



KERJA PRAKTIK - IF184801

Pengembangan Dashboard Visualisasi Data untuk Aplikasi Conservana dengan Integrasi Google Spreadsheet

PT. Pasar Mikro Global

Graha Sentana, Lt. Dasar Jl. Buncit Raya No.2, RT.3/RW.2,
Duren Tiga, Kec. Pancoran, Kota Jakarta Selatan, Daerah Khusus
Ibukota Jakarta 12740

Periode: 1 Juli 2024 – 30 September 2024

Oleh:

Nadya Zuhria Amana 5025211058

Najma Ulya Agustina 5025211239

Pembimbing Departemen

Imam Kuswardayan, S.Kom., MT.

Pembimbing Lapangan

Eko Juniarto

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

[Halaman ini sengaja dikosongkan]

**LEMBAR
PENGESAHANKERJA
PRAKTIK**

**Pengembangan Dashboard Visualisasi Data untuk Aplikasi
Conservana dengan Integrasi Google Spreadsheet**

Nadya Zuhria Amana

5025211058

Najma Ulya Agustina

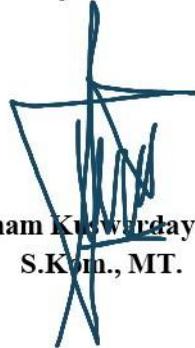
5025211239

Mengetahui,
Pembimbing Lapangan
Kerja Praktik



Eko Juniarto

Menyetujui,
Dosen Pembimbing
Kerja Praktik



**Imam Kuswardayan,
S.Kom., MT.**

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

LEMBAR PENGESAHANKERJA PRAKTIK.....	v
DAFTAR ISI	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xiii
ABSTRAK	xv
KATA PENGANTAR.....	xvii
BAB I PENDAHULUAN	19
1.1. Latar Belakang	19
1.2. Tujuan.....	20
1.3. Manfaat.....	20
1.4. Rumusan Masalah	20
1.5. Lokasi dan Waktu Kerja Praktik.....	20
1.6. Metodologi Kerja Praktik.....	21
1.6.1. Perumusan Masalah.....	21
1.6.2. Studi Literatur.....	21
1.6.3. Analisis dan Perancangan Sistem	1
1.6.4. Implementasi Sistem	1
1.6.5. Pengujian dan Evaluasi	1
1.7. Sistematika Laporan.....	1
1.7.1. Bab I Pendahuluan.....	1
1.7.2. Bab II Profil Perusahaan.....	2
1.7.3. Bab III Tinjauan Pustaka.....	2
1.7.4. Bab IV Analisis dan Perancangan Infrastruktur Sistem.....	2
1.7.5. Bab V Implementasi Sistem.....	2
1.7.6. Bab VI Pengujian dan Evaluasi.....	2
1.7.7. Bab VII Kesimpulan dan Saran	2
BAB II PROFIL PERUSAHAAN.....	3
2.1. PT. Pasar Mikro Global.....	3
2.2. Lokasi.....	3
BAB III TINJAUAN PUSTAKA	5

3.1.	Pemrograman Web.....	5
3.2.	React.Js	5
3.3.	Node.Js	6
3.4.	Chart.Js.....	6
3.5.	Google Spreads API	6
3.6.	CSS.....	7
3.7.	Visual Studio Code.....	7
3.8.	Web Server	8
BAB IV ANALISIS DAN PERANCANGAN INFRASTRUKTUR		
SISTEM		9
4.1	Analisis Sistem	9
4.1.1.	<i>Definisi Umum Aplikasi</i>	<i>9</i>
4.1.2.	<i>Analisis Kebutuhan.....</i>	<i>10</i>
A.	<i>Kebutuhan Fungsional</i>	<i>12</i>
B.	<i>Kebutuhan Non-Fungsional</i>	<i>13</i>
4.2.	Diagram Kasus Penggunaan.....	13
4.3.	Spesifikasi Diagram Kasus Pengguna Administrator.....	14
4.3.1.	<i>Melihat data chart/grafik</i>	<i>14</i>
4.3.2.	<i>Melihat detail chart/grafik</i>	<i>16</i>
4.4.	Perancangan Infrastruktur Sistem	20
4.4.1.	<i>Rancangan Backend.....</i>	<i>20</i>
4.4.2.	<i>Conceptual Data Model (CDM)</i>	<i>20</i>
4.4.3.	<i>Physical Data Model (PDM).....</i>	<i>21</i>
BAB V IMPLEMENTASI SISTEM		23
5.1.	Implementasi <i>Backend</i>	23
5.1.1.	<i>Struktur Folder pada Project Dashboard Conservana.....</i>	<i>23</i>
5.1.2.	<i>Models/user.js</i>	<i>24</i>
5.1.3.	<i>Routes/auth.js.....</i>	<i>26</i>
5.1.4.	<i>Routes/sheets.js</i>	<i>26</i>
5.1.5.	<i>Server.js</i>	<i>27</i>
5.2	Implementasi <i>Frontend</i>	29
5.2.1	<i>Halaman Dashboard.....</i>	<i>29</i>
5.2.2	<i>Halaman Login</i>	<i>31</i>
BAB VI PENGUJIAN DAN EVALUASI.....		35
1.1.	Tujuan Pengujian	35

1.2.	Kriteria Pengujian	35
1.3.	Skenario Pengujian	35
1.4.	Evaluasi Pengujian.....	36
BAB VII KESIMPULAN DAN SARAN		39
7.1.	Kesimpulan	39
7.2.	Saran	39
DAFTAR PUSTAKA		42
BIODATA PENULIS I.....		45
BIODATA PENULIS II.....		46

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 4.1 Arsitektur Aplikasi Dashboard Conservana.....	10
Gambar 4.2 Diagram Kasus Pengguna.....	14
Gambar 4.3 Melihat chart/grafik.....	16
Gambar 4.4 Melihat detail chart/grafik.....	18
Gambar 4.5 Melihat Perubahan data cahart otomatis.....	20
Gambar 4.6 Conceptual Data Model.....	21
Gambar 4.7 Physical Data Model.....	22
Gambar 5.1 Struktur Folder Backend Dashboard Conservana.....	24
Gambar 5.2 Models pada module user.js.....	25
Gambar 5.3 Routes pada module auth.js.....	26
Gambar 5.4 Routes ada module sheets.js.....	27
Gambar 5.5 Modul Server.js.....	28
Gambar 5.6 Halaman Dashboards.....	30
Gambar 5.7 Halaman Login Admin.....	31
Gambar 5.8 Screenflow Aplikasi Dashboard Conservana.....	32

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 4.1 Kebutuhan Fungsional	12
Tabel 4.2 Kebutuhan Non-Fungsional.....	13
Tabel 4.3 Menambahkan role pada user	15
Tabel 4.4 Melihat detail chart atau grafik.....	16
Tabel 4.5 Melihat list kelas.....	19
Tabel 6.1 Hasil Pengujian.....	36

[Halaman ini sengaja dikosongkan]

Pengembangan Dashboard Visualisasi Data untuk Aplikasi Conservana dengan Integrasi Google Spreadsheet

Nama Mahasiswa : Nadya Zuhria Amana
NRP : 5025211058
Departemen : Teknik Informatika
Nama Mahasiswa : Najma Ulya Agustina
NRP : 5025211239
Departemen : Teknik Informatika
Pembimbing : Imam Kuswardayan, S.Kom., MT.
Departemen :
Pembimbing Lapangan : Eko Juniarto

ABSTRAK

Aplikasi Dashboard Conservana dirancang untuk memfasilitasi pemantauan dan analisis data secara real-time bagi para penggunanya yaitu admin dan data analyst pada perusahaan PT. Pasar Mikro Global yang membutuhkan data tersebut. Untuk mendukung kebutuhan ini, dikembangkan sebuah Dashboard Visualisasi Data yang terintegrasi dengan Google Spreadsheet. Dashboard ini berfungsi sebagai portal yang menyediakan layanan untuk menampilkan data-data penting dari Google Spreadsheet dalam bentuk visualisasi yang interaktif dan mudah dipahami. Pengembangan dashboard ini menggunakan framework React.js serta teknologi front-end lainnya seperti JavaScript dan Chart.js, dengan dukungan Google Sheets API untuk pengambilan dan pemrosesan data secara otomatis. Dengan adanya dashboard ini, diharapkan proses pemantauan dan analisis data menjadi lebih efisien, akurat, dan dapat diakses oleh pengguna secara real-time, sehingga mendukung pengambilan keputusan yang lebih baik dalam pengelolaan data pada aplikasi Conservana.

Kata Kunci: Visualisai Data, Google Sheets API.

ABSTRACT

The Conservana Dashboard application is designed to facilitate real-time monitoring and data analysis for its users, namely administrators and data analysts at PT. Pasar Mikro Global, who require this data. To support these needs, a Data Visualization Dashboard integrated with Google Sheets has been developed. This dashboard serves as a portal that provides services to display important data from Google Sheets in the form of interactive and easy-to-understand visualizations. The development of this dashboard utilizes the React.js framework and other front-end technologies such as JavaScript and Chart.js, with the support of the Google Sheets API for automated data retrieval and processing. With this dashboard, it is expected that the process of monitoring and analyzing data will become more efficient, accurate, and accessible to users in real time, thereby supporting better decision-making in data management within the Conservana application.

