

SKRIPSI

VISUALISASI DATA SENSOR DI WIRELESS SENSOR NETWORK



Dimas Kurniawan

NPM: 6181901019

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2023

UNDERGRADUATE THESIS

**SENSOR DATA VISUALIZATION IN WIRELESS SENSOR
NETWORK**



Dimas Kurniawan

NPM: 6181901019

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2023**

LEMBAR PENGESAHAN

VISUALISASI DATA SENSOR DI WIRELESS SENSOR NETWORK

Dimas Kurniawan

NPM: 6181901019

Bandung, 4 Juli 2023

Menyetuju,

Pembimbing

Elisati Hulu, M.T.

Ketua Tim Penguji

Anggota Tim Penguji

Chandra Wijaya, M.T.

Gede Karya, M.T.

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

VISUALISASI DATA SENSOR DI WIRELESS SENSOR NETWORK

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 4 Juli 2023



Dimas Kurniawan
NPM: 6181901019

ABSTRAK

Wireless Sensor Network (WSN) adalah sekumpulan *node* sensor yang dihubungkan dalam jaringan komputer dengan memanfaatkan teknologi nirkabel. Setiap node sensor memiliki kemampuan memantau kondisi seperti suhu, suara, getaran, kelembapan, dan lain sebagainya. Nilai pengukuran yang didapatkan setiap node dikumpulkan dalam sebuah *base station*. Kemudian oleh *base station* akan dikirimkan ke aplikasi yang membutuhkannya.

Visualisasi data merupakan tampilan berupa grafik atau bentuk tampilan lainnya yang berfungsi untuk memberikan suatu informasi dari data agar lebih mudah dipahami serta dapat memberikan *insight* lebih. Visualisasi data dapat membantu pengguna untuk memahami data dengan memberikan rangkuman visual informasi yang lebih mudah untuk dipahami. Dalam visualisasi data untuk memberikan *insight* yang sesuai dan akurat maka dibutuhkan kesesuaian antara data yang diukur oleh sensor, kebutuhan pengguna, serta jenis visualisasi data. Salah satu tantangan masalah yang ada pada pembuatan visualisasi data adalah pada kebutuhan pengguna yang beragam seperti adanya kebutuhan *real-time*, kebutuhan untuk melihat fluktuasi, kebutuhan untuk melihat statistik, dan lain sebagainya.

Pembuatan perangkat lunak untuk menampilkan visualisasi data berhasil dibuat sehingga perangkat lunak berhasil menampilkan visualisasi data sesuai dengan kebutuhan dan ketersediaan data. Pengguna memilih sendiri kesesuaian antara kebutuhannya dengan jenis visualisasi.

Kata-kata kunci: Wireless Sensor Network, WSN, Visualisasi data, Preon32, Virtenio, API, HTTP, Plotly, Retrofit, FastApi, Bootstrap

ABSTRACT

Wireless Sensor Network (WSN) is a collection of sensor nodes connected in a computer network using wireless technology. Each sensor node has the capability to monitor conditions such as temperature, sound, vibration, humidity, and so on. The measurement values obtained by each node are collected at a base station. The base station then transmits them to the applications that require the data.

Data visualization is a display in the form of graphs or other visual formats that aim to provide information from the data in an easily understandable manner and provide additional insights. Data visualization can help users understand the data by providing a visually summarized information that is easier to comprehend. In order to provide suitable and accurate insights, data visualization needs to align the measured sensor data, user requirements, and the type of data visualization. One of the challenges in data visualization is the diverse user needs, such as real-time requirements, the need to observe fluctuations, the need to view statistics, and so on.

The development of software for displaying data visualization has been successful. The software is able to display data visualization according to the requirements and availability of data. Users can choose the suitability between their needs and the type of visualization.

Keywords: Wireless Sensor Network, WSN, Data Visualization, Preon32, Virtenio, API, HTTP, Plotly, Retrofit, FastApi, Bootstrap

Skripsi ini saya persembahkan kepada ilmu pengetahuan juga para pihak yang sekiranya membutuhkan dan mampu membaca skripsi ini hingga tuntas :)

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan yang Maha Esa, atas berkat dan penyertaan-Nya penulis mampu menyelesaikan skripsi ini. Skripsi ini disusun untuk memenuhi syarat kelulusan di program studi Teknik Informatika UNPAR.

Penulis juga menyadari bahwa penyusunan skripsi ini tidak akan berhasil tanpa adanya dukungan dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada pihak-pihak yang membantu dan mendukung penggerjaan skripsi ini, yaitu sebagai berikut.

1. Kepada kedua orang tua dan kakak penulis yang selalu memberikan dukungan, doa, dan semangat dalam perkuliahan dan proses penggerjaan skripsi.
2. Kepada Bapak Elisati Hulu, S.T, M.T. selaku dosen pembimbing yang telah membimbing penulis dengan baik dengan memberikan nasihat, ilmu, pengalaman, motivasi, komentar, dan *tag-tag* pada grup Line sehingga penulis “terpaksa” mengerjakan skripsi ini yang berakibat skripsi ini dapat selesai dengan “sangat” tepat waktu. Tidak lupa juga penulis ucapan terima kasih atas pinjaman sensor Preon32.
3. Kepada Bapak Chandra Wijaya, S.T, M.T dan Bapak Gede Karya, S.T, M.T selaku penguji yang telah memberikan masukan kepada skripsi ini agar menjadi lebih baik.
4. Kepada Bapak Kritopher David Harjono, S.T, M.T. yang telah memberikan rancangan awal skema basis data yang diimplementasikan pada skripsi ini.
5. Kepada Geraldi Akira Surya selaku rekan dengan dosen pembimbing yang sama. Penulis ucapan terima kasih atas diskusi yang selama ini dilakukan.
6. Kepada kelompok manajemen proyek penulis yang telah mengizinkan penulis menggunakan tema dan sumber daya pada tugas akhir mata kuliah manajemen proyek.
7. Kepada kakak-kakak tingkat seerbimbungan yang sudah mengunggah kode-kode untuk Preon32 di Github sehingga penulis dapat lebih mudah mengerti kode tanpa perlu banyak menerjemahkan dokumentasi pabrik yang berbahasa Jerman.
8. Kepada Bapak Chandra Wijaya, S.T, M.T yang telah meminjamkan perangkat Raspberry Pi dan juga telah memberikan kesempatan serta bimbingan kepada penulis pada lomba dari Huawei dan Cisco.
9. Kepada Kepala Lab Bapak Raymond Chandra, S.T, M.T. dan para admin yang telah membantu penulis dengan memberikan banyak saran dan mendukung selama penggerjaan skripsi.
10. Kepada rekan-rekan digrup Discord Gabut Club dan S L yang sudah membantu, mendukung, dan menemani penulis semasa kuliah dan penggerjaan skripsi.
11. Kepada seluruh pihak yang penulis tidak dapat cantumkan satu persatu demi keberlangsungan dan ukuran kata pengantar skripsi ini agar sekiranya tidak lebih panjang dari isi skripsi ini. Terima kasih penulis ucapan telah mendukung, membantu dan mempermudah selama masa kuliah dan penyelesaian skripsi ini.

Akhir kata sekaligus sebagai penutup penulis berterima kasih kepada para pembaca dokumen skripsi ini. Penulis berharap skripsi ini dapat memberikan manfaat yang nyata bagi pihak-pihak yang kuat membacanya hingga akhir.

Bandung, Juli 2023

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xxi
DAFTAR TABEL	xxv
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	2
2 LANDASAN TEORI	5
2.1 Visualisasi Data	5
2.1.1 Kaidah Visualisasi Data	5
2.1.2 Jenis-jenis Visualisasi Data	6
2.1.3 Jenis-Jenis Visualisasi Data Grafik	6
2.1.4 Aplikasi Visualisasi Data pada WSN	30
2.2 Wireless Sensor Network	32
2.2.1 <i>Node</i> Sensor	32
2.2.2 Protokol <i>Stack</i> WSN	33
2.2.3 Standar Protokol IEEE 802.15.4 dan Zigbee	34
2.2.4 Sistem Operasi Pada <i>Wireless Sensor Network</i> (WSN)	35
2.2.5 Protokol Komunikasi	35
2.2.6 Topologi <i>Wireless Sensor Network</i> (WSN)	36
2.3 Preon32	39
2.3.1 Spesifikasi Jenis-Jenis Sensor Modul Preon32	40
2.3.2 PreonVM	40
2.4 HTTP	41
2.4.1 HTTP Methods	41
2.4.2 Kode Status	41
2.4.3 HTTP Messages	42
2.5 Application Programming Interface	42
2.6 Library dan Framework	42
2.6.1 Bootstrap	42
2.6.2 Retrofit	42
2.6.3 Plotly	43
2.6.4 Fast API dan Unicorn	43
2.6.5 Laravel	43

2.6.6	Fetch	43
2.7	Bahasa Pemrograman	44
2.7.1	Bahasa Pemrograman Java	44
2.7.2	Bahasa Pemrograman Python	44
3	ANALISIS	45
3.1	Deskripsi Perangkat Lunak	45
3.1.1	Analisis Desain Sistem	46
3.1.2	Analisis Fungsi Perangkat Lunak	46
3.1.3	Use Case Diagram dan Use Case Skenario Website	46
3.1.4	<i>Wireless Sensor Network (WSN)</i>	49
3.1.5	API	52
3.1.6	Basis Data	53
3.2	Analisis Tujuan Pembuatan Visualisasi Data Sensor	56
3.2.1	Tujuan Pemantauan (<i>Monitoring</i>)	56
3.2.2	Tujuan Melihat Data Lampau	57
3.2.3	Tujuan Melihat Statistik	57
3.3	Format Pertukaran Data	58
3.3.1	Analisa Format Data <i>Node Sensor</i> ke <i>Base Station</i>	58
3.3.2	Analisa Format Data <i>Base Station</i> ke Server	59
3.3.3	Analisa Format Dari <i>Base station</i> ke <i>Node Sensor</i>	59
3.3.4	Analisa Format Dari <i>Node Sensor</i> ke <i>Base station</i>	60
3.4	Analisis Masalah Pada WSN	60
3.4.1	Penempatan <i>Node Sensor</i>	60
3.4.2	Preon32 Tidak Terdapat Baterai khusus untuk Mengaktifkan Modul RTC	62
4	PERANCANGAN	63
4.1	Perancangan Antarmuka	63
4.1.1	Diagram <i>Sequence</i>	63
4.1.2	<i>Mock-up</i>	67
4.1.3	Tata Letak Visualisasi Data Antara Jenis Data	69
4.2	Perancangan <i>Application Programming Interface (API)</i>	71
4.3	Perancangan Perangkat Lunak pada WSN	74
4.3.1	<i>Node Sensor</i> pada Preon32	74
4.3.2	<i>Base Station</i> pada Preon32	76
4.3.3	<i>Base Station</i> pada Komputer	77
4.3.4	<i>Node Sensor</i> pada Komputer	80
4.3.5	Perancangan Sinkronisasi Waktu dan Interval	80
5	IMPLEMENTASI DAN PENGUJIAN	81
5.1	Implementasi	81
5.1.1	Lingkungan Perangkat Keras	81
5.1.2	Lingkungan Perangkat Lunak	81
5.1.3	Implementasi tampilan	82
5.1.4	Implementasi API	86
5.2	Pengujian	96
5.2.1	Pengujian Fungsionalitas	96
5.2.2	Pengujian Eksperimental	101
5.2.3	Kesimpulan Hasil Eksperimen	105
6	KESIMPULAN DAN SARAN	107
6.1	Kesimpulan	107

6.2 Saran	107
DAFTAR REFERENSI	109
A KODE PROGRAM NODE SENSOR PADA PREON32	111
B KODE PROGRAM BASE STATION PADA PREON32	121
C KODE PROGRAM BASE STATION PADA PC	129
D KODE PROGRAM NODE SENSOR PADA PC	135
E KODE PROGRAM API	137
F KODE PROGRAM WEBSITE	155

DAFTAR GAMBAR

2.1	Gambar Ilustrasi Visualisasi Data Dalam Bentuk Tulisan Sederhana	6
2.2	Analogi Diagram Titik (<i>Dot Plot</i>)	7
2.3	Analogi Diagram Batang (<i>Bar Chart</i> atau Diagram Kolom <i>Column Chart</i>)	8
2.4	Analogi <i>Floating Bar</i> atau (<i>Gantt Chart</i>)	8
2.5	Analogi <i>Pixelated Bar Chart</i>	9
2.6	Analogi <i>Histogram</i>	10
2.8	Analogi <i>Radial Chart</i>	10
2.7	Analogi <i>Slopegraph</i>	11
2.9	Analogi <i>Glyph chart</i>	12
2.10	Analogi Diagram Sankey	12
2.11	Analogi <i>Area Size Chart</i>	13
2.12	Analogi <i>Trellis Chart</i>	13
2.13	Analogi <i>Word Cloud</i>	13
2.14	Analogi Diagram Lingkaran (<i>Pie Chart</i>)	14
2.15	Analogi Diagram Batang Bertumpuk (<i>Stacked Bar Chart</i>)	15
2.16	Analogi <i>Square Pie</i> atau <i>Unit Chart</i> atau <i>Waffle Chart</i>	15
2.17	Analogi <i>Tree Map</i>	16
2.18	Analogi <i>Circle Packing Diagram</i>	16
2.19	Analogi Hierarki Gelembung (<i>Bubble Hierarchy</i>)	17
2.20	Analogi Hierarki Pohon (<i>Tree Hierarchy</i>)	17
2.21	Analogi Diagram Garis (<i>Line Chart</i>)	18
2.22	Analogi <i>Sparklines</i>	18
2.23	Analogi Diagram Luas (<i>Area Chart</i>)	19
2.24	Analogi <i>Horizon Chart</i>	20
2.25	Analogi Diagram Luas Bertumpuk (<i>Stacked Area Chart</i>)	20
2.26	Analogi <i>Stream Chart</i>	21
2.27	Analogi <i>Candlestick Chart</i> atau <i>Box Plot</i> atau <i>Whiskers Plot</i>	21
2.28	Analogi <i>Barcode Chart</i>	22
2.29	Analogi <i>Flow Map</i>	22
2.30	Analogi <i>Scatter Plot</i>	23
2.31	Analogi <i>Bubble Plot</i>	23
2.32	Analogi <i>Scatter Plot Matrix</i>	24
2.33	Analogi Diagram Matriks (<i>Heatmap</i>)	25
2.34	Analogi <i>Parallel Sets</i> atau Koordinat Parallel	26
2.35	Analogi Diagram Senar atau <i>Radial Network</i>	27
2.36	Analogi Diagram Jaringan	27
2.37	Analogi <i>Choropleth map</i>	28
2.38	Analogi <i>Dot Plot Map</i>	28
2.39	Analogi <i>Bubble Plot Map</i>	29
2.40	Analogi <i>Isarithmic Map</i>	29
2.41	Analogi <i>Particle Flow Map</i>	30
2.42	Analogi Katogram (<i>Cartogram</i>)	30

