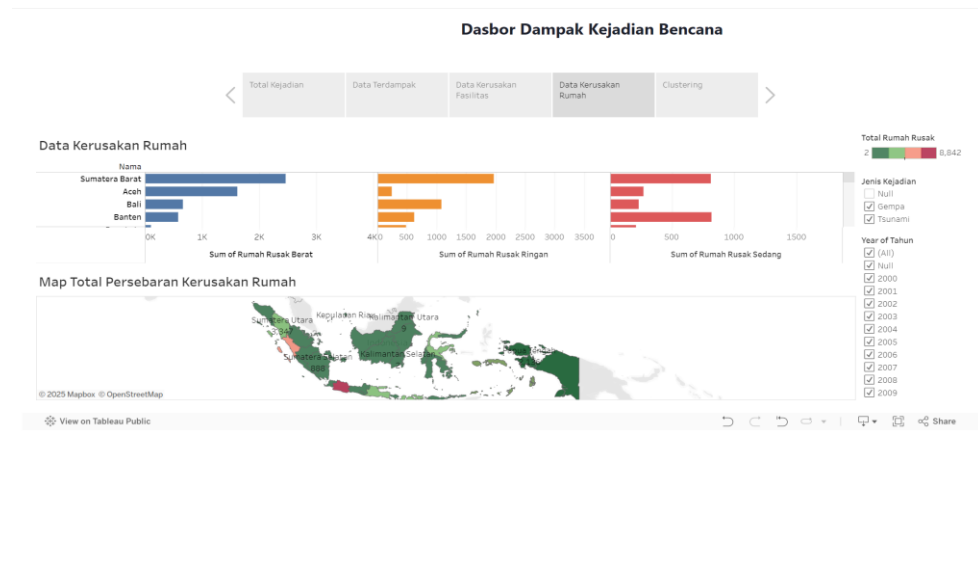


Gambar 3 19 Dashboard Dampak Kejadian Bencana(Data Kerusakan Fasilitas)

Dashboard diatas menyajikan informasi kerusakan fasilitas akibat kejadian bencana. Visualisasi data berupa grafik barchat menampilkan jumlah kerusakan fasilitas kesehatan, pendidikan, peribadatan, dan umum di beberapa wilayah, dilengkapi peta persebaran kerusakan fasilitas di Indonesia. Filter "Jenis Kejadian" dan "Year of Tahun" memungkinkan pengguna untuk menyaring data berdasarkan jenis bencana dan tahun kejadian. Dashboard ini berfungsi untuk memantau kerusakan fasilitas akibat bencana, menganalisis tren, dan mendukung pengambilan keputusan terkait perbaikan dan rekonstruksi pasca bencana.

Berikut beberapa filter yang bisa digunakan di dashboard tersebut:

1. **Jenis Kejadian:** Memungkinkan pengguna untuk memfilter data berdasarkan jenis bencana, seperti Gempa, dan Tsunami.
2. **Year of Tahun:** Memungkinkan pengguna untuk memfilter data berdasarkan tahun kejadian bencana, seperti 2000, 2001, 2002, dan seterusnya.
3. **Map total Persebaran Kerusakan Fasilitas:** Pengguna menklik provinsi mana yang akan dilihat maka visualisasi lain seperti data kerusakan fasilitas akan menyesuaikan terhadap provinsi mana yang akan dipilih.