Keywords: Data Visualization, Google Sheets API.

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT atas penyertaan dan karunia-Nya sehingga penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Pengembangan Dashboard Visualisasi Data untuk Aplikasi Conservana dengan Integrasi Google Spreadsheet.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang tua penulis;
2. Bapak Imam Kuswardayan, S.Kom., M.T. selaku dosen pembimbing kerja praktik;
3. Bapak Yoseph Budianto selaku pembimbing lapangan selama kerja praktik berlangsung;
4. Bapak Eko Juniarto dan Mas R. Wahyu Triessaputra selaku pembimbing teknis yang memberikan arahan teknis; dan
5. Seluruh staff Pasar Mikro yang terlibat dalam kegiatan pembuatan Dashboard Conservana yang tidak dapat penuliskan satu-persatu.

Surabaya, Oktober 2024

Nadya, Ulya

[Halaman ini sengaja dikosongkan]

BAB I PENDAHULUAN

1.1. Latar Belakang

Conservana merupakan salah satu klien Pasar Mikro yang bergerak di bidang agrikultur dan komoditas. Dalam upaya memahami kebutuhan dan tantangan yang dihadapi Conservana, PT. Pasar Mikro Global telah melakukan berbagai survei terkait distribusi komoditas, sebaran usia responden, penggunaan layanan perbankan digital, dan aspek lainnya. Namun, hasil survei tersebut umumnya didokumentasikan dalam bentuk spreadsheet yang kurang efektif untuk menyajikan data secara jelas dan interaktif.

Untuk mengatasi keterbatasan ini, diperlukan sebuah platform yang dapat menampilkan data dalam bentuk visualisasi interaktif, seperti grafik atau chart, sehingga memudahkan pengguna dalam memahami informasi dan melakukan analisis data yang lebih mendalam. Pengembangan dashboard ini menggunakan framework React.js untuk antarmuka pengguna yang dinamis, serta Chart.js untuk menghasilkan visualisasi data yang menarik dan informatif. Integrasi dengan Google Spreadsheet dilakukan melalui Google Sheets API, yang memungkinkan pengambilan data secara real-time dan otomatis dari spreadsheet yang telah digunakan sebelumnya oleh pengguna. Dengan integrasi ini, data yang ditampilkan pada dashboard akan selalu up-to-date, tanpa perlu intervensi manual dari pengguna.

Selain itu, untuk menjaga keamanan dan privasi data, platform ini perlu dilengkapi dengan fitur login yang hanya mengizinkan pengguna yang berwenang untuk mengaksesnya. Dengan adanya dashboard visualisasi data yang terintegrasi dengan Google Spreadsheet dan dilengkapi sistem otentikasi, pengelolaan data dapat dilakukan secara lebih terstruktur, aman, dan efisien, sehingga mendukung proses pengambilan keputusan yang lebih baik.

1.2. Tujuan

Tujuan dari kerja praktik ini adalah untuk mengembangkan dashboard visualisasi data interaktif yang terintegrasi dengan Google Spreadsheet, sehingga memudahkan admin dan data analyst dalam memantau dan menganalisis data secara real-time.

1.3. Manfaat

Manfaat yang diperoleh dengan adanya Dashboard Conservana *antara lain*:

1. Mempermudah admin data dan data analyst dalam memantau data hasil survey.
2. Memperkuat Keamanan Akses Data: Dengan adanya fitur login yang hanya mengizinkan pengguna yang berwenang untuk mengakses dashboard, keamanan data terjaga dan risiko akses tidak sah dapat diminimalisir.

1.4. Rumusan Masalah

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana cara menyajikan data hasil survei dalam format visualisasi yang interaktif agar memudahkan admin data dan data analyst dalam memantau dan menganalisis informasi yang relevan
2. Bagaimana memastikan keamanan akses data pada dashboard agar hanya pengguna yang berwenang yang dapat mengakses informasi penting tersebut, sehingga risiko akses tidak sah dapat diminimalisir

1.5. Lokasi dan Waktu Kerja Praktik

Kerja praktik ini dilaksanakan pada waktu dan tempat sebagai berikut :

Lokasi : Kantor PT. Pasar Mikro Global
Alamat : Graha Sentana, l. Buncit Raya No.2, RT.3/RW.2, Duren Tiga,
Kec. Pancoran, Kota Jakarta Selatan, Daerah Khusus Ibukota
Jakarta 12740

Waktu: 1 Juli 2024 – 30 September 2024

Kegiatan dilakukan secara offline atau WFO (Work from Office) selama 2 bulan dan remote selama satu bulan. Selama kerja praktik secara remote, laporan harian mengenai progress tetap dilakukan secara intensif setiap harinya.

1.6. Metodologi Kerja Praktik

Metodologi dalam pembuatan buku kerja praktik meliputi:

1.6.1. Perumusan Masalah

Dalam proses menentukan masalah, diberikan penjelasan kebutuhan dari aplikasi yang akan dibuat serta alur penggunaan dalam menggunakan system. Penjelasan mengenai hal ini dijelaskan oleh Head of Technology atau Kepala bagian Teknologi, Bapak Eko Juniarto. Dari penjelasan beliau dihasilkan catatan-catatan penting mengenai gambaran dashboard yang akan di integrasikan dengan google spreadsheet. Dengan begitu dapat diputuskan untuk membuat dashboard berbasis web dengan menggunakan bahasa pemrograman React.js untuk kebutuhan frontend dan Node.js, Express.js sebagai backend serta menggunakan Google Spreads API untuk mengambil data dari Spreadsheet dan Chart.js untuk visualisasi data. Hal ini dikarenakan bahasa pemrograman web jenis ini telah umum dan populer digunakan.

1.6.2. Studi Literatur

Setelah ditentukan cara pengambilan data, bahasa pemrograman, serta tools tambahan yang akan digunakan, dilakukan studi literatur mengenai cara implementasinya dalam membangun sistem sesuai yang dibutuhkan. Pada tahap ini dilakukan proses pencarian, pembelajaran, pengumpulan dan

pemahaman informasi serta literatur yang berkaitan untuk membantu dalam implementasi aplikasi ini. Informasi bisa didapat dari internet untuk istilah-istilah umum yang digunakan dalam mengimplementasikan suatu sistem informasi dan juga dokumentasi yang berkaitan dengan React.Js, Node.Js, Chart.Js, Google Spreadsheet API.

1.6.3. Analisis dan Perancangan Sistem

Langkah ini meliputi penjelasan awal tentang sistem. Bagaimana cara kerja sistem dengan skenario tertentu. Dari penjelasan awal telah didapatkan beberapa kebutuhan fungsional dan non-fungsional secara garis besar. Kemudian dilanjutkan dengan memperjelas dan menspesifikkan kebutuhan- kebutuhan tersebut. Maka dibuat sebuah diagram kasus penggunaan yang mewakili skenario- skenario untuk penggunaan dashboard Conservana berbasis Web yang di Integrasikan dengan Google Spreadsheets.

1.6.4. Implementasi Sistem

Implementasi sistem didasarkan pada perancangan dan analisis sebelumnya. Kasus penggunaan dan penentuan tools juga turut mendasari pengimplementasian sistem ini. Pada tahap ini setidaknya ada dua pekerjaan utama yang dilakukan, yakni desain web pada tampilan atau front-end dan juga desain fungsi-fungsi yang bekerja dalam sistem atau dikenal sebagai back-end. Pengerjaan dilakukan dengan progres setiap hari, dengan setiap harinya menargetkan perkembangan dari hari sebelumnya. Progres penyelesaian dashboard terus dipantau oleh Bapak Eko Juniarto selaku Head of Technology yang memimpin setiap peloporan progress masing masing staff technology.

1.6.5. Pengujian dan Evaluasi

Pengujian dilakukan dengan menguji fitur-fitur yang telah dibuat. Kesesuaian sistem dengan kebutuhan akan menentukan keberhasilan dalam pengujian. Hal ini akan menghasilkan hasil evaluasi apakah sistem sudah sesuai dengan tujuan dan kebutuhan atau belum.

1.7. Sistematika Laporan

1.7.1. Bab I Pendahuluan

Bab I berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

1.7.2. **Bab II Profil Perusahaan**

Bab II menjelaskan secara rinci tentang profil perusahaan tempat dilaksanakannya kerja praktik yaitu Kantor PT. Pasar Mikro Global

1.7.3. **Bab III Tinjauan Pustaka**

Bab III menjelaskan mengenai tinjauan pustakan dan literatur yang digunakan.

1.7.4. **Bab IV Analisis dan Perancangan Infrastruktur Sistem**

Bab IV menjelaskan hasil analisis terhadap kebutuhan sistem serta rencana untuk implementasi.

1.7.5. **Bab V Implementasi Sistem**

Bab V berisi uraian tahap - tahap yang dilakukan untuk proses implementasi.

1.7.6. **Bab VI Pengujian dan Evaluasi**

Bab VI berisi hasil uji coba dan evaluasi dari *website* yang telah dikembangkan selama pelaksanaan kerja praktik.