2.43 Analogi Kartogram Dorling (<i>Dorling Cartogram</i>)	31
2.44 Protokol <i>Stack</i> WSN	34
2.45 Protokol komunikasi <i>Single-hop</i> dan <i>Multi-hop</i>	36
2.46 Topologi Poin ke Poin	36
2.47 Topologi Linear	37
2.48 Topologi Bus	37
2.49 Topologi Ring	37
2.50 Topologi Bintang	38
2.51 Topologi Pohon	38
2.52 Topologi <i>Fully Connected Mesh</i> dan <i>Partially Connected Mesh</i>	38
3.1 Skema Desain Sistem	46
3.2 Diagram Use Case Perangkat Lunak	47
3.3 Diagram Kelas Perangkat Lunak pada <i>Base Station</i>	50
3.4 Diagram Kelas Sederhana Perangkat Lunak pada <i>Node Sensor</i>	51
3.5 Diagram Relasi entitas Basis Data	54
3.6 Data yang di format dengan XML	59
3.7 Data yang di format dengan JSON	59
3.8 Data yang di format dengan YAML	59
3.9 Diagram Akselerasi ADXL 345	61
3.10 Preon32 VariSen	61
3.11 Preon32	62
4.1 Diagram Sequence Melakukan Login	63
4.2 Diagram Sequence Membuka Dashboard	64
4.3 Diagram Sequence Melihat Visualisasi Data	65
4.4 Diagram Sequence Menambah Kota Baru	65
4.5 Diagram Sequence Menambah Base Station	66
4.6 Diagram Sequence Menambah Base Station	66
4.7 Mock-up Halaman Login	67
4.8 Mock-up Halaman Daftar	68
4.9 Mock-up Halaman Visualisasi Data History	68
4.10 Mock-up Halaman Visualisasi Data Statistik	69
4.11 Mock-up Halaman Visualisasi Data RealTime	69
4.12 Contoh Halaman Visualisasi Data untuk Banyak Jenis Data	70
4.13 Contoh Halaman Visualisasi Data untuk Banyak Node Sensor	71
4.14 Kelas Diagram Node Sensor pada Preon32	74
4.15 Kelas Diagram Base Station pada Preon32	77
4.16 Kelas Diagram Base Station pada Komputer	78
4.17 Diagram Sequence Sinkronisasi Waktu dan Interval pada Node	80
5.1 Tangkapan Layar Halaman Admin	82
5.2 Tangkapan Layar Halaman Realtime	83
5.3 Tangkapan Layar Halaman Realtime Aktif	83
5.4 Tangkapan Layar Halaman History	84
5.5 Tangkapan Layar Halaman Statistik	84
5.6 Tangkapan Layar Halaman Tabel	85
5.7 Tangkapan Layar Halaman Ganti Kata Sandi	85
5.8 Tangkapan Layar Website Menampilkan Diagram Garis	96
5.9 Tangkapan Layar Website Menampilkan Scatter Plot	96
5.10 Tangkapan layar <i>Log</i> pada <i>Node Sensor</i> diPercobaan Kedua Eksperimen Ganti Interval	97
5.11 Tangkapan Layar <i>Log</i> pada <i>Node Sensor</i> Eksperimen Memberikan Interval Negatif	98

5.12 Tangkapan Layar Website Admin Memberikan Notifikasi Gagal Input Node	98
5.13 Tangkapan Layar Website Admin Memberikan Notifikasi <i>Key Node</i> yang Baru disimpan	99
5.14 Tangkapan Layar Website Menampilkan Pesan belum Memilih Lokasi	99
5.15 Tangkapan Layar Website Menampilkan Pesan Belum Memilih Tanggal Awal	100
5.16 Tangkapan Layar Website Menampilkan pesan belum memilih tanggal Awal	100
5.17 Tangkapan Layar Website Menampilkan Pesan Belum Memilih Waktu Awal	100
5.18 Tangkapan Layar Website Menampilkan Pesan Belum Memilih Tanggal Akhir	100
5.19 Tangkapan Layar Website Menampilkan Pesan Belum Memilih Waktu Akhir	100
5.20 Tangkapan Layar Website Menampilkan Pesan Belum Mengisi Interval	100
5.21 Tangkapan Layar Website Menampilkan Pesan Belum Memilih Tipe Visualisasi	100
5.22 Tangkapan Layar Website Menampilkan Pesan Belum Memilih Tipe Visualisasi untuk Akselerasi	100
5.23 Tangkapan Layar Mencoba untuk Menginput Data <i>Sensing</i> dengan Key Tidak Cocok	101
5.24 Tangkapan Layar Percobaan ke-3 <i>Node</i> Sensor Melakukan Sensing Kembali	102
5.25 Tangkapan Layar Mencoba Memasukan Masukan yang Salah	103
5.26 Tangkapan Layar API <i>Crash</i>	104
5.27 Raspberry Mengaktifkan Preon32	104
5.28 Data Diterima Di <i>Base Station</i>	104
5.29 Raspberry Mengaktifkan Preon32 yang Bertindak Sebagai <i>Base Station</i>	105

DAFTAR TABEL

2.1	Tabel Klasifikasi Grafik Berdasarkan Tujuan Visualisasi [1, 2]	7
2.2	Tabel Jenis Sensor	33
2.3	Sistem Operasi pada WSN	35
2.4	Tabel Spesifikasi <i>Micro-Controller</i> Preon32 [3]	39
2.5	Tabel Spesifikasi <i>High Frequency Transceiver</i> Preon32 [3]	39
2.6	Tabel Spesifikasi <i>Interface</i> Preon32 [3]	39
2.7	Tabel Spesifikasi <i>General</i> Preon32 [3]	40
3.1	Tabel Use Case Skenario Melihat Visualisasi Data	47
3.2	Tabel Use Case Skenario Menambahkan Kota	48
3.3	Tabel Use Case Skenario Menambahkan <i>Base Station</i> Baru	48
3.4	Tabel Use Case Skenario Mengganti Selang Waktu <i>Sensing</i>	48
3.5	Tabel Use Case Skenario Menambahkan <i>Node Sensor</i> Baru	49
3.6	Tabel Use Case Skenario Memberikan Data <i>Sensing</i>	49
3.7	Rancangan Fisik Entitas <i>Base Station</i>	53
3.8	Rancangan Fisik Entitas Kota	53
3.9	Rancangan Fisik Tipe Sensor	54
3.10	Rancangan Fisik <i>Queue Update</i>	54
3.11	Rancangan Fisik Entitas <i>Node Sensor</i>	55
3.12	Rancangan Fisik Entitas User	55
3.13	Rancangan Fisik Entitas Data <i>Sensing</i>	55
3.14	Tabel format Data Tipe Sensor	58
4.1	Endpoint API untuk Pendaftaran	71
4.2	Endpoint API untuk Masuk	71
4.3	Endpoint API untuk Memasukan Kota Baru	72
4.4	Endpoint API untuk Mengambil Kota	72
4.5	Endpoint API untuk Memasukan <i>Base Station</i> Baru	72
4.6	Endpoint API untuk Mengambil <i>Base Station</i>	72
4.7	Endpoint API untuk Memasukan <i>Node Sensor</i> Baru	72
4.8	Endpoint API untuk Mengambil <i>Node Sensor</i> Baru	72
4.9	Endpoint API untuk Menyimpan Data <i>Sensing</i>	73
4.10	Endpoint API untuk Melakukan Pembaharuan Interval <i>Sensing</i>	73
4.11	Endpoint API untuk Mengambil Selang Waktu <i>Node Sensor</i>	73
4.12	Endpoint API untuk Mengambil Data untuk Divisualisasikan	73
4.13	Endpoint API untuk Mengambil Data <i>Real Time</i> untuk Visualisasi	73
5.1	Tabel Lingkungan Perangkat Keras	81
5.2	Tabel Lingkungan Perangkat Lunak	81
5.3	Tabel Lingkungan Library dan Framework	82
5.4	Implementasi API untuk Login	86
5.5	Implementasi API untuk Daftar	87
5.6	Implementasi API untuk Memasukan Kota Baru	87

5.7	Implementasi API untuk mengambil Seluruh Kota	88
5.8	Implementasi API untuk Memasukan Data <i>Sensing</i>	89
5.9	Implementasi API untuk Memasukan <i>Base Station</i> Baru	90
5.10	Implementasi API untuk Mengambil Seluruh <i>Base Station</i>	90
5.11	Implementasi API untuk Memasukan <i>Node Sensor</i> Baru	91
5.12	Implementasi API untuk Mengambil Seluruh <i>Node Sensor</i>	92
5.13	Implementasi API untuk Mengganti Selang Waktu <i>Sensing Node Sensor</i>	93
5.14	Implementasi API untuk Mengambil Interval Waktu <i>Node Sensor</i>	93
5.15	Implementasi API untuk Mengirim Data <i>Sensing</i>	94
5.16	Implementasi API untuk Mengambil Data <i>Sensing</i>	95
5.17	Implementasi API untuk Mengambil Data <i>Sensing Realtime</i>	96
5.18	Percobaan <i>Node Sensor</i> Ganti Interval	97
5.19	Percobaan <i>Node Sensor</i> Tersambung Kembali	103

BAB 1

PENDAHULUAN

Pada bab ini akan dipaparkan latar belakang yang mendeskripsikan gambaran besar permasalahan yang ada. Dilanjutkan dengan pembahasan rumusan masalah, tujuan penelitian, dan batasan penelitian. Pada tiga buah bagian tersebut, dijelaskan inti masalah, tujuan yang ingin dicapai sebagai solusi masalah, serta asumsi yang digunakan untuk membatasi ruang lingkup penelitian ini.

1.1 Latar Belakang

Wireless Sensor Network (WSN) adalah sekumpulan *node* sensor yang dihubungkan dalam jaringan komputer dengan memanfaatkan teknologi nirkabel. Pada umumnya WSN memiliki kemampuan penyimpanan dan pengolah data yang tergolong kecil, jika dibandingkan dengan komputer maupun *smartphone*. Setiap *node* sensor memiliki kemampuan memantau kondisi seperti suhu, suara, getaran, kelembapan, dan lain sebagainya. Nilai pengukuran yang didapatkan setiap *node* dikumpulkan dalam sebuah *base station*. Kemudian oleh *base station* akan dikirimkan ke aplikasi yang membutuhkannya.

Pada aplikasi jika WSN hanya menyediakan berbentuk data numerik, misalkan temperatur yang dibaca oleh *node* sensor adalah 30°C, maka *insight*—suatu informasi berharga— yang didapatkan sebatas kondisi temperatur saat ini panas atau dingin saja. Namun jika data nilai pengukuran dari *node* sensor dikumpulkan dari waktu ke waktu dan dibuatkan visualisasi data, misalkan dibuatkan visualisasi data berupa grafik garis pada satu rentang waktu tertentu maka dapat memunculkan informasi lain seperti bagaimana fluktuasi atau tren data tersebut, apakah cenderung meningkat atau cenderung menurun pada suatu rentang waktu tertentu, oleh karena itu untuk mendapatkan *insight* akan dibuatkan visualisasi data yang sesuai dengan kebutuhan pengguna.

Visualisasi data merupakan tampilan berupa grafik atau bentuk tampilan lainnya yang berfungsi untuk memberikan suatu informasi dari data agar lebih mudah dipahami serta dapat memberikan *insight* lebih. Visualisasi data dapat membantu pengguna untuk memahami data dengan memberikan rangkuman visual informasi yang lebih mudah untuk dipahami. Dalam visualisasi data untuk memberikan *insight* yang sesuai dan akurat maka dibutuhkan kesesuaian antara data yang diukur oleh sensor, kebutuhan pengguna, serta jenis visualisasi data.

Salah satu tantangan masalah yang ada pada pembuatan visualisasi data adalah pada kebutuhan pengguna yang beragam. Ada pengguna yang memiliki kebutuhan untuk mengetahui kondisi suatu lingkungan secara *real-time* maka perangkat lunak dan visualisasi data harus dapat menampilkan data hasil pengukuran secara *real-time*. Kemudian ada pengguna yang memiliki kebutuhan untuk mengetahui fluktuasi suatu parameter pada lingkungan maka visualisasi data yang dibuatkan dengan grafik yang dapat menunjukkan fluktuasi seperti grafik garis. Masih banyak lagi kebutuhan pengguna oleh karena itu diperlukan analisis mendalam mengenai parameter *node* sensor terhadap perkiraan kebutuhan pengguna.

Pada skripsi ini akan dilakukan analisis visualisasi data yang dibutuhkan agar bersesuaian tepat dengan kebutuhan pengguna dan sesuai dengan data *sensing* yang dihasilkan WSN dengan cara mengembangkan aplikasi yang dapat menampilkan visualisasi data tertentu dan didasarkan pada kebutuhan pengguna.