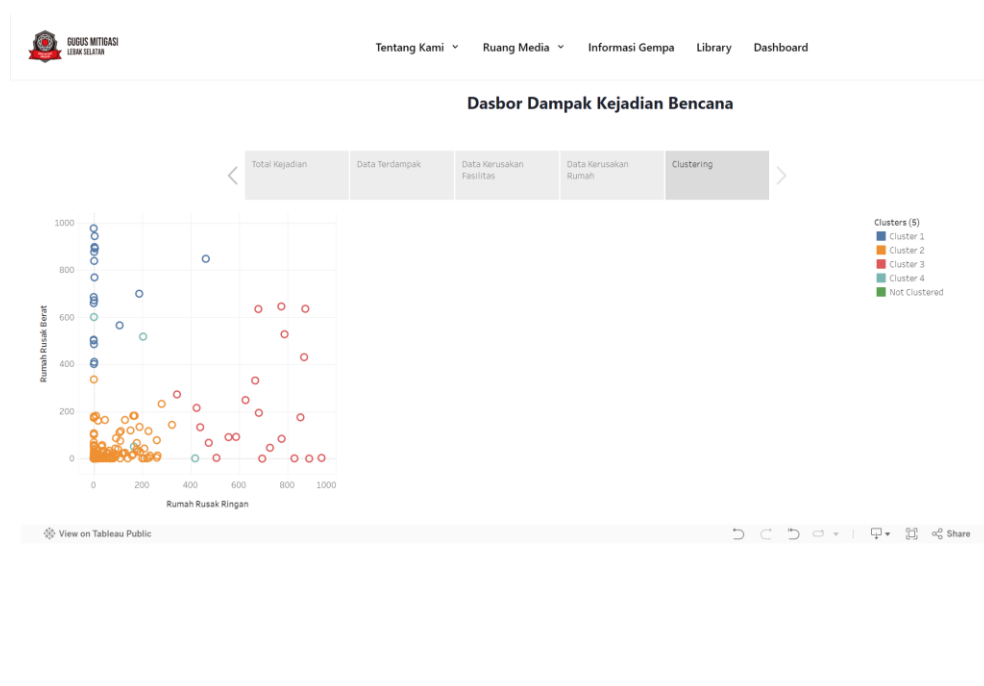
Gambar 3 20 Dashboard Dampak Kejadian Bencana(Data Keruskan Rumah)

Dashboard diatas menyajikan data kerusakan rumah akibat bencana. Grafik barchat menampilkan jumlah rumah rusak berat, ringan, dan sedang di beberapa wilayah di Indonesia. Peta di bawahnya memvisualisasikan sebaran kerusakan rumah tersebut secara geografis. Filter "Jenis Kejadian" dan "Year of Tahun" memungkinkan pengguna untuk menganalisis data berdasarkan jenis bencana dan tahun kejadian, sehingga dapat memantau dampak bencana

terhadap perumahan, mengidentifikasi wilayah terdampak, dan mendukung pengambilan keputusan terkait bantuan dan rekonstruksi.

Berikut beberapa filter yang bisa digunakan di dashboard tersebut:

1. **Jenis Kejadian:** Memungkinkan pengguna untuk memfilter data berdasarkan jenis bencana (Gempa, Tsunami, dll.).
2. **Year of Tahun:** Memungkinkan pengguna untuk menampilkan data berdasarkan tahun kejadian.
3. **Map total Persebaran Kerusakan Rumah:** Pengguna menklik provinsi mana yang akan dilihat maka visualisasi lain seperti data kerusakan rumah akan menyesuaikan terhadap provinsi mana yang akan dipilih.



Gambar 3 21 Dashboard Dampak Kejadian Bencana(Clustering)

Dashboard diatas menampilkan analisis klaster (clustering) dari dampak kejadian bencana, khususnya kerusakan rumah. Visualisasi data berupa scatter plot menunjukkan korelasi antara jumlah rumah rusak ringan dengan

persentase rumah rusak berat berdasarkan jenis bencananya, dengan setiap titik mewakili suatu wilayah dan dikelompokkan dalam 4 klaster berbeda berdasarkan warna.

### **Pengoperasian Dashboard yang Terintegrasi dengan Tableau Public**

Setelah dashboard berhasil diintegrasikan menggunakan Tableau Public, pengguna dapat mengoperasikan dashboard ini untuk menganalisis dan mendapatkan wawasan dari data yang telah disajikan. Berikut adalah langkah-langkah operasional beserta detail fitur yang tersedia:

#### 1. Akses Dashboard

Dashboard dapat diakses melalui URL yang telah diintegrasikan ke dalam situs web Gugus Mitigasi Lebak Selatan (GMLS).

Untuk mengaksesnya:

1. Langkah: Pengguna hanya perlu membuka browser, menyetikkan URL yang disediakan, atau mengklik tautan langsung yang terdapat di menu utama situs web GMLS.
2. Kebutuhan Teknis: Karena Tableau Public adalah platform berbasis web, akses ke dashboard memerlukan koneksi internet yang stabil. Hal ini memastikan data dan visualisasi yang interaktif dapat dimuat dengan lancar tanpa gangguan.

#### 2. Navigasi antar dashboard

Dashboard menyediakan berbagai visualisasi data yang dapat diakses dengan mudah melalui menu navigasi interaktif. Menu ini membantu pengguna untuk berpindah dari satu dashboard data ke dashboard lainnya. Contohnya dashboard yang ada di website gmls, yaitu:

1. Dashboard titik evakuasi situregen
2. Dashboard sensus penduduk situregen

### 3. Dashboard dampak kejadian bencana

Navigasi Tab: Pengguna cukup mengklik tab di bagian kanan pada website gmls untuk mengganti dashboard yang mau dilihat.