1.7.7. **Bab VII Kesimpulan dan Saran**

Bab VII berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

BAB II PROFIL PERUSAHAAN

2.1. PT. Pasar Mikro Global

PasarMIKRO adalah perusahaan rintisan teknologi Indonesia yang berfokus pada sektor pertanian dan komoditas. Didirikan pada tahun 2020 oleh tiga pendiri, yaitu Wong Lok Dien, Demetrius Edo Djayaputra, dan Hugo Verwayen, perusahaan ini beroperasi dengan nama PT. Pasar Mikro Global dan dikenal dengan merek PasarMIKRO. PasarMIKRO adalah perusahaan swasta yang berkantor pusat di Graha Sentana, Jl. Warung Buncit Raya No. 2, Duren Tiga, Pancoran, Jakarta Selatan, DKI Jakarta, 12760.

PasarMIKRO memiliki sekitar 1900 pengguna aktif dan 260 petani. PasarMikro memiliki model bisnis pembiayaan dengan skema pembayaran tempo untuk para petani, karena mereka biasanya tidak menerima penerimaan pembayaran hasil panen tepat waktu. Sementara dengan dengan tengkulak atau pengepul mereka menjalankan skema bagi hasil.

2.2. Lokasi

Graha Sentana, Jl. Warung Buncit Raya No. 2, Duren Tiga, Pancoran, Jakarta Selatan, DKI Jakarta, 12760.

[Halaman ini sengaja dikosongkan]

BAB III TINJAUAN PUSTAKA

3.1. Pemrograman Web

Aplikasi berbasis *website* merupakan suatu perangkat lunak yang dapat diakses melalui *website* sehingga tidak memerlukan instalasi langsung di perangkat pengguna. Kelebihan dari penggunaan aplikasi berbasis *website* adalah pengguna dapat mengakses dengan mudah melalui URL atau alamat web aplikasi. Aplikasi ini beroperasi secara online sehingga memerlukan internet untuk dapat mengaksesnya. Keunggulan lain yang diperoleh dari penggunaan aplikasi berbasis *website* yaitu kemudahan akses dari berbagai perangkat seperti dekstop, laptop, tablet, maupun *smartphone*[1].

Sedangkan pemrograman *website* adalah proses pembuatan aplikasi berbasis *website* yang meliputi proses menulis, mencoba, memeriksa, memperbaiki (debug), serta memelihara kode untuk membuat sebuah program. Tujuan dari pemrograman *website* yaitu untuk membuat *website* yang memuat suatu program untuk melakukan suatu pekerjaan manusia atau proses sesuai dengan keinginan pemrogram [2].

3.2. React.Js

ReactJS adalah library JavaScript populer yang dibuat oleh Facebook yang digunakan dalam proses pengembangan aplikasi mobile dan web. untuk memfasilitasi pembuatan daripada komponen antarmuka yang interaktif, stateful, serta mudah untuk digunakan ulang. ReactJS sangatlah cocok digunakan untuk rendering antarmuka yang kompleks dengan performa tinggi.

Dalam pengembangan, ReactJS memiliki berbagai macam kemampuan seperti dapat digunakan untuk membuat animasi suatu objek dengan efek transisi, menjalankan permainan di web browser yang sepenuhnya diprogram dengan menggunakan React. Selain itu, ReactJS juga mampu melakukan validasi form secara real-time saat pengguna mengisi data pada form tersebut [3].

3.3. Node.js

Node.js merupakan platform yang diciptakan secara khusus untuk membantu pengembangan aplikasi berbasis web. Walau demikian, Node.js bukan bahasa pemrograman yang baru, tetapi *runtime environment* atau interpreter untuk menjalankan bahasa pemrograman JavaScript sebagai kebutuhan *back-end developing*. Adapun Node.js dibangun dengan JavaScript V8 milik Google sehingga Node.js memiliki performa yang tinggi. Node.js juga memiliki library sendiri. Dengan begitu, pengguna tak perlu menggunakan webserver NGINX maupun Apache. Selain itu, Node.js juga mengusung model *event-driven* dan *non-blocking I/O* nya sehingga lebih mampu menangani banyak proses secara bersamaan dibandingkan platform bersifat *thread-based-networking* [4].

3.4. Chart.js

Chart.js adalah sebuah pustaka JavaScript open-source yang digunakan untuk membuat grafik interaktif dan responsif pada halaman web. Dengan Chart.js, kamu bisa membuat berbagai jenis grafik seperti: Line Chart, Bar Chart, Pie Chart, Radar Chart dan masih banyak lagi. Chart.js sangat mudah digunakan dan cocok untuk visualisasi data sederhana. Hanya perlu memasukkan data dan opsi konfigurasi ke dalam kode JavaScript, lalu Chart.js akan menggambarkan grafik yang diinginkan pada elemen `<canvas>` di halaman HTML. Fitur utama chart.js adalah responsif, interaktif, customizable, dan ringan[5].

3.5. Google Spreads API

Google Sheets API adalah sebuah layanan yang disediakan oleh Google untuk memungkinkan aplikasi atau pengguna berinteraksi dengan Google Sheets secara programatik. Dengan menggunakan API ini, akan bisa mengakses, memodifikasi, dan memanipulasi data di dalam spreadsheet Google Sheets melalui kode program, tanpa harus membuka dokumen Google Sheets secara langsung[6].

Google Spreads API menyediakan fitur utama seperti akses

data yang memungkinkan untuk membaca data dari sel, rentang sel, atau seluruh sheet; memperbarui data dengan menulis atau mengubah data di sel atau rentang sel tertentu; manipulasi struktur yang mencakup menambahkan, menghapus, atau memodifikasi sheet, baris, dan kolom; serta formatting yang memungkinkan perubahan format sel seperti warna, gaya font, dan perataan. API ini juga mendukung otorisasi berbasis OAuth2 untuk akses yang aman terhadap spreadsheet pengguna.

3.6. CSS

Cascading Style Sheets (CSS) merupakan bahasa pemrograman yang biasa digunakan dalam pengembangan halaman web. Menggunakan CSS pengembang web dapat mengendalikan secara detail tampilan dan format dari berbagai komponen dalam suatu website, sehingga menciptakan struktur yang terorganisir dan seragam. CSS memberikan kontrol terhadap berbagai aspek seperti ukuran gambar, warna *body* teks, warna tabel, ukuran border, warna border, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, margin kiri/kanan/atas/bawah, dan parameter lainnya yang memungkinkan pengembang untuk menciptakan halaman web yang menarik [7].

CSS tidak hanya memiliki kemampuan untuk mengatur tampilan halaman secara keseluruhan tetapi juga memungkinkan pengembang untuk menyesuaikan format dokumen dengan lebih fleksibel. Dengan menggunakan CSS, sebuah website dapat menampilkan halaman yang sama dengan berbagai format yang berbeda sesuai dengan preferensi pengguna atau tuntutan desain tertentu. CSS tidak hanya meningkatkan kualitas visual dari suatu website, tetapi juga memberikan fleksibilitas yang diperlukan dalam pengembangan web [8].

3.7. Visual Studio Code

Visual Studio Code adalah *source code editor* yang dikembangkan Microsoft yang bisa digunakan pada berbagai *platform* (Linux, Mac OS dan Microsoft). Visual Studio Code juga dapat digunakan sebagai *editor source code* berbagai bahasa pemrograman dari PHP, Java, Python, Javascript, Node.js, C/C++,

Go, C# sampai *editor* untuk *script web* seperti HTML dan CSS. Visual Studio Code juga telah terintegrasi ke GitHub [9].

3.8. Web Server

Web Server adalah perangkat lunak yang berfungsi sebagai penerima permintaan yang dikirimkan melalui *browser* kemudian memberikan tanggapan permintaan dalam bentuk halaman situs *website* atau lebih umumnya dalam dokumen HTML [10].

BAB IV ANALISIS DAN PERANCANGAN INFRASTRUKTUR SISTEM

4.1 Analisis Sistem

Tahapan dalam membangun infrastruktur *website* Dashboard Conservana yaitu analisis dari infrastruktur sistem yang akan dibangun. Hal tersebut dijelaskan menjadi tiga bagian, yaitu definisi umum aplikasi, arsitektur aplikasi dan analisis kebutuhan.

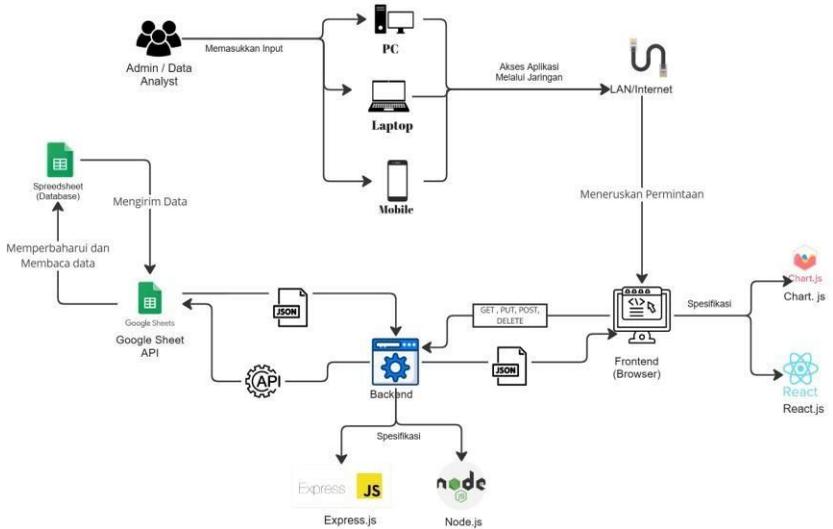
4.1.1. Definisi Umum Aplikasi

Secara Umum, Dashboard Visualisasi Data untuk Aplikasi Conservana dengan Integrasi Google Spreadsheet adalah sebuah antarmuka berbasis web yang dirancang untuk menampilkan data penting secara visual dalam bentuk grafik dan tabel, dengan menggunakan data yang diperoleh dari Google Spreadsheet. Dashboard ini membantu admin dalam memantau, menganalisis, dan mengambil keputusan berdasarkan informasi yang tersaji. Melalui integrasi dengan Google Spreadsheet, data yang ada di spreadsheet secara otomatis diperbarui di dalam dashboard, memungkinkan admin melihat data terbaru tanpa perlu melakukan sinkronisasi manual.