1.2 Rumusan Masalah

Berdasarkan latar belakang pada bagian 1.1 yang telah dipaparkan, didapatkan rumusan masalah sebagai berikut,

1. Bagaimana menentukan jenis visualisasi data yang tepat dan sesuai dengan kebutuhan pengguna dari data yang diberikan oleh WSN, agar dapat lebih memberikan *insight* yang tepat dan sekaligus tampilan yang menarik?
2. Bagaimana membangun perangkat lunak yang dapat menampilkan visualisasi data yang sesuai dengan kebutuhan pengguna dan data yang diukur oleh WSN?

1.3 Tujuan

Berdasarkan rumusan pada bagian 1.2 yang ada, maka tujuan skripsi ini sebagai berikut,

1. Mempelajari cara penentuan visualisasi yang tepat, sesuai dengan kebutuhan dan jenis data *node* sensor pada WSN.
2. Membangun perangkat lunak yang dapat menampilkan visualisasi yang tepat sesuai kebutuhan pengguna dan jenis data yang diukur oleh WSN.

1.4 Batasan Masalah

Batasan untuk pembuatan skripsi ini adalah sebagai berikut,

1. Sensor yang digunakan hanya sensor untuk mengukur suhu, kelembapan, getaran dan juga tekanan udara.
2. Perangkat keras yang digunakan pada skripsi ini 4 buah *node* sensor.
3. Perangkat keras yang digunakan hanya Preon32.
4. Perangkat lunak hanya dibuat dalam bentuk *web*.
5. Pengujian kecocokan visualisasi data dengan data hanya dilakukan berdasarkan teori yang dijelaskan di bab 2.

1.5 Metodologi

Metodologi yang digunakan dalam pembuatan skripsi ini sebagai berikut

1. Mempelajari hal-hal berikut,
 - (a) Cara kerja *Wireless Sensor Network* (WSN).
 - (b) Cara kerja perangkat Preon32.
 - (c) Kaidah visualisasi data.
 - (d) Bahasa pemrograman Java.
 - (e) Cara kerja API.
2. Mendesain skema aplikasi.
3. Melakukan analisis terhadap kategori yang dibaca oleh *node* sensor.
4. Melakukan analisis dan perancangan perangkat lunak yang akan dibangun.
5. Membangun Perangkat lunak yang telah dirancang.
6. Melakukan *testing* perangkat lunak yang sudah dibangun.
7. Melakukan analisis terhadap visualisasi dan perangkat lunak secara keseluruhan.

1.6 Sistematika Pembahasan

Sistematika Pembahasan pada skripsi ini yang berjudul Visualisasi Data Sensor Di Wireless Sensor Networks ini adalah sebagai berikut:

Bab 1 membahas latar belakang masalah, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika pembahasan

Bab 2 berisikan dasar teori untuk visualisasi data, *wireless sensor networks*, Preon32, HTTP, dan *library* yang digunakan pada skripsi ini.

Bab 3 membahas mengenai deskripsi perangkat lunak yang berisikan analisis fungsi perangkat lunak. Analisis tujuan pembuatan visualisasi data sensor, analisis perangkat lunak, dan analisis masalah.

Bab 4 berisikan mengenai perancangan dimulai dari perancangan antarmuka, perancangan API, dan terakhir perancangan dari WSN.

Bab 5 berisikan implementasi terhadap perangkat lunak dimulai dari implementasi yang berisikan perangkat lunak lain yang dibutuhkan pada skripsi ini, kemudian dilampirkan juga hasil implementasi website dan API. Setelah implementasi terdapat pengujian terhadap perangkat lunak.

Bab 6 berisikan mengenai kesimpulan dan saran yang terkait pada skripsi ini berdasarkan dari masalah dan pengujian yang dilakukan pada bab 5.

BAB 2

LANDASAN TEORI

Bab ini diawali dengan pembahasan teori mengenai visualisasi data yang mencakup pengertian, tujuan, kaidah, dan pemilihan desain dari visualisasi data. Dilanjutkan dengan WSN yang membahas mengenai definisi sensor, jenis sensor, definisi WSN, dan protokol komunikasi WSN. Selanjutnya membahas tentang Preon32 yang digunakan pada skripsi ini.

2.1 Visualisasi Data

Visualisasi data merupakan sebuah pemrosesan terhadap data dengan cara dibuatkan tampilan berupa grafik, bagan, atau bentuk lainnya untuk membantu pengguna memahami data yang sangat banyak dan kompleks. Visualisasi data dapat membantu pengguna memahami data dengan jumlah yang sangat banyak dan kompleks dengan cara memberikan ringkasan visual, akan sangat sulit jika seorang manusia harus memperhatikan baris demi baris untuk menemukan pola atau informasi yang dibutuhkan. Maka dari itu, dibutuhkan visualisasi data agar menjadi lebih mudah dimengerti, lebih bermakna, dan lebih mudah untuk digunakan dalam berkomunikasi antar orang [4].

2.1.1 Kaidah Visualisasi Data

Pembuatan visualisasi data merupakan suatu tahapan yang tidak mudah, dikarenakan pembuat visualisasi data harus mempertimbangkan data yang tersedia dengan berbagai aspek. Aspek-aspek yang harus diperhatikan yaitu[1]:

1. Tujuan pembuatan visualisasi

Pembuat Visualisasi data sebaiknya menentukan tujuan pembuatan visualisasi. Dengan mengetahui tujuan maka pembuat visualisasi data bisa menentukan berbagai hal seperti pemilihan grafik, pemilihan data, dan lain sebagainya. Selain itu dengan menentukan tujuan maka pembuat dapat menguji kesesuaian visualisasi data dengan tujuan.

2. Mengenal data

Pembuat visualisasi data sebaiknya mengenal data yang dimiliki, agar dapat pembuat visualisasi dapat memastikan data yang dimiliki sudah cocok dan benar(*valid*) dengan tujuan pembuatan visualisasi agar tidak terjadi bias.

3. Nilai informasi disampaikan secara efektif

Pembuat visualisasi sebaiknya menentukan data yang perlu ditampilkan agar audiens tidak kehilangan fokus pada suatu informasi yang hendak ditunjukkan oleh pembuat visualisasi. Selain menentukan data yang perlu ditampilkan pembuat visualisasi data juga perlu menyesuaikan antara data yang ditampilkan dengan jenis visualisasi data.

4. Audiens

Pembuat visualisasi sebaiknya dapat menentukan audiens secara spesifik agar informasi yang disajikan sesuai dengan minat atau ketertarikan audiens, latar belakang audiens, dan lain sebagainya terhadap data yang dimiliki. Hal ini dilakukan agar audiens tertarik dan mendapatkan informasi menarik dari visualisasi data yang dibuat.

5. Media penyebaran

Pembuat visualisasi data sebaiknya mempertimbangkan media penyebaran dengan tipe visualisasi data yang dipilih. Pembuat visualisasi data juga sebaiknya berhati-hati dalam orientasi visualisasi, gradien warna, dan lain sebagainya karena setiap media penyebaran memiliki kemampuan visualisasi yang berbeda.

2.1.2 Jenis-jenis Visualisasi Data

Menurut Andy Kirk pada buku *Data Visualization: a Successful Design Process* Visualisasi data bergantung dengan desain, pemilihan desain harus sesuai dengan tujuan pembuatan visualisasi. Kirk juga berpendapat bahwa ada dua buah dasar yaitu representasi data dan penyajian data. Representasi data berhubungan erat dengan pemilihan jenis grafik, ada berbagai jenis grafik antara lain:

- Tulisan Sederhana

Jenis visualisasi data dalam bentuk tulisan sederhana digunakan pada saat data yang ditampilkan hanya berjumlah satu atau dua. Hal ini bertujuan agar audiens tidak salah fokus dan tidak lelah dalam menerima informasi.

Simple text

Gambar 2.1: Gambar Ilustrasi Visualisasi Data Dalam Bentuk Tulisan Sederhana

- Tabel

Jenis visualisasi data dalam bentuk tabel digunakan pada saat data yang ditampilkan memiliki satuan yang berbeda-beda.

- Grafik

Jenis Visualisasi data dalam bentuk grafik digunakan pada saat data yang ditampilkan memiliki satuan yang sama. Grafik memiliki keuntungan yaitu lebih cepat dimengerti oleh audiens jika dibandingkan dengan menggunakan visualisasi data dalam bentuk tabel. Visualisasi data grafik memiliki berbagai tipe, setiap tipe grafik dapat diklasifikasikan berdasarkan tujuan visualisasinya. Klasifikasi tujuan grafik dapat dilihat pada tabel 2.1 dan untuk tipe tipe grafik dapat dilihat pada sub-bagian 2.1.3.

2.1.3 Jenis-Jenis Visualisasi Data Grafik

- Membandingkan Kategori [1]

- Diagram Titik (*Dot Plot*)

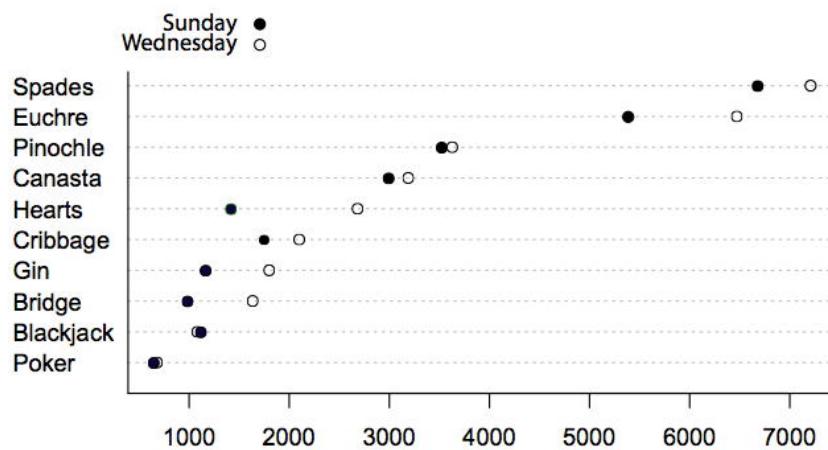
Tipe data yang dibutuhkan: 2 kategorikal dan 1 kuantitatif

Variabel Visual: posisi, hue, dan simbol

Diagram titik cocok digunakan jika ingin menampilkan perbandingan nilai dalam dua kategori atau lebih. Diagram ini menampilkan perbandingan dua buah atau lebih nilai kategorikal dan setidaknya satu kuantitatif disetiap kategorikal menggunakan sebuah titik atau simbol pada kolom dan baris yang bersesuaian. Contoh visualisasi dapat dilihat pada gambar 2.2.

Tabel 2.1: Tabel Klasifikasi Grafik Berdasarkan Tujuan Visualisasi [1, 2]

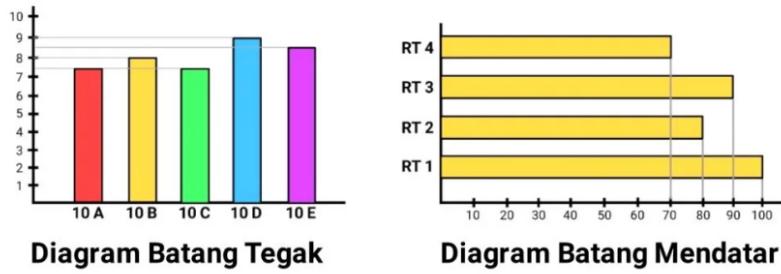
Klasifikasi	Tujuan
Membandingkan kategori	Untuk membandingkan sejumlah kategori, dengan cara membandingkan nilai relatif atau mutlak dari data kategori.
Nilai hierarki dan hubungan bagian ke keseluruhan (part-to-whole)	Untuk menampilkan rincian nilai kategori dalam hubungannya populasi nilai atau sebagai elemen penyusun hierarki.
Menampilkan data yang terus berubah-ubah	Untuk Menampilkan data sementara, tren, dan pola dari waktu yang berkelanjutan.
Menampilkan plot dan hubungan	Untuk menilai asosiasi, distribusi, dan pola yang ada di antara kumpulan data multivariat. Biasanya ditujukan untuk memfasilitasi analisis atau eksplorasi.
Menampilkan data geo-spasial	Untuk menampilkan plot dan menyajikan data geospasial.

Gambar 2.2: Analogi Diagram Titik (*Dot Plot*) ¹

- Diagram Batang (*Bar Chart* atau Diagram Kolom *Column Chart*)
 - Tipe data yang dibutuhkan: 1 Kategorikal dan 1 rasio kuantitatif
 - Variabel Visual: panjang, hue

Pada diagram batang ini digunakan jika ingin menampilkan perbandingan nilai rasio kuantitatif antar data kategorikal. Diagram ini dapat membandingkan nilai kuantitatif setiap kategorinya dengan panjang batang yang diatur lebih tinggi atau rendah sesuai dengan nilai kuantitatifnya. Pada Diagram ini hendaknya nilai terendah batang selalu dimulai dari nilai nol. Contoh visualisasi dapat dilihat pada gambar 2.3.

¹https://www.onlinestatbook.com/2/graphing_distributions/dotplots.html Diakses Juli 2022

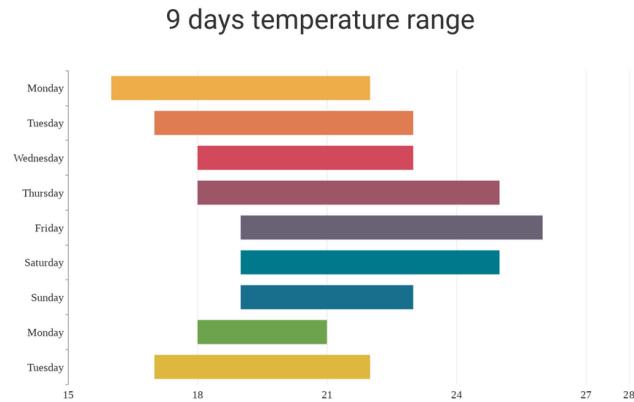


Gambar 2.3: Analogi Diagram Batang (*Bar Chart* atau Diagram Kolom *Column Chart*) ²

■ *Floating Bar* atau (*Gantt Chart*)

Tipe data yang dibutuhkan: 1 kategorikal dan 1 rasio kuantitatif
Variabel Visual: Panjang atau lebar dan hue

Pada diagram *floating bar* digunakan jika ingin menampilkan perbandingan nilai terendah dan tertinggi antar kategori. Perbedaan diagram ini dengan [diagram batang](#) adalah diagram ini bersifat relatif atau tidak wajib untuk mulai dari nilai nol, sedangkan [diagram batang](#) nilai wajib dimulai dari nol. Contoh visualisasi dapat dilihat pada gambar [2.4](#).