## 4.2 Kendala yang Ditemukan

Beberapa kendala yang ditemukan dalam masa magang ini:

1. Salah satu kendala utama yang dihadapi selama magang adalah kesulitan dalam mengatur waktu secara efektif. Mahasiswa sering kali merasa tertekan dengan banyaknya tugas yang harus diselesaikan dalam waktu terbatas, terutama dengan tenggat waktu yang ketat. Ketidakmampuan untuk mengelola waktu dengan baik menyebabkan keterlambatan dalam menyelesaikan pekerjaan dan memperburuk kualitas hasil yang dicapai. Hal ini bisa menjadi masalah yang signifikan, terutama ketika banyak tugas harus diselesaikan secara bersamaan, seperti pengembangan fitur web dan pemahaman tentang framework yang digunakan. Terlebih lagi, tekanan untuk memenuhi tenggat waktu seringkali mempengaruhi kualitas pekerjaan, karena mahasiswa terpaksa bekerja dalam kondisi terburu-buru. Tidak sepenuhnya memahami framework Next.js dan bahasa pemrograman JavaScript, yang keduanya sangat penting untuk menyelesaikan masalah
2. Selain manajemen waktu, kurangnya pemahaman yang mendalam tentang framework Next.js dan bahasa pemrograman JavaScript juga menjadi kendala besar. Kedua alat ini sangat penting dalam pengembangan website dan aplikasi berbasis web, namun mahasiswa merasa kesulitan dalam memahaminya dengan baik. Pemahaman yang tidak mendalam ini menyebabkan implementasi fitur menjadi lebih lambat dan hasilnya kurang optimal. Misalnya, ketika mahasiswa harus menggunakan Next.js untuk membangun aplikasi, kesulitan dalam memahami struktur framework atau cara kerja komponen-komponennya bisa membuat proses



pengembangan fitur menjadi lebih panjang dari yang seharusnya. Begitu pula dengan JavaScript, ketika tidak dipahami dengan baik, bisa mempengaruhi interaktivitas dan fungsionalitas aplikasi web yang sedang dikembangkan.

3. **Pengalokasian Waktu untuk Pembelajaran Luar Jam Kerja**  
Masalah lain yang muncul adalah pengalokasian waktu di luar jam kerja untuk mempelajari kedua bahasa pemrograman tersebut. Untuk mengatasi kesulitan dalam memahami Next.js dan JavaScript, mahasiswa diwajibkan untuk mempelajari bahasa-bahasa ini secara mandiri di luar waktu kerja yang sudah ditentukan. Hal ini mengurangi waktu yang tersedia untuk menyelesaikan tugas-tugas lainnya, seperti pengembangan fitur atau debugging aplikasi. Ketika waktu belajar yang diperlukan semakin banyak, mahasiswa semakin kesulitan mengatur keseimbangan antara tugas
4. **Keterbatasan waktu juga menjadi masalah besar, terutama ketika mahasiswa harus memenuhi tenggat waktu yang ketat dengan jumlah tugas yang banyak.** Tugas-tugas yang diberikan sering kali membutuhkan waktu yang lebih lama untuk diselesaikan, dan kombinasi dari tantangan dalam mempelajari teknologi baru serta mengelola waktu yang terbatas sering kali membuat mahasiswa merasa terbebani. Perasaan terbebani ini mempengaruhi kesehatan mental dan fisik mereka, menyebabkan kelelahan, serta menurunkan motivasi untuk terus belajar dan bekerja dengan efektif. Ditambah dengan tekanan tenggat waktu yang terus mendekat, masalah ini bisa membuat mahasiswa merasa frustrasi, dan dalam beberapa kasus, kualitas pekerjaan yang dihasilkan juga menjadi terganggu.
5. **Keterbatasan Data dalam Pengambilan Keputusan**  
Selain masalah teknis dan manajerial, keterbatasan data yang tersedia juga menjadi hambatan dalam pengembangan fitur. Seringkali, data yang dibutuhkan untuk menyelesaikan tugas pengembangan tidak tersedia atau

sangat terbatas, yang menyulitkan pengambilan keputusan yang tepat. Ketika bekerja pada proyek pengembangan website, data yang relevan untuk pengujian dan validasi aplikasi atau fitur tertentu sering kali sulit diakses, sehingga membuat proses pengembangan menjadi lebih lambat dan kurang akurat. Keterbatasan data ini tidak hanya mempengaruhi hasil kerja, tetapi juga mempersulit penyesuaian desain atau fungsionalitas aplikasi sesuai dengan kebutuhan pengguna atau pemangku kepentingan.