View Admin dalam dashboard ini mencakup fitur utama seperti:

- Visualisasi Data: Grafik interaktif yang memudahkan admin memahami tren data, seperti distribusi responden perkategori umur, gender, pendidikan dan juga distribusi komoditas.
- Akses Data Real-Time: Data dari Google Spreadsheet disajikan secara real-time, memastikan admin selalu mengakses informasi terbaru.

4.1.2. Arsitektur Sistem



Gambar 4. 1 Arsitektur Aplikasi Dashboard Conservana

Arsitektur sistem dashboard Conservana mengelola data survei melalui integrasi Google Sheets, React.js, Chart.js, Express.js, dan Node.js. Data survei disimpan dalam spreadsheet, diakses melalui backend (Node.js dan Express.js), lalu divisualisasikan menggunakan Chart.js di frontend yang dibangun dengan React.js. Admin mengakses aplikasi melalui browser di PC Client yang terhubung ke jaringan lokal (LAN) untuk memantau dan mengelola data survei secara real-time. Router menghubungkan perangkat ke jaringan, sementara database menyimpan data yang diolah untuk mendukung pengelolaan dan analisis yang efisien.

4.1.3. Analisis Kebutuhan

Dashboard Conservana memiliki fungsi-fungsi yang dirancang khusus untuk memenuhi kebutuhan dalam memantau, menganalisis, dan menyajikan data secara efisien bagi admin.

Dashboard ini terintegrasi dengan Google Spreadsheet sebagai sumber data utama, sehingga mampu memperbarui informasi secara otomatis. Kebutuhan-kebutuhan yang diidentifikasi terbagi ke dalam dua jenis, yakni kebutuhan fungsional dan kebutuhan non-fungsional.

A. *Kebutuhan Fungsional*

Kebutuhan fungsional menjelaskan alur kerja dari Dashboard Conservana ini. Kebutuhan fungsional dari Dashboard Conservana ditunjukkan pada Tabel 4.1.

Tabel 4.1 Kebutuhan Fungsional

No	Kebutuhan Fungsional
F01	Sistem dapat terintegrasi dengan Google Spreadsheet untuk menarik data hasil survey
F02	Sistem dapat menampilkan data dalam bentuk grafik dan tabel yang mudah dipahami.
F03	Sistem dapat memberikan akses hanya kepada admin untuk mengelola dashboard
F04	Sistem dapat memperbarui data secara real-time dari Google Spreadsheet tanpa penyegaran manual

B. Kebutuhan Non-Fungsional

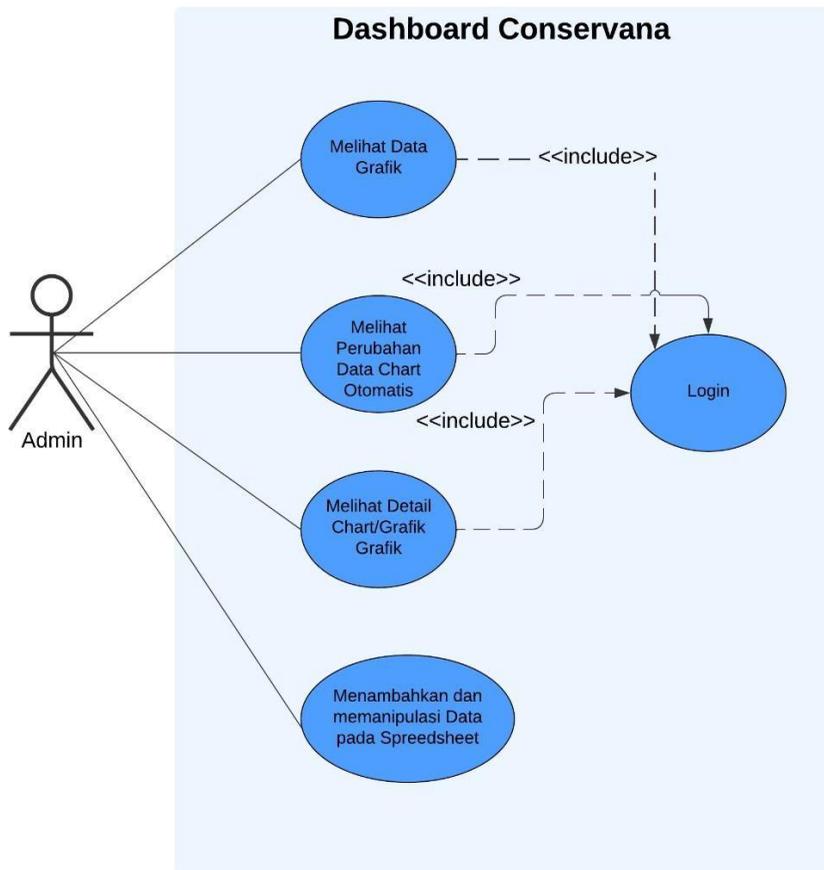
Kebutuhan non-fungsional menjelaskan karakteristik dan batasan dari Dashboard Conservana. Kebutuhan non-fungsional dari Dashboard Conservana ditunjukkan pada Tabel 4.2.

Tabel 4.2 Kebutuhan Non-Fungsional

Kode	Deskripsi
NF01	Sistem dapat diakses oleh pengguna selama 24 jam per hari.
NF02	Sistem dapat diakses pada semua jenis <i>web browser</i> .
NF03	Sistem memiliki UI/UX yang mudah untuk dipahami dan sesuai mekanisme yang dibutuhkan admin.
NF04	Sistem dapat memastikan data terlindungi dari akses yang tidakberwenang.

4.2. Diagram Kasus Penggunaan

Pembahasan dengan pembimbing lapangan terkait fitur Dashboard Conservana menghasilkan diagram kasus penggunaan (*Use Case Diagram*). Diagram ini dirancang agar fitur dan alur aplikasi dapat mudah dipahami dan terpenuhi dengan optimal. Diagram Aplikasi Dashboard Conservana dapat dilihat pada Gambar 4.1. Pengguna dalam diagram tersebut adalah Admin.



Gambar 4. 2 Diagram kasus Pengguna

4.3. Spesifikasi Diagram Kasus Pengguna Administrator

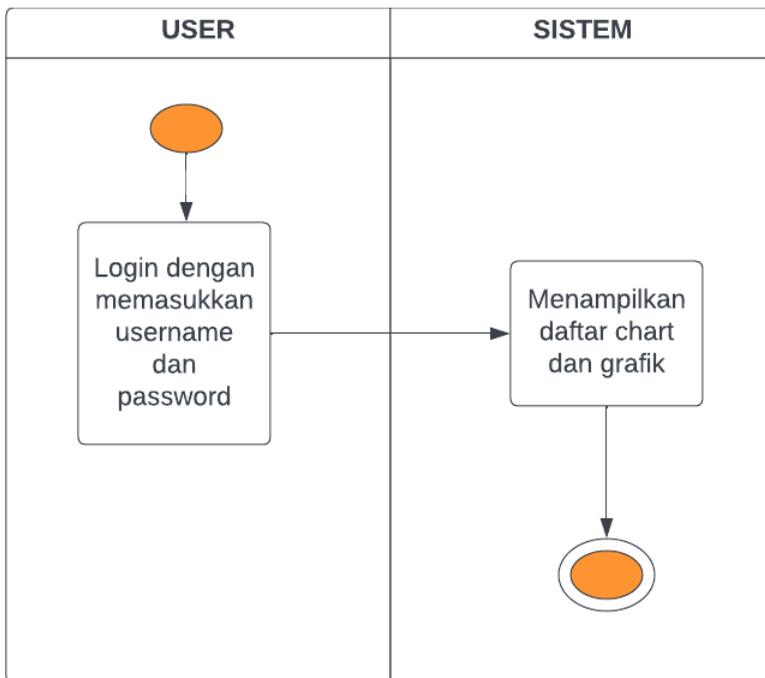
4.3.1. *Melihat data chart/grafik*

Fitur ini berfungsi untuk memberikan hak akses kepada admin untuk melihat daftar grafik yang ada dalam dashboard. Berbagai kasus terkait fitur ini dan bagaimana mekanisme sistem dari fitur ini berjalan digambarkan pada **Tabel 4.3**.