Gambar 2.4: Analogi *Floating Bar* atau (*Gantt Chart*) ³

■ *Pixelated Bar Chart*

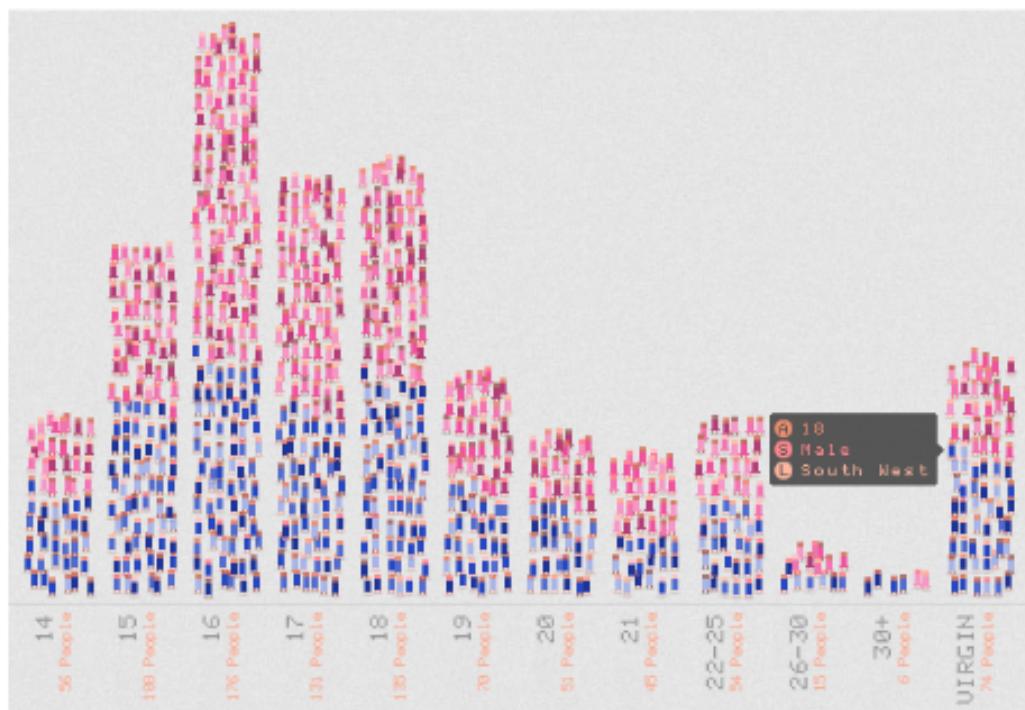
Tipe data yang dibutuhkan: Banyak kategorikal dan 1 kuantitatif
Variabel Visual: Tinggi, hue, dan simbol

Diagram ini mirip dengan diagram batang, namun diagram ini memiliki sifat interaktif yaitu jika kursor diarahkan atau melakukan *click* di simbol, maka akan muncul cerita atau informasi yang lebih rinci. Contoh visualisasi dapat dilihat pada gambar [2.5](#).

²<https://www.cilacapklik.com/2021/09/cara-membuat-diagram-batang-yang-benar.html> Diakses Juli 2022

³<https://online.visual-paradigm.com/charts/templates/floating-bar-charts/floating-bar-chart> Diakses Juli 2022

⁴Kirk, A. Data Visualization: a successful design process, 2st edition. (Birmingham: Packt Publishing,2012), hlm. 125



Gambar 2.5: Analogi *Pixelated Bar Chart*⁴

■ Histogram

Tipe data yang dibutuhkan: 1 interval kuantitatif dan 1 rasio kuantitatif
 Variabel Visual: Tinggi dan lebar

Histogram pada dasarnya mirip dengan [Diagram Batang](#) namun perbedaannya yaitu jika pada diagram batang satu kategori dan satu kuantitatif sedangkan *histogram* kategori tersebut berubah menjadi rentang interval dan untuk nilai kuantitatifnya masih sama. Contoh visualisasi dapat dilihat pada gambar [2.6](#).

■ Slopegraph

Tipe data yang dibutuhkan: 1 kategorikal dan 2 kuantitatif
 Variabel Visual: Posisi, koneksi, dan hue

Diagram ini digunakan untuk membandingkan dua atau lebih nilai kuantitatif yang kategorinya sama. Diagram ini umumnya digunakan untuk membandingkan sebuah kategori dengan nilai kuantitatifnya berbeda waktu. Contoh visualisasi dapat dilihat pada gambar [2.7](#).

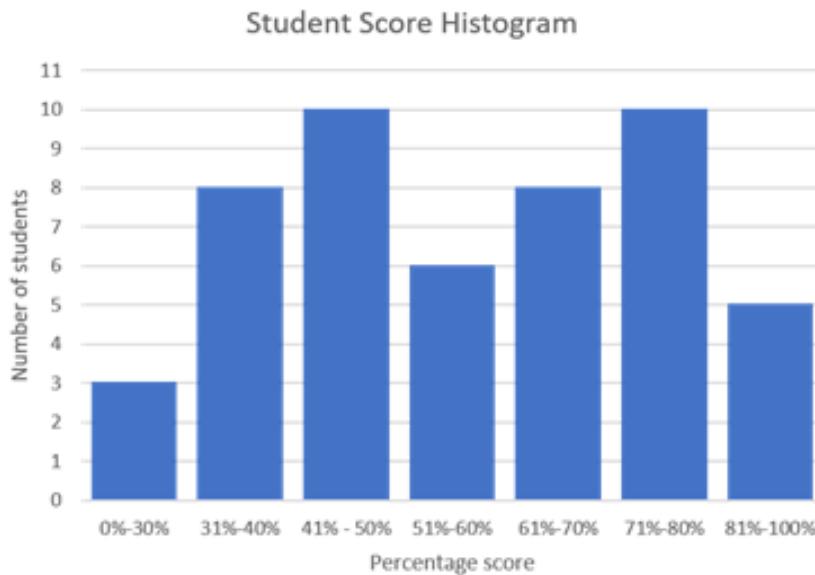
■ Radial Chart

Tipe data yang dibutuhkan: Banyak kategorikal dan 1 kategorikal-ordinal
 Variabel Visual: Posisi, hue, saturasi atau ketebalan, dan tekstur

Radial chart menampilkan data secara melingkar yang memiliki pusat yang sama(ko-sentris). Diagram ini sering digunakan untuk menampilkan data yang memiliki urutan kontinu seperti data yang berubah seiring dengan waktu. Contoh visualisasi dapat dilihat pada gambar [2.8](#).

⁵<https://www.tibco.com/reference-center/what-is-a-histogram-chart> Diakses Juli 2022

⁶<https://www.storytellingwithdata.com/blog/2018/6/1/swdchallenge-slopeograph> Diakses Juli 2022



Gambar 2.6: Analogi *Histogram*⁵



Gambar 2.8: Analogi *Radial Chart*⁷

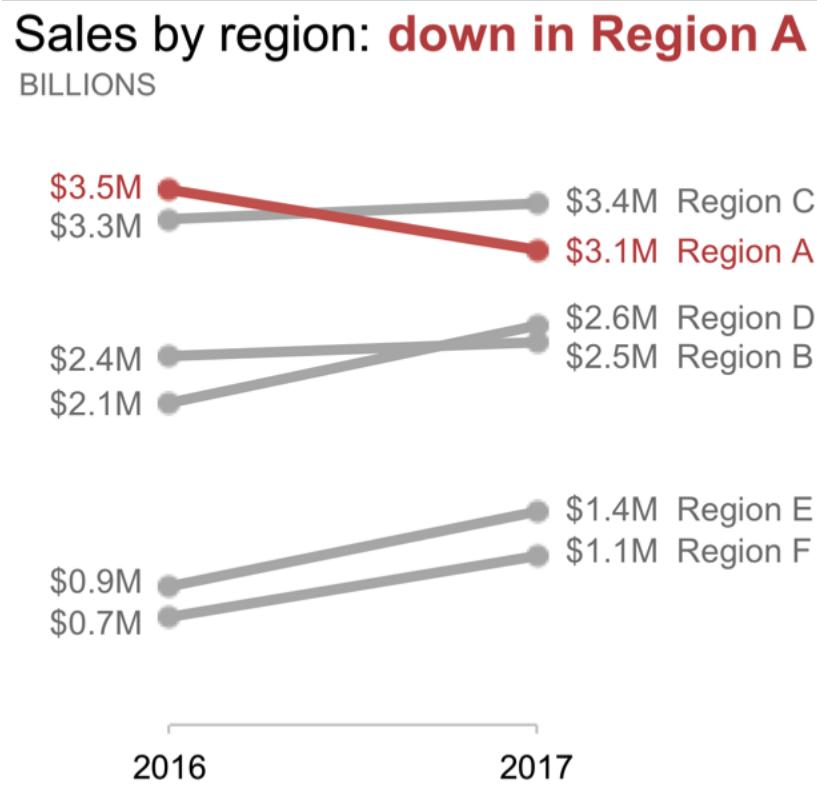
■ *Glyph chart*

Tipe data yang dibutuhkan: Banyak kategorikal, dan banyak kuantitatif
 Variabel Visual: Bentuk, urutan, posisi, dan hue

Diagram Glyph merupakan diagram yang bergantung bentuk sebagai artefak utamanya. Pada diagram ini sifat-sifat bentuk seperti ukuran, warna sebagai bentuk representasi data. Seperti sifat bentuk, bentuk yang berbeda menandakan kategori yang berbeda, sedangkan ukuran dan warna menyesuaikan dengan nilai kuantitatif. Contoh visualisasi dapat dilihat pada gambar 2.9.

⁷<https://www.amcharts.com/demos/radial-bar-chart/> Diakses Juli 2022

⁸Kirk, A. Data Visualization: a successful design process, 2st edition. (Birmingham: Packt Publishing,2012), hlm. 128



Gambar 2.7: Analogi *Slopegraph*⁶

■ Diagram Sankey

Tipe data yang dibutuhkan: Banyak kategorikal dan banyak kuantitatif
 Variabel Visual: Tinggi, posisi, hubungan, dan hue

Diagram sankey biasanya digunakan untuk menampilkan data dengan bentuk aliran. Diagram ini ditujukan untuk menampilkan kondisi atau saat data berubah atau terbagi atas suatu kejadian. Contoh visualisasi dapat dilihat pada gambar 2.10.

■ Area Size Chart

Tipe data yang dibutuhkan: 1 Kategorikal dan 1 rasio kuantitatif
 Variabel Visual: Luas dan hue

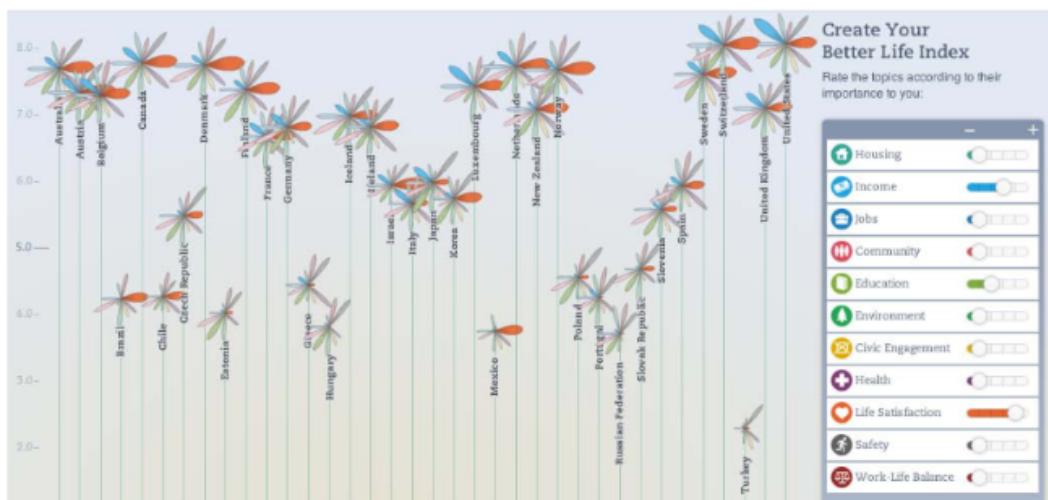
Diagram ini sangat sederhana, diagram ini hanya menampilkan rasio nilai kuantitatif antar kategorikal dengan cara memberi luas sesuai dengan rasio kuantitatifnya. Sedangkan untuk perbedaan kategori ditandai dengan warna yang berbeda. Diagram ini cocok untuk menampilkan proporsi data. Contoh visualisasi dapat dilihat pada gambar 2.11.