### **4.3 Solusi atas Kendala yang Ditemukan**

1. Mengajukan Pertanyaan dan Meminta Bantuan Untuk mengatasi masalah pemahaman teknis yang terkait dengan pengembangan website, mahasiswa dapat menyampaikan pertanyaan dan meminta bantuan kepada teman atau dosen yang lebih berpengalaman di tempat magang. Melalui diskusi dengan mereka, mahasiswa bisa mendapatkan pemahaman yang lebih mendalam mengenai pengembangan website, penggunaan framework Next.js, serta cara yang lebih efisien dalam menulis kode menggunakan JavaScript. Dengan bantuan ini, mahasiswa akan lebih cepat menemukan solusi atas masalah yang dihadapi dan mempercepat proses pengembangan fitur. Menghadiri sesi bimbingan dari dosen dan teman-teman pihak GMLS yang memberikan arahan dan saran praktis dalam pengembangan fitur, serta menjelaskan konsep-konsep teknis yang sulit dipahami.
2. Mengikuti Sesi Bimbingan dari Dosen, Sesi bimbingan yang diberikan oleh dosen dan teman-teman dari pihak GMLS sangat berguna untuk mendapatkan arahan lebih lanjut dalam pengembangan fitur. Dosen dan mentor dapat memberikan saran praktis dalam menyelesaikan masalah teknis, serta menjelaskan konsep-konsep sulit yang mungkin tidak dipahami dengan mudah. Dengan mengikuti sesi bimbingan ini secara rutin, mahasiswa bisa mendapatkan perspektif baru yang lebih jelas dan

praktis terkait dengan masalah yang dihadapi, sehingga dapat meningkatkan efisiensi dan kualitas pengembangan website. Mengaplikasikan pengetahuan yang diperoleh melalui latihan praktis dengan mengerjakan proyek kecil yang berfokus pada pengembangan website, guna meningkatkan pemahaman dan keterampilan teknis.

3. Memanfaatkan Sumber Daya Online untuk Belajar Mandiri Untuk mempercepat pemahaman tentang Next.js dan JavaScript, mahasiswa dapat memanfaatkan sumber daya online seperti tutorial, kursus daring, dan dokumentasi resmi. Kursus daring yang terstruktur dengan baik dapat membantu mahasiswa memahami konsep-konsep dasar dengan cara yang sistematis. Selain itu, tutorial video dan forum komunitas dapat memberikan wawasan tambahan dan memungkinkan mahasiswa untuk belajar dari pengalaman orang lain yang sudah lebih berpengalaman. Dengan memanfaatkan berbagai sumber daya ini, mahasiswa dapat mempercepat proses pembelajaran dan meningkatkan keterampilan teknis mereka.
4. Praktik Langsung dengan Proyek Kecil Agar pemahaman tentang teknologi menjadi lebih mendalam, mahasiswa bisa mengaplikasikan pengetahuan yang telah didapat melalui latihan praktis dengan mengerjakan proyek kecil yang berfokus pada pengembangan website. Melalui proyek kecil ini, mahasiswa dapat lebih memahami penerapan konsep-konsep yang telah dipelajari, serta belajar bagaimana mengatasi tantangan teknis yang muncul di dunia nyata. Setiap tantangan yang diselesaikan melalui proyek kecil ini akan memperkuat keterampilan teknis mahasiswa dan mempersiapkan mereka untuk tugas-tugas yang lebih besar dan kompleks di masa depan.
5. Membantu dengan Keterbatasan Data Melalui Relawan Dalam hal keterbatasan data, mahasiswa dapat bekerja sama dengan relawan dari desa Situregen yang datang untuk memberikan dukungan.

Relawan ini dapat membantu mengumpulkan data yang diperlukan untuk pengembangan aplikasi atau pengujian fitur, sehingga pengambilan keputusan bisa lebih tepat dan berbasis data yang valid. Dengan bantuan relawan, mahasiswa dapat memperoleh data yang lebih lengkap dan akurat, yang pada gilirannya mempercepat proses pengembangan dan meningkatkan kualitas hasil yang dicapai. Relawan juga dapat memberikan perspektif lokal yang berharga dalam menyesuaikan aplikasi dengan kebutuhan pengguna di daerah tersebut.