Tabel 4.3 Menambahkan role pada user

Nama	Melihat Data Chart/Grafik
Kode	F01
Aktor	Admin
Deskripsi	Admin melihat visualisai data dalam bentuk chart
Relasi	-
Kondisi Awal	Belum dapat melihat seluruh daftar chart yang ada
Kondisi Akhir	Sudah dapat melihat seluruh data yang diambil dari spreadsheet dalam bentuk chart
Alur Kejadian Normal	
Aktor	Sistem
1.1 Login Dashboard Conservana	1.1 Menampilkan halaman <i>list</i> Chart berisi data yang diambil dari spreadsheet
Alur Kejadian Alternatif	
Aktor	Sistem

Penggambaran diagram aktivitas terkait kasus melihat data chart atau grafik yang tersedia sesuai dengan **Gambar 4.2**.



Gambar 4. 3 Melihat chart atau grafik

4.3.2. *Melihat detail chart/grafik*

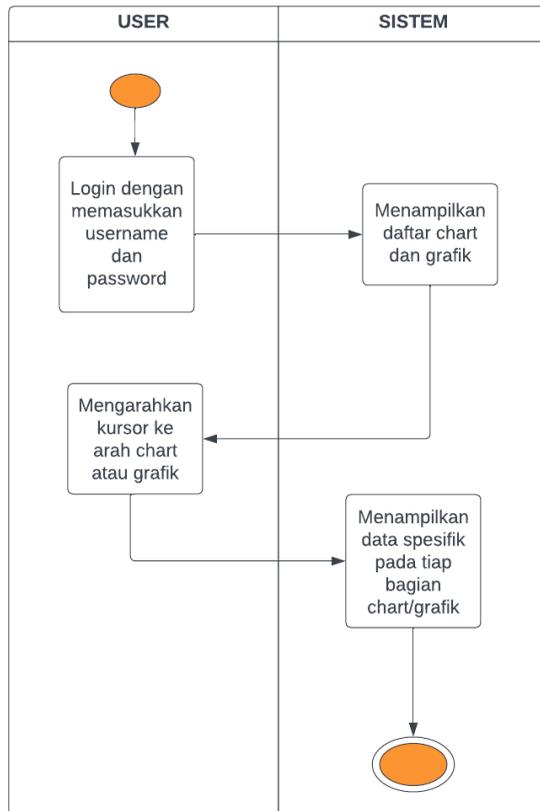
Fitur ini memiliki fungsi untuk melihat isi detail dari chart/grafik yang ada. Untuk alur dari proses akan digambarkan pada **Tabel 4.4.**

Tabel 4.4 Melihat detail chart atau grafik

Nama	Melihat Detail Chart/Grafik
Kode	F02
Aktor	Admin

Deskripsi	Admin melihat detail data dalam chart yang berisi jumlah responden pada setiap bagian dalam chart/grafik.
Relasi	-
Kondisi Awal	Terdapat keseluruhan chart/grafik tetapi belum terlihat detail data pada setiap chart/grafik
Kondisi Akhir	Berhasil melihat detail data pada setiap chart/grafik
Alur Kejadian Normal	
Aktor	Sistem
1.1 Melakukan login aplikasi	1.1 Menampilkan halaman yang berisi keseluruhan chart/grafik
2.1 Mengarahkan kursor ke arah chart/grafik yang tersedia	2.1 Menampilkan detail data pada setiap chart/grafik
Alur Kejadian Alternatif	
Aktor	Sistem
-	-

Penggambaran diagram aktivitas terkait kasus menghapus user pada role yang tersedia sesuai dengan **Gambar 4.3**.



Gambar 4. 4 Melihat detail chart/grafik

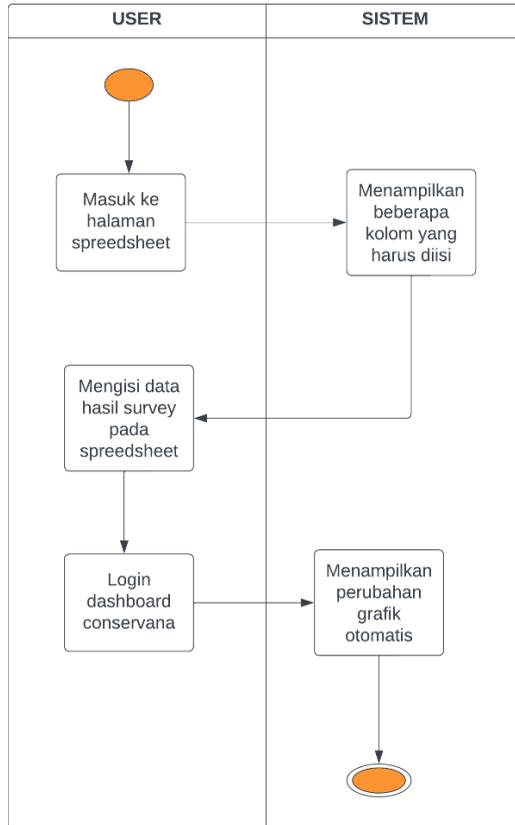
4.3.3. Melihat Perubahan Data Chart/Grafik Otomatis

Fitur ini memiliki fungsi untuk menampilkan list dari kelas yang telah dibuat atau kelas yang tersedia pada SIM-Mentoring. Berbagai kasus terkait fitur ini dan bagaimana mekanisme sistem dari fitur ini berjalan digambarkan pada **Tabel 4.5**.

Tabel 4.5 Melihat list kelas

Nama		Lihat Perubahan Data Chart Otomatis	
Kode		F03	
Aktor		Admin	
Deskripsi		Admin dapat melihat perubahan terhadap tampilan chart ketika dilakukan penambahan data/pengurangan data pada spreadsheet sumber.	
Relasi		-	
Kondisi Awal		Ingin melihat perubahan data chart/grafik yang akan ditampilkan	
Kondisi Akhir		Mengetahui perubahan data pada setiap chart/grafik yang ditambah atau dikurangi datanya dari data sumber.	
Alur Kejadian Normal			
Aktor		Sistem	
1.1 Membuka dashboard		1.1 Menampilkan halaman yang berisi tampilan chart yang telah berubah	
Alur Kejadian Alternatif			
Aktor		Sistem	
-		-	

Penggambaran diagram aktivitas terkait kasus melihat list kelas yang tersedia sesuai dengan Gambar 4.4



Gambar 4. 5 Melihat perubahan data chart otomatis

4.4. Perancangan Infrastruktur Sistem

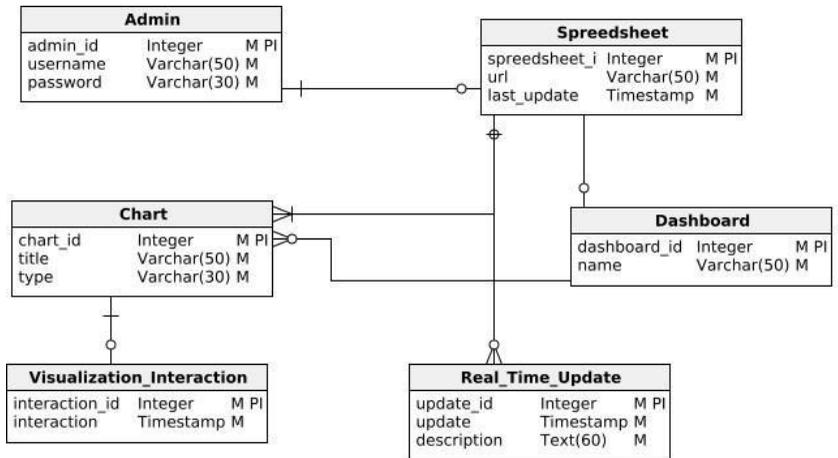
4.4.1. Rancangan *Backend*

Dashboard Conservana merupakan dibangun dengan beberapa sub modul, seperti modul login dan modul dashboard.

4.4.2. Conceptual Data Model (CDM)

Context Data Model (CDM) untuk Dashboard Conservana memberikan gambaran umum tentang entitas utama yang terlibat dan hubungan antara entitas tersebut. CDM ini dirancang untuk menangkap

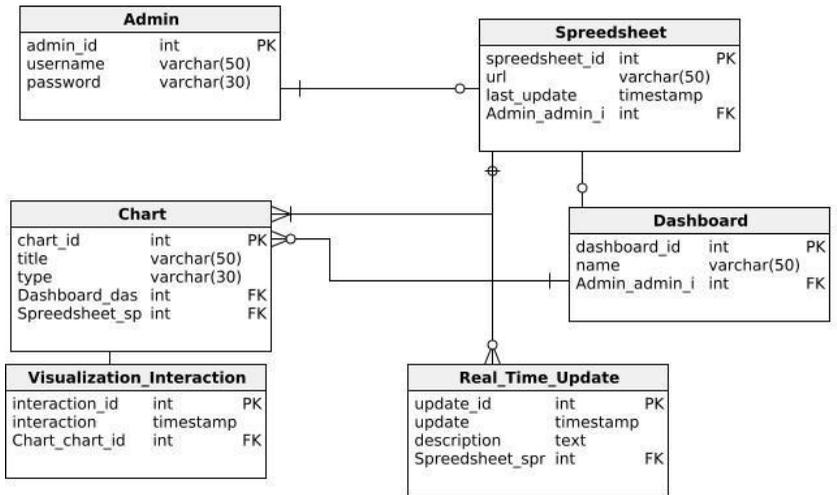
kebutuhan fungsional dan skenario pengguna.