■ Trellis Chart

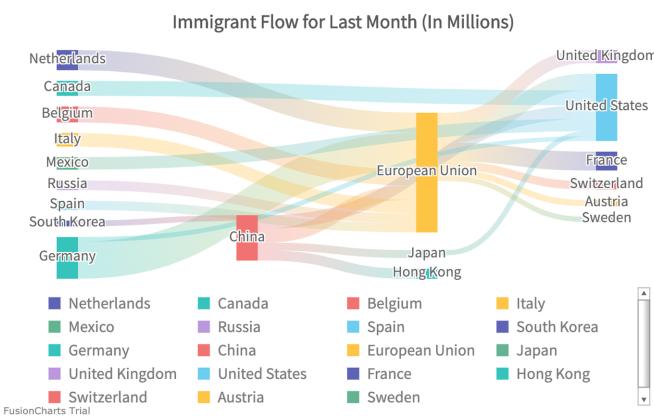
Tipe data yang dibutuhkan: Banyak kategorikal dan banyak kuantitatif
 Variabel Visual: Posisi

⁹ <https://www.fusioncharts.com/dev/chart-guide/standard-charts/sankey-diagram> Diakses Juli 2022

¹⁰ Kirk, A. Data Visualization: a successful design process, 2st edition. (Birmingham: Packt Publishing,2012), hlm. 129



Gambar 2.9: Analogi *Glyph chart*⁸



Gambar 2.10: Analogi Diagram Sankey⁹

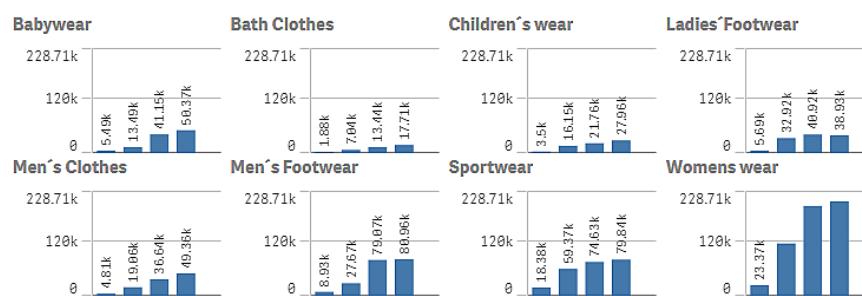
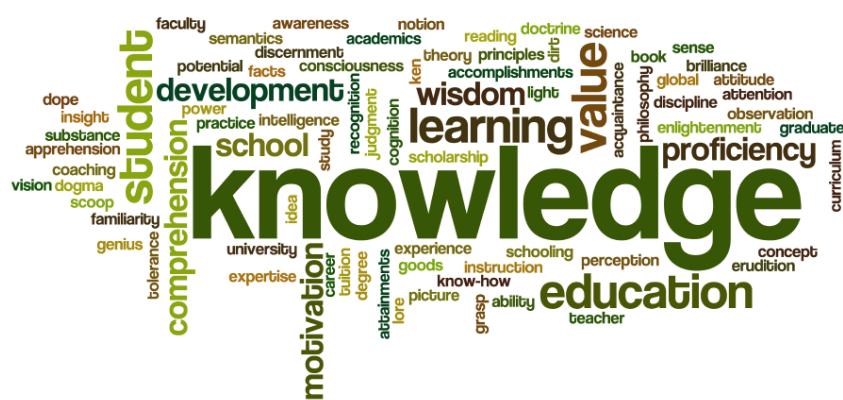
Trellis chart cocok digunakan untuk membandingkan data kategorikal yang memiliki rentang luas dengan data kuantitatif dalam satu waktu yang sama. Diagram ini sering digunakan untuk menampilkan tangkapan atau *snapshot* dari kejadian yang berubah seiring waktu. Contoh visualisasi dapat dilihat pada gambar 2.12.

■ Word Cloud

Tipe data yang dibutuhkan: 1 kategorikal dan 1 rasio kuantitatif
 Variabel Visual: Ukuran

Word Cloud digunakan untuk menampilkan frekuensi kata yang muncul atau ditulis dalam suatu teks tertentu. *Word Cloud* menampilkan frekuensi kata dengan cara mengatur ukuran kata, jika kata tersebut paling sering muncul maka kata tersebut harus memiliki ukuran terbesar. Contoh visualisasi dapat dilihat pada gambar 2.13.

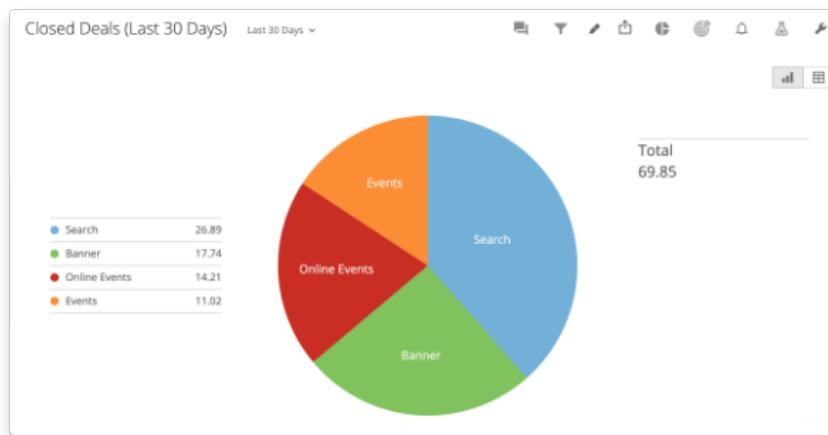
¹¹https://helpqlik.com/en-US/sense/August2022/Subsystems/Hub/Content/Sense_Hub/Visualizations/VisualizationBundle/trellis-container.htm Diakses November 2022

Gambar 2.11: Analogi *Area Size Chart*¹⁰Gambar 2.12: Analogi *Trellis Chart*¹¹Gambar 2.13: Analogi *Word Cloud*¹²

- Nilai hierarki dan hubungan bagian ke keseluruhan(part-to-whole)[1]
 - Diagram Lingkaran (*Pie Chart*)
Tipe data yang dibutuhkan: 1 Kategorikal dan 1 rasio kuantitatif
Variabel Visual: sudut, luas, dan hue

¹²<https://www.resilienteducator.com/classroom-resources/word-clouds-writing/> Diakses November 2022

Diagram lingkaran digunakan untuk menunjukkan hubungan bagian ke keseluruhan. Diagram Lingkaran menunjukkan antar kategori dan rasio kuantitatif kategori tersebut dengan ukuran sudut dalam diagram, semakin besar sudut yang dibentuk artinya kategori tersebut memiliki rasio kuantitatif yang semakin besar. Diagram lingkaran memiliki beberapa variasi lain seperti tengahnya dihilangkan sehingga terlihat seperti donat, hanya menampilkan setengah dari lingkaran atau sering disebut sebagai *gauge chart*. Contoh visualisasi dapat dilihat pada gambar 2.14.



Gambar 2.14: Analogi Diagram Lingkaran (*Pie Chart*) ¹³

■ Diagram Batang Bertumpuk (*Stacked Bar Chart*)

Tipe data yang dibutuhkan: 2 kategorikal dan 1 kuantitatif

Variabel Visual: panjang, hue, posisi, saturasi atau ketebalan

Diagram bertumpuk cocok digunakan untuk menampilkan nilai yang absolut atau sudah standarisasi untuk menunjukkan bagian. Diagram ini menampilkan perbedaan antar kategori nilai dengan warna dan posisi. Suatu nilai kategori yang terdapat kuantitatif di setiap kategori diurutkan dan diberi warna yang sama. Misalkan pada gambar 2.15 terdapat 3 kategori utama yaitu *coffee*, *tea*, dan *milk*. Ditiap kategori utama ada kategori lain yang memiliki nilai kuantitatif yaitu *Q1 Budget*, *Q2 Budget*, *Q1 Actual*, dan *Q1 Actual*. Untuk setiap kategori lain diberi warna yang serta penempatan yang relatif sama (kalau pada gambar 2.15 sebelah kiri maka posisinya *Q1 Budget* selalu berada di sebelah kiri bawah kategorinya). Diagram ini cocok digunakan untuk menampilkan komposisi sentimen dari semua kategori. Contoh visualisasi dapat dilihat pada gambar 2.15.

■ *Square Pie* atau *Unit Chart* atau *Waffle Chart*

Tipe data yang dibutuhkan: 1 Kategorikal dan 1 rasio kuantitatif

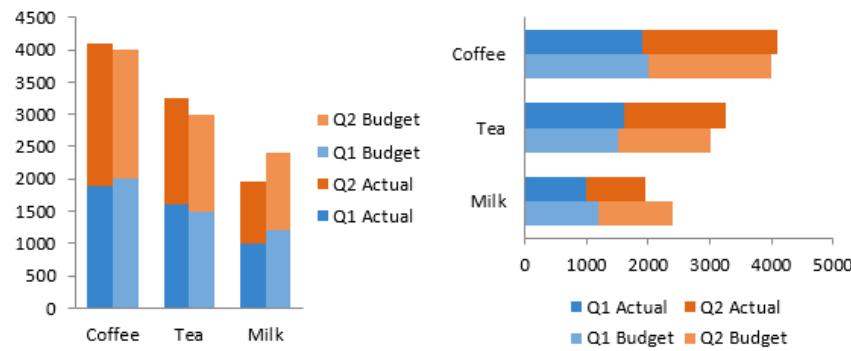
Variabel Visual: posisi, hue, simbol

Diagram *Square Pie* terdapat beberapa jenis, namun yang paling umum adalah *grid* atau simbol. Sebuah *grid* atau simbol yang sama digunakan untuk menampilkan sebuah rasio kuantitatif dari sebuah kategori. Contoh visualisasi dapat dilihat pada gambar 2.16.

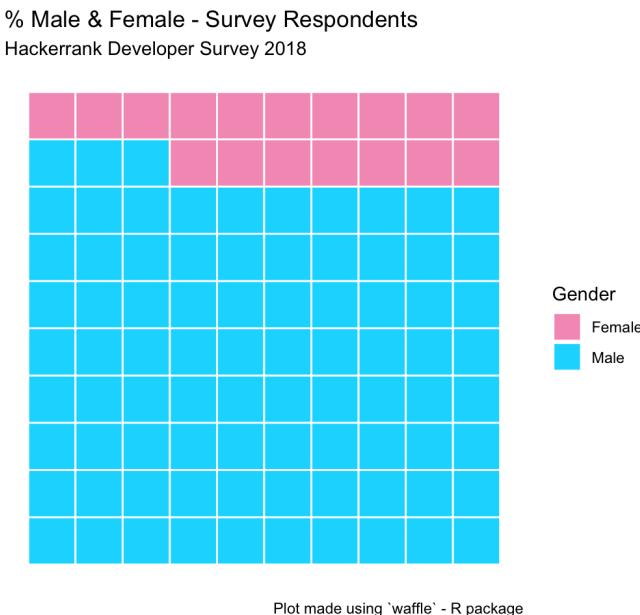
¹³<https://www.domo.com/charts/pie-charts> Diakses November 2022

¹⁴<https://www.peltiertech.com/clustered-stacked-column-bar-charts> Diakses November 2022

¹⁵<https://www.r-bloggers.com/2019/08/how-to-make-square-pie-charts-for-infographics-in-r/> Diakses November 2022



Gambar 2.15: Analogi Diagram Batang Bertumpuk (*Stacked Bar Chart*) ¹⁴



Plot made using 'waffle' - R package

Gambar 2.16: Analogi *Square Pie* atau *Unit Chart* atau *Waffle Chart* ¹⁵

■ *Tree Map*

Tipe data yang dibutuhkan: Banyak kategorikal nominal dan 1 kuantitatif rasio
Variabel Visual: Luas, posisi, hue, saturasi

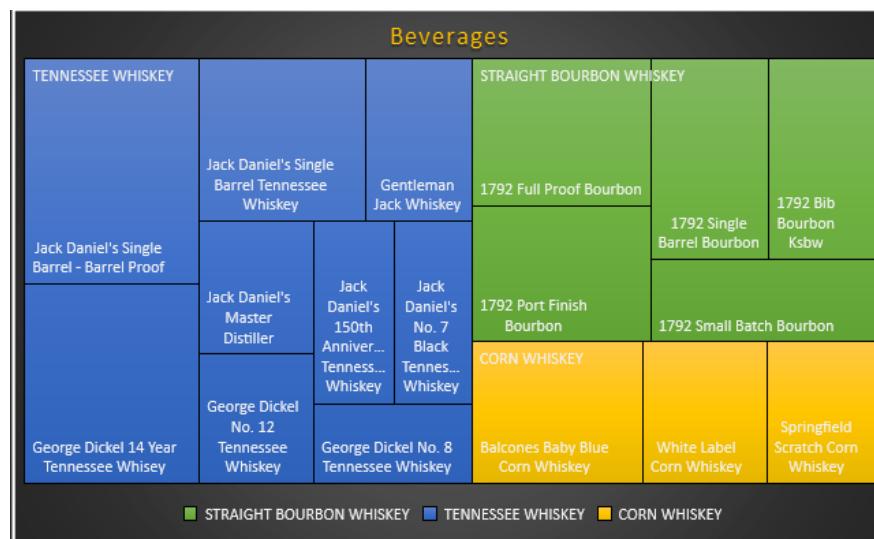
Tree map menggambarkan bagian dari keseluruhan dengan cara membagi luas daerah visualisasi menjadi beberapa bagian yang luas daerahnya sesuai dengan proporsi kategorikal. Pada diagram ini juga dapat ditambahkan variasi yaitu warna untuk menangani tambahan kategori. Contoh visualisasi dapat dilihat pada gambar [2.17](#).

■ *Circle Packing Diagram*

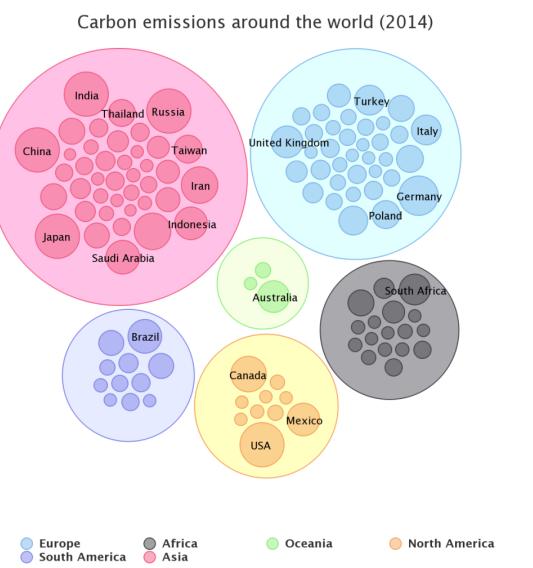
Tipe data yang dibutuhkan: 2 kategorikal dan 1 rasio kuantitatif
Variabel Visual: Luas, hue, dan posisi

Sesuai dengan namanya, diagram ini mengandung lingkaran-lingkaran yang dikumpulkan dan dikemas menjadi satu buah diagram. tiap lingkaran memiliki ukurannya masing-masing untuk menampilkan perbedaan kuantitatif rasio antar kategori. Diagram ini

¹⁶<https://www.spreadsheetweb.com/treemap-excel-charts-hierarchical-data/> Diakses Desember 2022

Gambar 2.17: Analogi Tree Map ¹⁶

juga memiliki beberapa tambahan variasi seperti warna dan posisi yang ditujukan untuk meningkatkan arti dari visualisasi. Contoh visualisasi dapat dilihat pada gambar 2.18.