Gambar 4. 6 Conceptual Data Model

4.4.3. Physical Data Model (PDM)

Physical Data Model (PDM) untuk Dashboard Conservana merinci struktur fisik dari database, termasuk tabel, kolom, tipe data, dan hubungan antar tabel. PDM ini dirancang untuk mengoptimalkan penyimpanan dan akses data



Gambar 4. 7 Physical Data Model

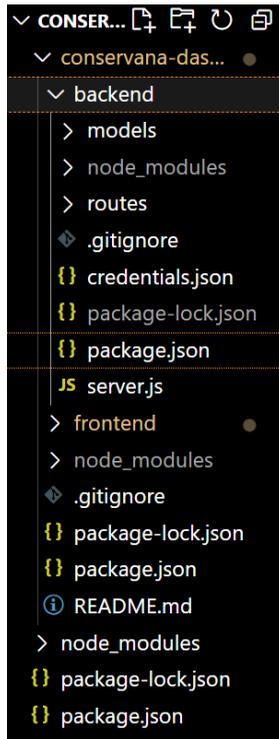
BAB V IMPLEMENTASI SISTEM

5.1. Implementasi *Backend*

Implementasi Backend menggunakan framework Express.js yang berjalan pada Node.js. Backend bertanggung jawab untuk menyediakan API untuk pengambilan dan pengelolaan data dari Google Spreadsheet menggunakan layanan Google API. Selain itu, backend mengatur autentikasi menggunakan hashing password dengan bcrypt dan menyediakan API untuk manipulasi data.

5.1.1. Struktur Folder pada Project Dashboard Conservana

Struktur folder yang digunakan pada *website* ini terdiri dari beberapa model presentation. Berikut struktur folder pada *presentation*, dengan layer utama *Model*, *Routes* dan *Server.js* dibantu dengan node modules..



Gambar 5. 1 Struktur Folder Backend Dashboard Conservana

5.1.2. *Models/user.js*

File ini bertugas mendefinisikan struktur data atau model untuk pengguna dalam sistem. Pada implementasi ini, file tersebut digunakan untuk mengelola data pengguna seperti nama pengguna (*username*) dan kata sandi (*password*) yang telah di-*hash* menggunakan bcrypt. Struktur model ini menjadi referensi utama untuk operasi terkait pengguna, seperti validasi data pengguna saat proses login.

```
backend > models > user.js > ...
1 import React, { useState, useEffect } from 'react';
2
3 Tabnine | Edit | Test | Explain | Document | Ask
4 function User() {
5   const [user, setUser] = useState(null);
6
7   // Simulasi mengambil data pengguna dari API atau localStorage
8   useEffect(() => {
9     const loggedInUser = localStorage.getItem('user');
10    if (loggedInUser) {
11      setUser(JSON.parse(loggedInUser));
12    } else {
13      window.location.href = '/'; // Redirect ke halaman login jika user tidak ada di localStorage
14    }
15  }, []);
16
17  const logout = () => {
18    localStorage.removeItem('user'); // Menghapus data user dari localStorage
19    setUser(null); // Menghapus data user dari state
20    window.location.href = '/'; // Redirect ke halaman login
21  };
22
23  if (!user) {
24    return <p>Loading...</p>; // Tampilkan loading
25  }
26
27  return (
28    <div>
29      <h2>Welcome, {user.username}</h2>
30      <p>Email: {user.email}</p>
31      <button onClick={logout}>Logout</button>
32    </div>
33  );
34
35  export default User;
36
```

Gambar 5. 2 Models pada module user.js

5.1.3. *Routes/auth.js*

File ini berisi rute yang menangani proses autentikasi pengguna. Fungsi utamanya adalah menerima data *username* dan *password* dari klien, memvalidasi keberadaan *username*, dan mencocokkan *password* yang diberikan dengan *password hash* yang disimpan menggunakan *bcrypt*. Jika validasi berhasil, file ini memberikan respons berhasil login, sedangkan jika gagal, file ini memberikan pesan kesalahan yang relevan. Rute ini bertujuan untuk memastikan hanya pengguna terdaftar yang dapat mengakses sistem.

```
backend > routes > auth.js > ...
1  const express = require('express');
2  const bcrypt = require('bcryptjs');
3  const router = express.Router();
4
5  const users = [
6    {
7      username: 'admin',
8      passwordHash: '$2a$10$K1W7Qn6bEyyG9.tEJmJ6xuViA2zXe1gHF9n8MwWzEzjVBBkNcE02G6' // Hash dari 'admin123'
9    }
10 ];
11
12 router.post('/login', async (req, res) => {
13   const { username, password } = req.body;
14   const user = users.find((u) => u.username === username);
15
16   if (!user) {
17     return res.status(400).json({ message: 'Invalid username' });
18   }
19
20   const isMatch = await bcrypt.compare(password, user.passwordHash);
21
22   if (!isMatch) {
23     return res.status(400).json({ message: 'Invalid password' });
24   }
25
26   // Kirim respons dengan data user
27   res.status(200).json({ message: 'Login successful', user: { username } });
28 });
29
30 module.exports = router;
31
```

Gambar 5. 3 Routes pada module *auth.js*

5.1.4. *Routes/sheets.js*

File ini bertanggung jawab untuk mengelola integrasi dengan Google Sheets menggunakan Google API. Rute dalam file ini menyediakan dua fungsi utama: (1) mengambil data dari spreadsheet dan mengembalikannya dalam format JSON untuk digunakan di frontend, dan (2) menambahkan data baru ke

spreadsheet berdasarkan permintaan dari klien. File ini menggunakan *Service Account* untuk autentikasi ke Google API dan mendukung pengelolaan data secara dinamis.

```
1 const express = require('express');
2 const { google } = require('googleapis');
3 const keys = require('./credentials.json');
4 const router = express.Router();
5
6 // Buat client JWT menggunakan kunci Service Account
7 const client = new google.auth.JWT(
8   keys.client_email,
9   null,
10  keys.private_key,
11  ["https://www.googleapis.com/auth/spreadsheets"]
12 );
13
14 // ID spreadsheet dan rentang yang digunakan
15 const spreadsheetId = "1IsEGUM6hMta1_9iGu7TD1L8Ifg8iVjXkkgWfGfC";
16 const range = "Form responses !1A1:AM11";
17
18 // Route untuk mengambil data dari Google Sheets
19 tabName[0] adalah [document] dan
20 router.get('/data', async (req, res) => {
21   try {
22     await client.authorize();
23     const sheets = google.sheets({ version: 'v4', auth: client });
24
25     const response = await sheets.spreadsheets.values.get({
26       spreadsheetId,
27       range,
28     });
29     res.status(200).json({ data: response.data.values });
30   } catch (error) {
31     console.error('Error accessing spreadsheet:', error);
32     res.status(500).json({ error: 'Error accessing spreadsheet' });
33   }
34 });
35
36 // Route untuk menambahkan data ke Google Sheets
37 tabName[0] adalah [document] dan
38 router.post('/data', async (req, res) => {
39   try {
40     await client.authorize();
41     const sheets = google.sheets({ version: 'v4', auth: client });
42
43     const { values } = req.body; // Data yang akan ditambahkan, pastikan
44     const resource = {
45       values: [values],
46     };
47
48     await sheets.spreadsheets.values.append({
49       spreadsheetId,
50       range,
51       valueInputOption: 'RAW',
52       resource,
53     });
54     res.status(201).json({ message: 'Data added successfully' });
55   } catch (error) {
56     console.error('Error adding data to spreadsheet:', error);
57     res.status(500).json({ error: 'Error adding data to spreadsheet' });
58   }
59 });
60
61 module.exports = router;
62
```

Gambar 5. 4 Routes pada module sheets.js

5.1.5. Server.js

File utama ini berfungsi sebagai pengendali aplikasi backend. Dalam file ini, server Express.js diinisialisasi dan rute dari folder routes dihubungkan ke jalur API yang relevan. File ini juga menangani konfigurasi middleware seperti *CORS* untuk mengizinkan permintaan lintas-origin dan menyediakan rute default untuk memastikan server

berjalan dengan baik. Dengan kata lain, file ini adalah titik pusat pengaturan server yang mengatur alur permintaan dari klien ke rute-rute yang tersedia.

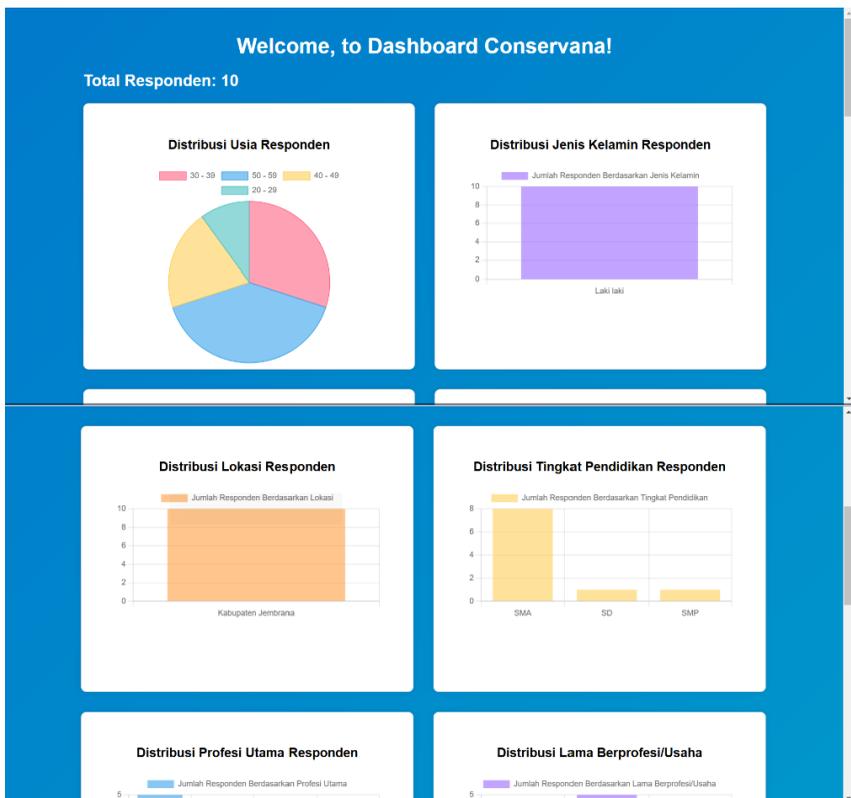
```
backend > server > appget@bmgkita: callback
1  const express = require('express');
2  const cors = require('cors'); // Tambahkan ini
3  const { google } = require('googleapis'); // File credentials Google API, pastikan sudah benar
4  const keys = require('./credentials.json');
5
6  const app = express();
7  const PORT = 5000; // Backend berjalan di port 5000
8
9  app.use(cors()); // Gunakan middleware ini untuk mengizinkan permintaan dari origin yang berbeda
10
11 // Inisialisasi client JWT untuk Google API
12 const client = new google.auth.JWT(
13   keys.client_email,
14   null,
15   keys.private_key,
16   ['https://www.googleapis.com/auth/spreadsheets.readonly'] // Pastikan Anda memiliki izin yang sesuai
17 );
18
19 // ID dan rentang dari spreadsheet
20 const spreadsheetId = '1IsEGUAGh0Mle1_9iGw7TDILBtIfqD1V3X8qgNKf6ic'; // Ganti dengan ID spreadsheet Anda
21 const range = 'Form responses !1A1:AH1!'; // Rentang data yang digunakan
22
23 // Route untuk mengambil data dari Google Sheets
24 app.get('/sheets/data', async (req, res) => {
25   try {
26     await client.authorize();
27     const sheets = google.sheets({version: 'v4', auth: client});
28
29     const response = await sheets.spreadsheets.values.get({
30       spreadsheetId,
31       range,
32     });
33
34     res.json(response.data); // Kirim data sebagai response dalam format JSON
35   } catch (error) {
36     console.error('Error accessing Google Sheets:', error);
37     res.status(500).json({ error: 'Error accessing Google Sheets' });
38   }
39 });
40 // Route default untuk root
41 app.get('/', (req, res) => {
42   res.send('Backend is running!');
43 });
44
45 app.listen(PORT, () => {
46   console.log('Backend server is running on http://localhost:${PORT}');
47 });
48
49
```

Gambar 5. 5 Module pada Server.js

5.2 Implementasi *Frontend*

Frontend pada sistem ini dirancang untuk menyediakan antarmuka pengguna yang interaktif dan responsif, dibangun menggunakan React.js. Struktur frontend disimpan di direktori `src/` yang memuat komponen-komponen utama aplikasi seperti halaman dashboard, halaman login, dan halaman visualisasi data. Antarmuka pengguna didesain untuk memberikan pengalaman navigasi yang intuitif bagi pengguna dengan peran yang berbeda

5.2.1 Halaman Dashboard

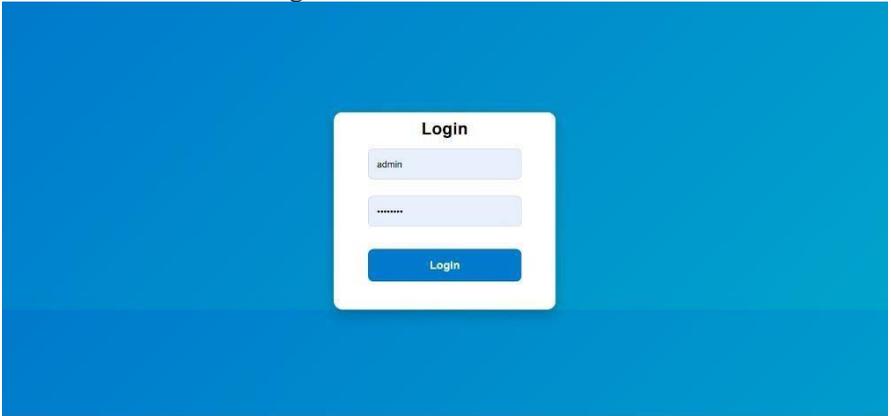




Gambar 5. 6 Halaman Dashboard

Error! Reference source not found. Halaman ini merupakan halaman utama yang dilihat oleh pengguna setelah berhasil login. Pada halaman dashboard, pengguna dapat mengakses berbagai fitur melalui sidebar navigasi yang tersedia. Antarmuka ini menampilkan visualisasi data secara real-time menggunakan Chart.js, serta informasi penting seperti statistik atau pemberitahuan yang relevan dengan pengguna.

5.2.2 Halaman Login

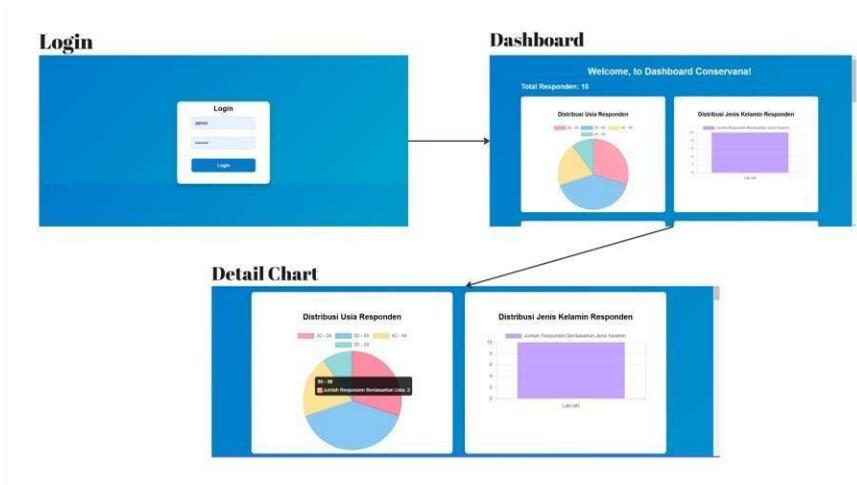


Gambar 5. 7 Halaman login admin

Error! Reference source not found. Halaman login merupakan pintu masuk utama ke dalam sistem, di mana pengguna harus memasukkan kredensial mereka untuk mengakses fitur-fitur yang sesuai dengan perannya. Halaman ini juga dilengkapi dengan validasi form untuk memastikan keamanan dan keakuratan data yang di input.

Setiap halaman dalam sistem ini dirancang dengan fokus pada responsivitas dan aksesibilitas, memastikan bahwa aplikasi dapat digunakan dengan baik pada berbagai perangkat, termasuk desktop, tablet, dan smartphone.

5.2.3 Screen Flow Aplikasi



Gambar 5. 8 Screen Flow Aplikasi Dashboard Conservana

Screen flow aplikasi ini dimulai dari halaman login, di mana pengguna (admin) harus memasukkan username dan password untuk dapat mengakses aplikasi. Halaman login memiliki form input yang sederhana dengan dua kolom utama, yaitu untuk username dan password, serta tombol "Login" untuk memverifikasi kredensial. Setelah berhasil login, admin diarahkan ke halaman dashboard, sementara jika login gagal, sistem akan menampilkan pesan kesalahan. Di halaman dashboard, fungsi utama adalah menampilkan data visual dalam bentuk *chart* yang diambil langsung secara real-time dari Google Sheets menggunakan Google Sheets API. Data ini diperbarui otomatis tanpa perlu penyegaran manual, sehingga admin selalu melihat data terkini. Dashboard dilengkapi dengan fitur *interactive chart*, di mana ketika kursor diarahkan ke elemen tertentu dalam chart, detail data seperti angka atau persentase akan muncul dalam bentuk tooltip. Chart juga dirancang menggunakan Chart.js, yang mendukung berbagai tipe visualisasi seperti bar chart, pie chart, atau line chart. Selain itu, admin dapat menavigasi lebih lanjut (opsional) untuk melihat detail data dengan mengklik elemen tertentu pada chart. Dashboard juga menyediakan komponen tambahan seperti header dengan nama aplikasi dan opsi logout, serta area chart utama yang memvisualisasikan data secara interaktif. Proses kerja dimulai dengan memanggil data dari Google Sheets API menggunakan backend yang dibangun dengan *Express.js* dan *Node.js*,

kemudian data ditampilkan di frontend menggunakan *React.js* dan *Chart.js*. Dengan struktur ini, aplikasi memastikan pengalaman yang lancar dan informatif bagi admin untuk memantau data real-time.