Gambar 2.18: Analogi Circle Packing Diagram ¹⁷

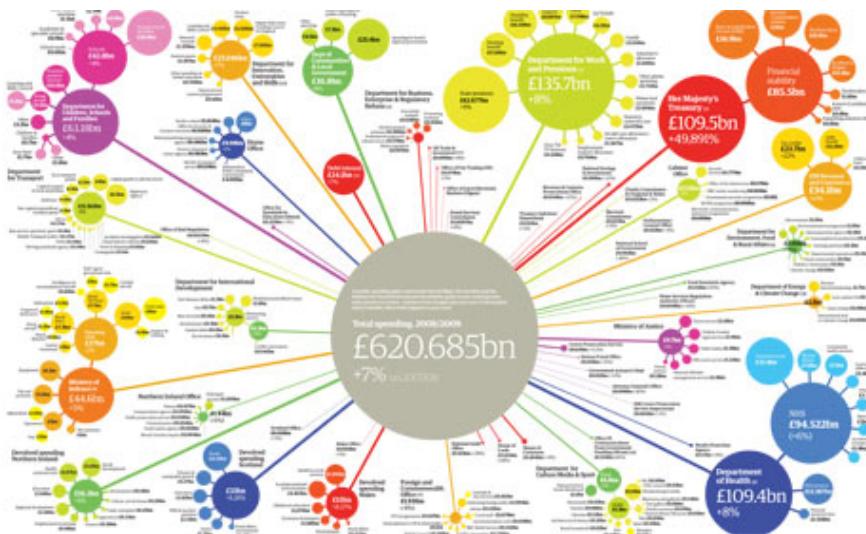
■ Hierarki Gelembung (*Bubble Hierarchy*)

Tipe data yang dibutuhkan: Banyak kategorikal dan 1 rasio kuantitatif
Variabel Visual: Luas, hue, dan posisi

Bubble hierarchy cocok digunakan menggambarkan kategorikal-kategorikal dengan sebuah struktur hierarki. Sebuah lingkaran atau gelembung pada diagram ini merupakan representasi dari sebuah kategori sedangkan ukuran gelembung pada diagram ini menggambarkan ukuran kuantitatif kategori tersebut dan warna yang sama merepresentasikan masih dalam satu *super* kategori. Contoh visualisasi dapat dilihat pada gambar 2.19.

¹⁷<https://www.dataforvisualization.com/charts/circle-packing/> Diakses Desember 2022

¹⁸<https://www.theguardian.com/news/datablog/2010/may/17/uk-public-spending-departments-money-cuts>
Diakses Desember 2022



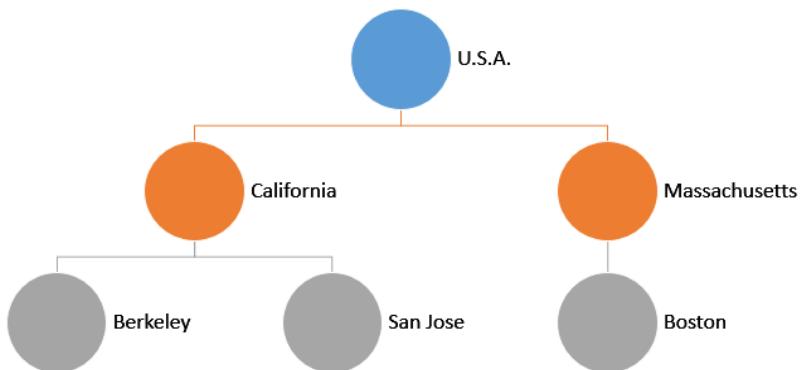
Gambar 2.19: Analogi Hierarki Gelembung (*Bubble Hierarchy*) ¹⁸

■ Hierarki Pohon (*Tree Hierarchy*)

Tipe data yang dibutuhkan: 2 kategorikal dan 1 rasio kuantitatif

Variabel Visual: Luas atau sudut, hue, dan posisi

Sama seperti diagram *bubble hierarchy* pada poin 2.1.3 namun pada diagram ini hierarki digambarkan dalam bentuk pohon. Contoh visualisasi dapat dilihat pada gambar 2.20.



Gambar 2.20: Analogi Hierarki Pohon (*Tree Hierarchy*) ¹⁹

- Menampilkan perbedaan berdasarkan waktu [1]

■ Diagram Garis (*Line Chart*)

Tipe data yang dibutuhkan: 1 interval kuantitatif, 1 rasio kuantitatif, 1 kategorikal

Variabel Visual: Posisi, hue, dan lekukan

Diagram garis digunakan untuk membandingkan sebuah nilai kuantitatif yang berkelanjutan pada sumbu horizontal dan nilai pada sumbu vertikal. Pada diagram ini terdapat titik yang terdapat pada garis untuk menandakan terjadinya perubahan gradien. Peng-

¹⁹<https://berkeley-cs61as.github.io/textbook/hierarchical-structures-capital-t-trees.html> Diakses Desember 2022

gunaan diagram garis dapat membantu menyampaikan cerita mengenai transisi relatif atau nilai kategoris. Diagram garis tidak seperti diagram batang yang nilai sumbu vertikal harus dimulai dari 0, karena pada diagram garis hal atau tujuan utamanya yaitu menampilkan pola relatif dari rentang waktu. Contoh visualisasi dapat dilihat pada gambar 2.21.



Gambar 2.21: Analogi Diagram Garis (*Line Chart*) ²⁰

■ Sparklines

Tipe data yang dibutuhkan: 1 rentang kuantitatif dan 1 rasio kuantitatif
Variabel Visual: Posisi dan lekukan

Sparklines tidak selalu diagram garis. Diagram ini memanfaatkan persepsi manusia dalam membedakan sedikit saja perubahan. Visualisasi jenis ini ditujukan untuk menampilkan visualisasi data yang padat dalam luas yang kecil. Contoh visualisasi dapat dilihat pada gambar 2.22.



Gambar 2.22: Analogi *Sparklines* ²¹

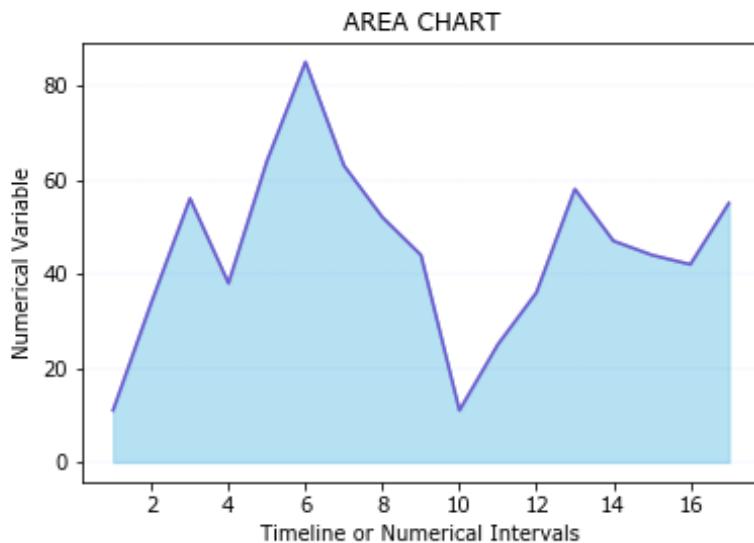
²⁰<https://www.datapine.com/documentation/line-chart> Diakses Desember 2022

²¹<https://www.ag-grid.com/javascript-data-grid/sparklines-overview/> Diakses Desember 2022

■ Diagram Luas (*Area Chart*)

Tipe data yang dibutuhkan: 1 interval kuantitatif, 1 kategorikal, dan 1 rasio kuantitatif
 Variabel Visual: Tinggi, lekukan, luas, dan hue

Diagram luas digunakan untuk menampilkan perkembangan dari waktu ke waktu. Diagram luas memiliki banyak variasi visualisasi pendukung seperti luas yang diberi warna untuk menekankan perubahan. Contoh visualisasi dapat dilihat pada gambar 2.23.



Gambar 2.23: Analogi Diagram Luas (*Area Chart*) ²²

■ *Horizon Chart*

Tipe data yang dibutuhkan: 1 rentang kuantitatif, 1 kategorikal, 2 rasio kuantitatif
 Variabel Visual: Tinggi, lekukan, hue, ketebalan

Diagram ini merupakan variasi dari diagram luas^{2.1.3}. Perbedaan diagram ini dengan diagram luas yaitu diagram ini dapat menampilkan nilai positif dan negatif dalam satu diagram. Nilai negatif dibedakan dengan nilai positif dengan diberikan warna yang berbeda menandakan polaritas negatif. Dengan menggunakan diagram ini maka hanya memerlukan satu buah visualisasi yang menyebabkan dapat menampung lebih banyak informasi dalam satu tampilan dan untuk memfasilitasi perbandingan antar kategorinya. Contoh visualisasi dapat dilihat pada gambar 2.24.

■ Diagram Luas Bertumpuk (*Stacked Area Chart*)

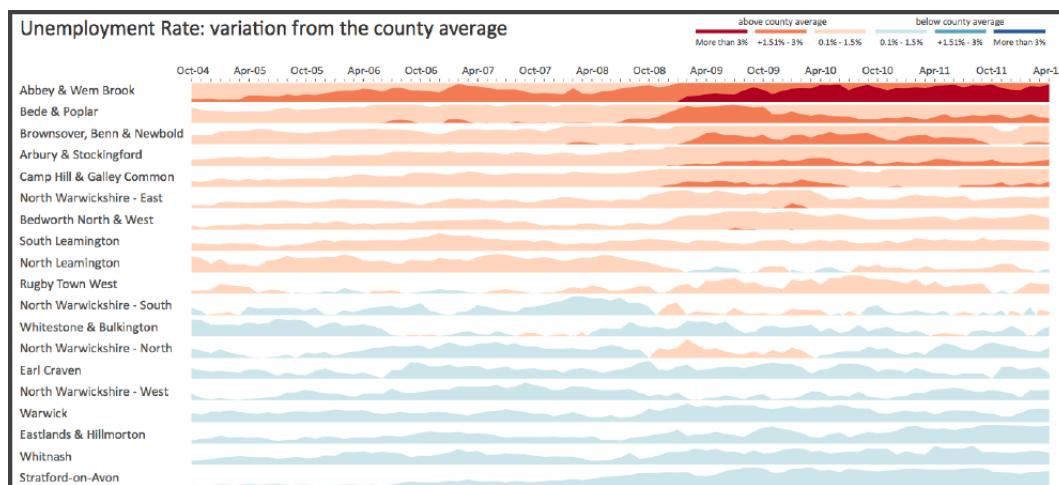
Tipe data yang dibutuhkan: 1 rentang kuantitatif, 1 kategorikal, dan 1 rasio kuantitatif
 Variabel Visual: Tinggi, luas, dan hue

Diagram luas bertumpuk menampilkan perubahan setiap kategoris dari waktu ke waktu. Diagram ini digunakan untuk membandingkan satu kategori dengan kategori-kategori lainnya. Diagram ini membedakan satu kategoris dengan kategoris lainnya berdasarkan perbedaan warna. Contoh visualisasi dapat dilihat pada gambar 2.25.

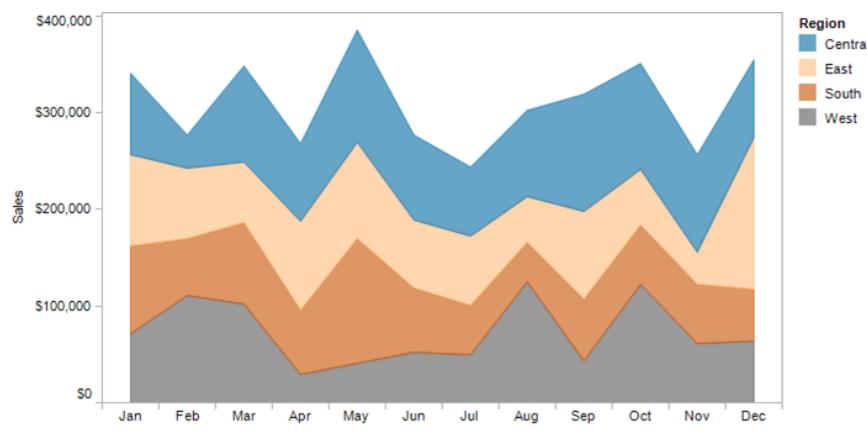
²²<https://towardsdatascience.com/area-charts-standard-stacked-overlapping-d9b77df5dec8> Diakses Desember 2022

²³Kirk, A. Data Visualization: a successful design process, 2st edition. (Birmingham: Packt Publishing,2012), hlm. 139

²⁴<https://www.vizwiz.com/2012/10/stacked-area-chart-vs-line-chart-great.html> Diakses Desember 2022



Gambar 2.24: Analogi *Horizon Chart*²³



Gambar 2.25: Analogi Diagram Luas Bertumpuk (*Stacked Area Chart*)²⁴

■ *Stream Chart*

Tipe data yang dibutuhkan: 1 rentang kuantitatif, 1 kategorikal, 1 rasio kuantitatif
 Variabel Visual: Tinggi, luas, hue

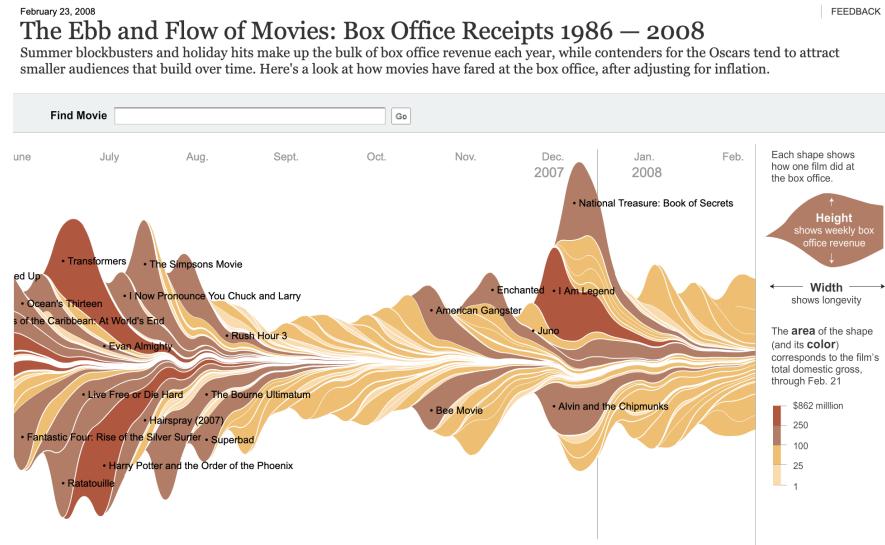
Stream Chart mirip dengan diagram luas bertumpuk^{2.1.3} yang dapat menyajikan data dengan banyak lapisan yang ditumpuk-tumpuk. Perbedaan diagram ini dengan diagram luas bertumpuk adalah pada diagram ini tidak memiliki garis utama pada sumbu vertikal, dan tidak memiliki konsep positif atau negatif karena diagram ini ditujukan untuk menekankan puncak dan lembah. Contoh visualisasi dapat dilihat pada gambar 2.26.

■ *Candlestick Chart* atau *Box Plot* atau *Whiskers Plot*

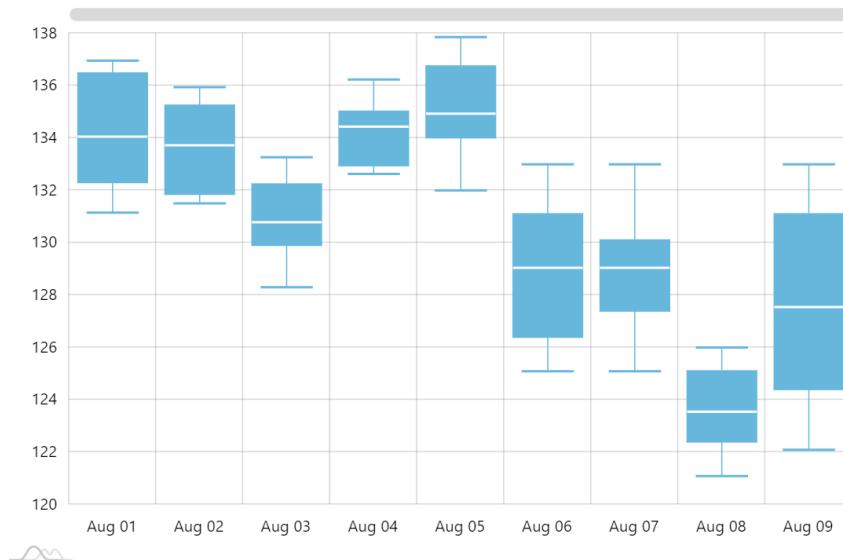
Tipe data yang dibutuhkan: 1 rentang kuantitatif dan 3 kategorikal
 Variabel Visual: Posisi, simbol, hue

Diagram ini umumnya digunakan pada konteks keuangan untuk menampilkan statistik pada pasar saham. Tinggi pada batang tengah menandakan perbedaan antara data pertama dengan data terakhir, sedangkan warna pada diagram ini menandakan apakah

²⁵ <https://www.flourish.studio/blog/streamgraphs/> Diakses Desember 2022

Gambar 2.26: Analogi *Stream Chart*²⁵

data terakhir lebih kecil atau lebih besar dari data pertama. Fokus utama pada diagram ini untuk menampilkan distribusi statistik pada data seperti kuartil atas, kuartil bawah, median, dan lain sebagainya. Contoh visualisasi dapat dilihat pada gambar 2.27.

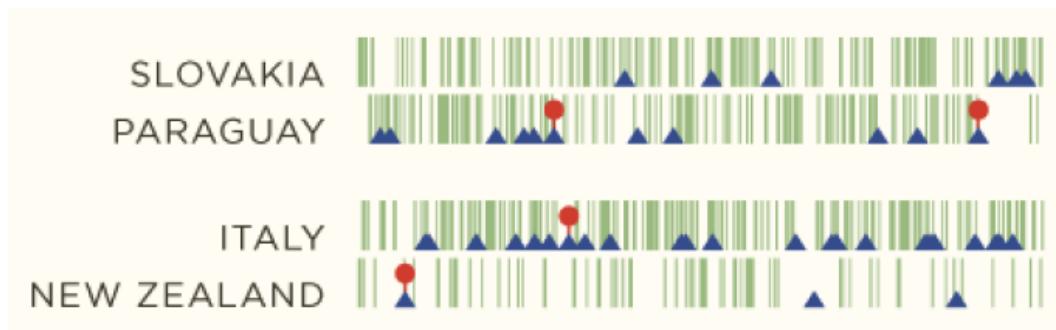
Gambar 2.27: Analogi *Candlestick Chart* atau *Box Plot* atau *Whiskers Plot*²⁶

■ *Barcode Chart*

Tipe data yang dibutuhkan: 1 rentang kuantitatif dan 3 kategorikal
Variabel Visual: Posisi, simbol, dan hue

Barcode Chart merupakan diagram yang sangat padu. Diagram ini menggunakan simbol dan warna untuk menandakan sebuah peristiwa yang terjadi pada suatu rentang waktu. Contoh visualisasi dapat dilihat pada gambar 2.28.

²⁶ <https://www.amcharts.com/demos-v4/box-plot-chart-v4/> Diakses Desember 2022



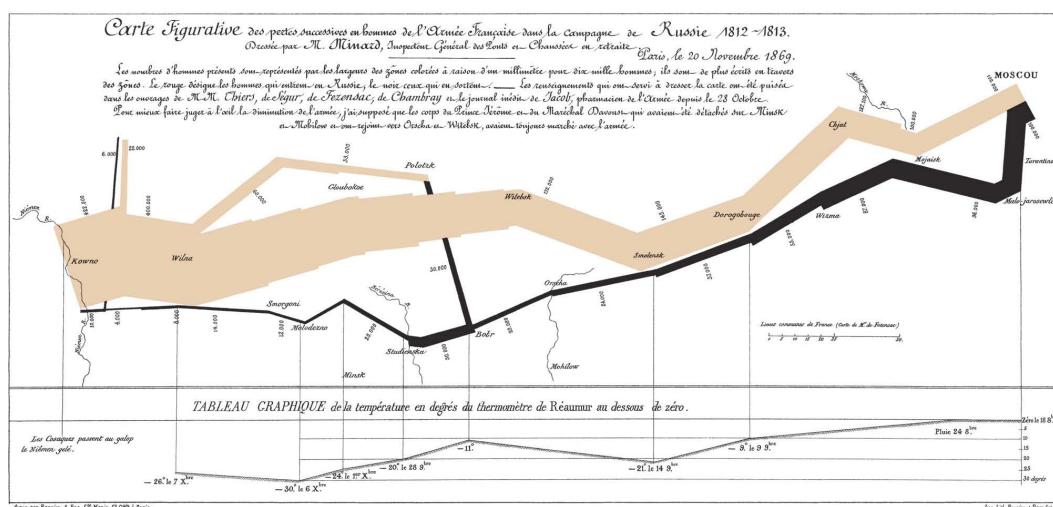
Gambar 2.28: Analogi *Barcode Chart*²⁷

■ Flow Map

Tipe data yang dibutuhkan: Banyak rentang interval, 1 kategorikal, dan 1 rasio kuantitatif

Variabel Visual: Posisi, tinggi atau lebar, dan hue

Flow Map menampilkan aliran nilai kuantitatif yang berubah seiring dengan waktu dan / atau ruang. Contoh visualisasi dapat dilihat pada gambar 2.29.



Gambar 2.29: Analogi *Flow Map*²⁸

- Menampilkan plot koneksi dan hubungan [1].

■ Scatter Plot

Tipe data yang dibutuhkan: 2 kuantitatif

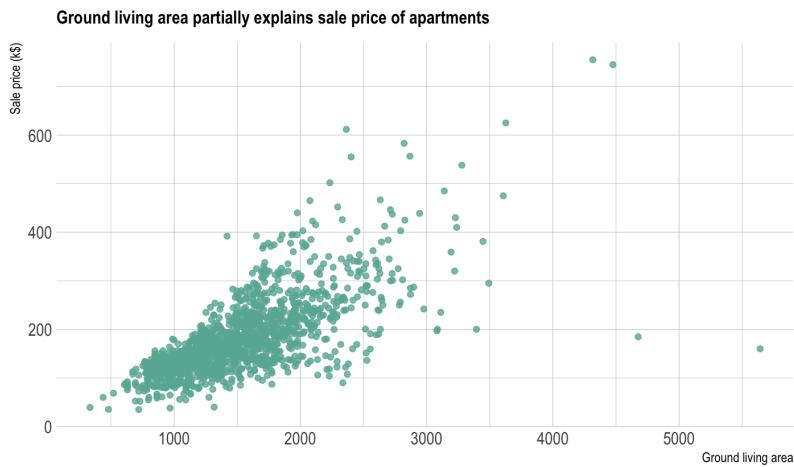
Variabel Visual: Posisi, hue

Scatter plot umumnya digunakan untuk menampilkan pola korelasi, *cluster*, dan penciran. Cara *scatter plot* menampilkan hal-hal tersebut dengan memberikan sebuah titik pada sumbu x dan y yang bersesuaian sehingga *scatter plot* sering digunakan untuk meng-explorasi kumpulan data atau yang sering disebut (*dataset*). Contoh visualisasi dapat dilihat pada gambar 2.30.

²⁷Kirk, A. Data Visualization: a successful design process, 2st edition. (Birmingham: Packt Publishing,2012), hlm. 143

²⁸Kirk, A. Data Visualization: a successful design process, 2st edition. (Birmingham: Packt Publishing,2012), hlm. 143

²⁹<https://www.data-to-viz.com/graph/scatter.html> Diakses Desember 2022

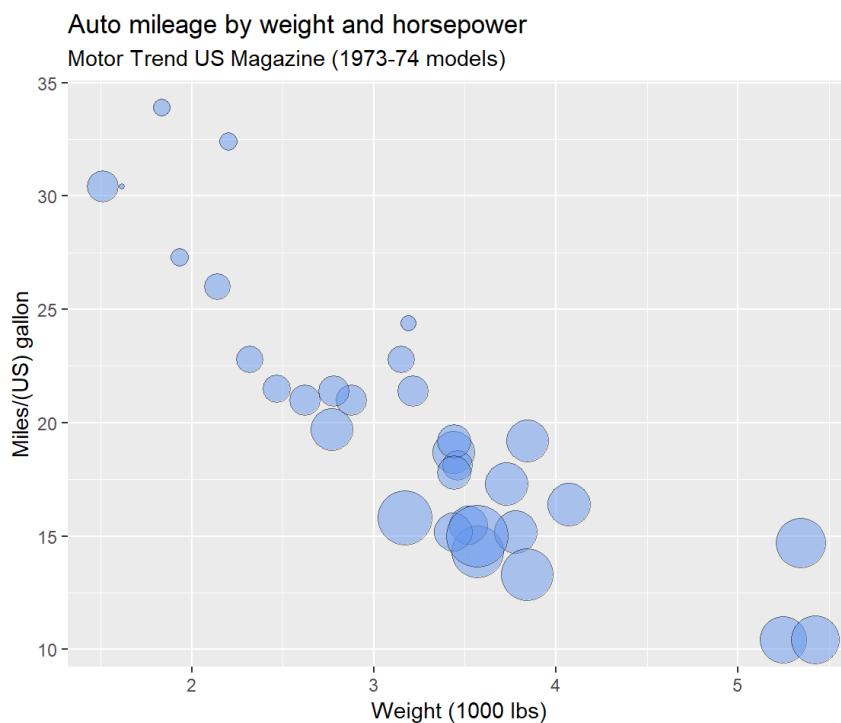


Gambar 2.30: Analogi *Scatter Plot*²⁹

■ Bubble Plot

Tipe data yang dibutuhkan: 2 kuantitatif dan 2 kategorikal
 Variabel Visual: Posisi, luas, dan hue

Bubble Plot merupakan pengembangan dari *scatter plot*, pengembangan yang dilakukan yaitu dapat menampilkan beberapa kategorikal dengan cara membedakan warna yang diberikan. Selain itu ada juga pengembangan berupa ukuran titik atau gelembung bervariasi sesuai dengan data. Contoh visualisasi dapat dilihat pada gambar 2.31.