BAB VI PENGUJIAN DAN EVALUASI

1.1. Tujuan Pengujian

Pengujian dilakukan pada sistem Dashboard Conservana untuk menguji kemampuan arsitektur backend dan frontend dalam melayani berbagai fungsionalitas yang telah diimplementasikan. Pengujian ini bertujuan untuk memastikan sistem berjalan dengan baik, sesuai dengan kebutuhan pengguna, dan mendukung semua logika bisnis yang telah dirancang.

1.2. Kriteria Pengujian

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memperhatikan beberapa hasil yang diharapkan berikut:

- a. Kemampuan server backend untuk melayani tampilan antarmuka aplikasi.
- b. Kemampuan server backend dalam menangani query data dan mengelola data melalui API Google Sheets.
- c. Kemampuan server dalam menyimpan data yang diinputkan oleh pengguna.
- d. Kemampuan server dalam melayani upload data oleh pengguna melalui antarmuka aplikasi.
- e. Kemampuan sistem untuk mengelola autentikasi pengguna.
- f. Kemampuan sistem untuk menampilkan data dan visualisasi dashboard secara real-time.

1.3. Skenario Pengujian

Pengujian dilakukan dengan menjalankan skenario uji pada setiap fitur sistem dengan langkah-langkah berikut:

1. Autentikasi Pengguna: Administrator dapat masuk ke sistem menggunakan username dan password.
2. Pengelolaan Data Google Sheets:
 - Pengguna dapat melihat data yang diambil dari Google Sheets.
 - Administrator dapat menambahkan data baru ke Google Sheets.
3. Visualisasi Data Dashboard: Pengguna dapat melihat grafik yang divisualisasikan menggunakan Chart.js.

4. Manajemen Data:

- Administrator dapat menambah dan menghapus data tertentu melalui backend.
- Sistem dapat menampilkan data secara real-time sesuai perubahan di Google Sheets.

5. Respons Server:

- Server mampu menangani permintaan API dengan cepat meskipun diakses oleh banyak pengguna secara bersamaan.

Untuk mengevaluasi pengalaman pengguna, dilakukan survei terhadap 30 pengguna yang terbagi menjadi 20 pengguna biasa, 8 administrator, dan 2 pengelola utama. Survei dilakukan dengan memberikan skor pada rentang 1-5, di mana (1) Sangat Tidak Puas, (2) Tidak Puas, (3) Cukup Puas, (4) Puas, dan (5) Sangat Puas.

Hasil survei menunjukkan tingkat kepuasan rata-rata sebagai berikut:

1. Pengguna Umum: Skor rata-rata kepuasan adalah 3.8.
2. Administrator: Skor rata-rata kepuasan adalah 4.0.
3. Pengelola Utama: Skor rata-rata kepuasan adalah 4.2.

Berdasarkan hasil pengujian dan survei, sistem Dashboard Conservana dinilai memenuhi harapan pengguna dalam hal fungsionalitas dan kemudahan penggunaan. Beberapa umpan balik telah dicatat untuk perbaikan di iterasi pengembangan selanjutnya.

1.4. Evaluasi Pengujian

Hasil pengujian menunjukkan bahwa sistem Dashboard Conservana berhasil memenuhi semua kriteria pengujian yang telah ditetapkan. Berikut adalah tabel hasil pengujian sistem:

Tabel 0.1 Hasil Pengujian

Kriteria Pengujian	Hasil Pengujian
Server dapat melayani tampilan aplikasi	Terpenuhi

Server dapat menyediakan data yang diminta aplikasi	Terpenuhi
Server dapat menyimpan data yang diinputkan	Terpenuhi
Server dapat melayani upload file	Terpenuhi
Sistem dapat menampilkan data secara real-time	Terpenuhi
Sistem dapat mengelola autentikasi pengguna	Terpenuhi

[Halaman ini sengaja dikosongkan]

BAB VII

KESIMPULAN DAN SARAN

7.1. Kesimpulan

Berdasarkan hasil pengembangan, implementasi, pengujian, dan evaluasi sistem Dashboard Conservana, dapat disimpulkan beberapa poin berikut:

1. Sistem berhasil dikembangkan dengan menggunakan arsitektur backend berbasis Node.js dan Express.js yang terintegrasi dengan Google Sheets melalui API. Frontend dibangun menggunakan React.js untuk menyediakan antarmuka pengguna yang responsif.
2. Sistem mampu memenuhi kebutuhan utama pengguna, seperti autentikasi, pengambilan dan visualisasi data real-time menggunakan Chart.js, serta pengelolaan data melalui dashboard administrasi.
3. Hasil pengujian menunjukkan bahwa server mampu menangani permintaan API, menyimpan data, dan melakukan visualisasi dengan baik, bahkan pada kondisi beban tinggi.
4. Berdasarkan survei pengguna, sistem memperoleh skor kepuasan rata-rata yang menunjukkan bahwa sistem telah cukup memenuhi kebutuhan pengguna, dengan skor tertinggi pada aspek kemudahan akses data dan visualisasi informasi.
5. Sistem ini berhasil memberikan solusi untuk manajemen data yang terintegrasi, efisien, dan mudah digunakan oleh pengguna dengan berbagai peran.

7.2. Saran

Untuk pengembangan lebih lanjut, berikut adalah beberapa saran yang dapat diterapkan:

1. Sistem perlu dilengkapi dengan lapisan keamanan tambahan, seperti implementasi JSON Web Token (JWT) untuk autentikasi pengguna, serta enkripsi data sensitif untuk meningkatkan keamanan aplikasi.
2. Meskipun sistem sudah berjalan dengan baik, optimisasi query API dan proses pengambilan data dari Google Sheets dapat ditingkatkan untuk mempersingkat waktu respons, terutama pada

data berukuran besar.

3. Disarankan untuk menambahkan fitur notifikasi otomatis kepada pengguna, baik dalam bentuk email maupun push notification, untuk memberikan pembaruan data atau pengingat penting.
4. Meningkatkan desain antarmuka agar lebih responsif di berbagai perangkat, termasuk smartphone dan tablet, sehingga pengalaman pengguna menjadi lebih baik.
5. Melakukan survei kepuasan secara berkala untuk mengidentifikasi kebutuhan pengguna dan memperbaiki sistem berdasarkan umpan balik yang diberikan.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] Sianipar, R. H. (2015). *HTML 5 dan CSS 3: Belajar dari Kasus* (Vol. 1). Penerbit INFORMATIKA.
- [2] Alvian, 2019. *Panduan Awal Belajar Pemrograman Web Dalam 10 Menit*. Diakses pada 21 November 2021 dari: <https://sis.binus.ac.id/2019/02/25/hubungan-dan-perbedaan-javascript-html-css-jquery-dan-php-di-dalam-web-development/>, Januari 2021.
- [3] M, V. A., & Sonpatki, P. (2016). *ReactJS by Example: Building Modern Web Applications with React*. <https://doi.org/10.1016/j.jelekin.2006.06.006>
- [4] K&C, n.d. *Node.js development - an introduction to the popular JS runtime environment*. [online] Kruschecompany.com. Tersedia di: <https://kruschecompany.com/node-js-development-introduction/> [Diakses 6 Desember 2024].
- [5] Chart.js (2024) *Chart.js Documentation*. Available at: <https://www.chartjs.org> (Accessed: 6 December 2024).
- [6] Coefficient.io. (2024) *How to Use Google Sheets API: Complete Guide for 2024*. Available at: <https://coefficient.io> (Accessed: 6 December 2024).
- [7] W3Schools 2024, *CSS Introduction*, W3Schools. Tersedia pada: https://www.w3schools.com/css/css_intro.asp [Diakses 6 Desember 2024].
- [8] DEV Community 2024, *CSS (Cascading Style Sheets): Styling and Layout of Web Pages*, DEV Community. Tersedia pada: <https://dev.to> [Diakses 6 Desember 2024].
- [9] Wijaya, Ketut Krisna, 2016. *Visual Studio Code: Aplikasi Editor Kode dari Microsoft untuk Windows, Linux, dan OS X*. Diakses pada 21 November 2021 dari: <https://id.techinasia.com/visual-studio-code-editor-kode-microsoft/>, Januari 2021.
- [10] Yasin, 2018. *Apa Itu Web Server dan Fungsinya?*. Diakses pada 21 November 2021 dari:

<https://www.niagahoster.co.id/blog/web-server-adalah/>,
Januari 2021.

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS I

Nama : Najma Ulya Agustina
Tempat, Tanggal Lahir : Blitar, 30 Agustus 2001
Jenis Kelamin : Perempuan
Telepon : 085890442150
Email : agt.najmaul4477@gmail.com

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC , ITS
Angkatan : 2021
Semester : 7 (Tujuh)

BIODATA PENULIS II

Nama : Nadya Zuhria Amana
Tempat, Tanggal Lahir : Ponorogo, 03 Desember 2002
Jenis Kelamin : Perempuan
Telepon : 089604362868
Email : nadyazuhria@gmail.com

AKADEMIS

Kuliah : Departemen Teknik Informatika –
: FTEIC , ITS
Angkatan : 2021
Semester : 7 (Tujuh)