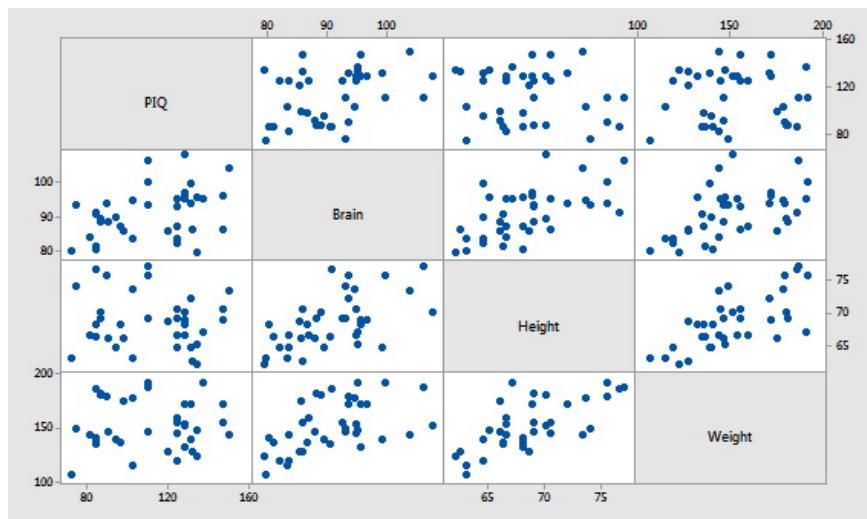
Gambar 2.31: Analogi *Bubble Plot*³⁰

³⁰<https://www.rkabacoff.com/post/better-bubble-charts/> Diakses Desember 2022

■ *Scatter Plot Matrix*

Tipe data yang dibutuhkan: 2 kuantitatif dan 2 kategorikal
 Variabel Visual: Posisi, hue

Scatter plot matrix adalah diagram *scatter plot* namun jumlah *scatter plot*-nya dapat lebih dari hanya satu. Hal ini ditujukan untuk mempermudah pencarian pola pada beberapa kategori. *Scatter plot matrix* mempermudah pencarian pola dengan cara ‘mendekatkan *scattter plot*. Contoh visualisasi dapat dilihat pada gambar 2.32.



Gambar 2.32: Analogi *Scatter Plot Matrix* ³¹

■ Diagram Matriks (*Heatmap*)

Tipe data yang dibutuhkan: Banyak kategorikal dan 1 rasio kuantitatif
 Variabel Visual: Posisi, saturasi warna

Heatmap ditujukan untuk membantu audiens dapat dengan cepat mencocokkan pola untuk mendeteksi urutan kategori dan hierarki kuantitatif dari seluruh kemungkinan kategori. Penggunaan warna pada *heatmap* seperti penurunan saturasi atau kecerahan dapat membantu memperjelas peringkat data. Contoh visualisasi dapat dilihat pada gambar 2.33.

■ *Parallel Sets* atau Koordinat Parallel

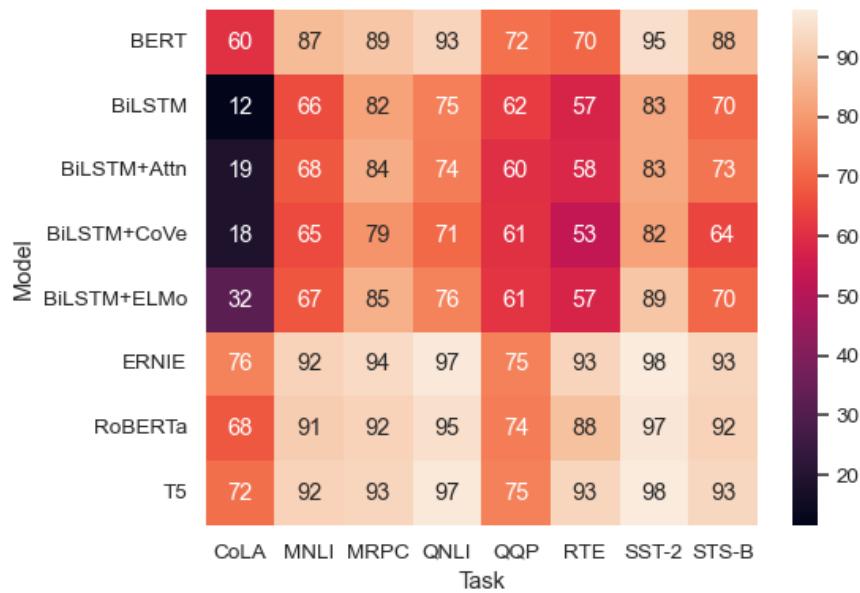
Tipe data yang dibutuhkan: Banyak kategorikal dan banyak rasio kuantitatif
 Variabel Visual: Posisi, lebar, tautan, dan hue

Visualisasi data koordinat parallel menggunakan cara yang unik untuk menampilkan eksplorasi dan analisis terhadap data. Cara kerja visualisasi ini yaitu dengan melakukan plotting seluruh data ke serangkaian sumbu. Kemudian satu per satu variabel disambungkan sehingga seolah-olah terdapat jalur. Jalur-jalur ini lah yang menunjukkan hubungan antara nilai pada data untuk setiap variable. Contoh visualisasi dapat dilihat pada gambar 2.34.

³¹<https://online.stat.psu.edu/stat501/lesson/create-simple-matrix-scatter-plots> Diakses Desember 2022

³²<https://www.fusioncharts.com/dev/chart-guide/standard-charts/heat-map-chart> Diakses Desember 2022

³³<http://www.eagereyes.org/parallel-sets> Diakses Desember 2022

Gambar 2.33: Analogi Diagram Matriks (*Heatmap*)³²

■ Diagram Senar atau *Radial Network*

Tipe data yang dibutuhkan: Banyak kategori dan 2 rasio kuantitatif

Variabel Visual: Posisi, koneksi, lebar, hue, ketebalan, simbol, dan ukuran

Diagram senar ini digunakan untuk membandingkan relasi yang kompleks antar kategori. Diagram ini memiliki keuntungan yaitu tidak ada batasan atau tidak terikat pada sumbu x dan sumbu y. Contoh visualisasi dapat dilihat pada gambar 2.35.

■ Diagram Jaringan

Tipe data yang dibutuhkan: Banyak kategori angka dan 1 rasio kuantitatif

Variabel Visual: Posisi, koneksi, luas, dan hue

Diagram jaringan umumnya digunakan untuk memfasilitasi eksplorasi terhadap data yang kompleks berdasarkan dari kekuatan relasi, koneksi dan logika. Tujuan dari penggunaan visualisasi ini untuk menampilkan pola yang menarik, mengamati kelompok, dan mengamati celah. Contoh visualisasi dapat dilihat pada gambar 2.36.

- Memetakan Geo-spasial Data [1].

■ *Choropleth map*

Tipe data yang dibutuhkan: 2 interval kuantitatif dan 1 rasio kuantitatif

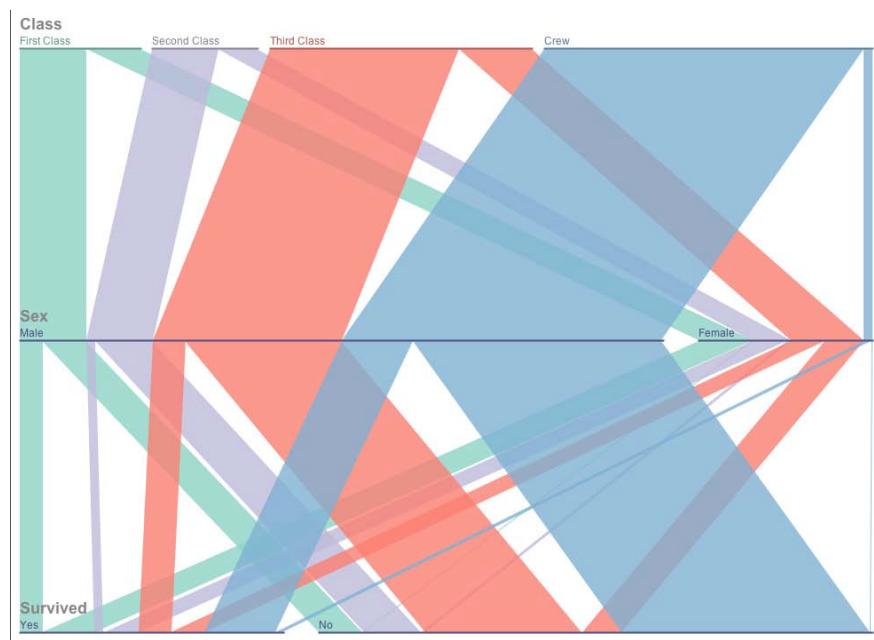
Variabel Visual: Posisi, saturasi atau ketebalan

Diagram ini memberikan warna pada sebuah unit geografis seperti provinsi atau kota. Setiap unit geografis diberikan ketebalan warna atau saturasi berdasarkan nilai kuantitatifnya. Contoh visualisasi dapat dilihat pada gambar 2.37.

³⁴<https://www.truth-and-beauty.net/projects/muesli-ingredient-network/> Diakses Desember 2022

³⁵https://www.researchgate.net/figure/Node-Link-diagram-from-Indiana-University-Wei04_fig2_5852578 Diakses Desember 2022

³⁶<https://www.datavizcatalogue.com/methods/choropleth.html> Diakses Desember 2022



Gambar 2.34: Analogi *Parallel Sets* atau Koordinat Parallel ³³

■ Dot Plot Map

Tipe data yang dibutuhkan: 2 interval kuantitatif

Variabel Visual: Posisi

Dot plot map merupakan visualisasi yang cocok digunakan untuk menampilkan sebaran data geografis. Visualisasi ini pada dasarnya merupakan *scatter plot* yang dikombinasikan dengan posisi garis bujur dan garis lintang ke posisi pada peta. Contoh visualisasi dapat dilihat pada gambar 2.38.

■ Bubble Plot Map

Tipe data yang dibutuhkan: 2 interval kuantitatif, 1 rasio kuantitatif, dan 1 kategorikal-nominal

Variabel Visual: Posisi, area, dan hue

Bubble plot map merupakan visualisasi yang digunakan untuk menampilkan sebaran data kuantitatif pada geografis. Visualisasi ini mengkombinasikan antara *scatter plot* dan *bubble plot* sehingga visualisasi ini menampilkan sebaran gelembung yang ukuran gelembungnya sesuai dengan data kuantitatif, kemudian setiap gelembung diletakkan pada garis bujur dan juga garis lintang yang bersesuaian. Contoh visualisasi dapat dilihat pada gambar 2.39.

■ Isarithmic Map

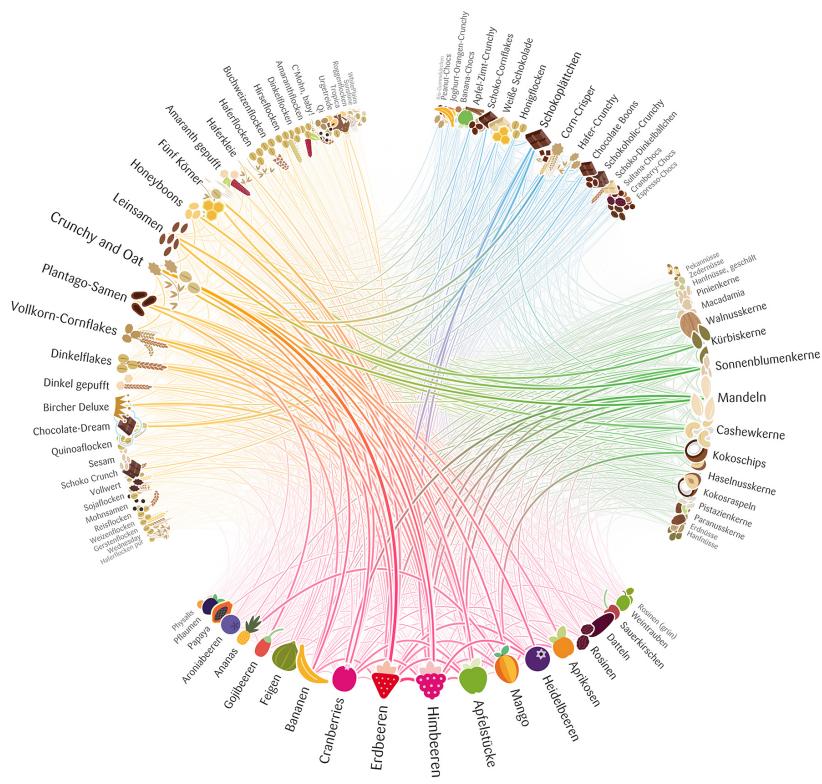
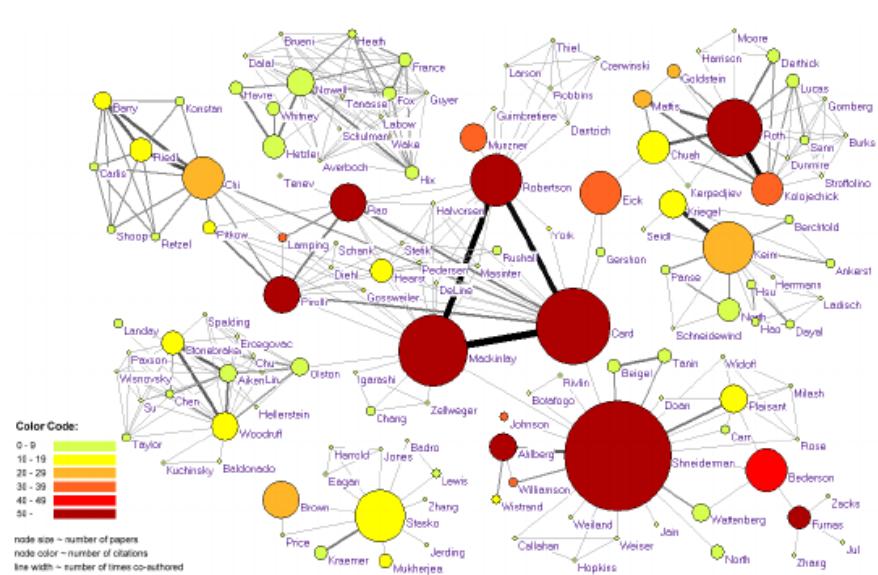
Tipe data yang dibutuhkan: Banyak kuantitatif dan banyak kategorikal

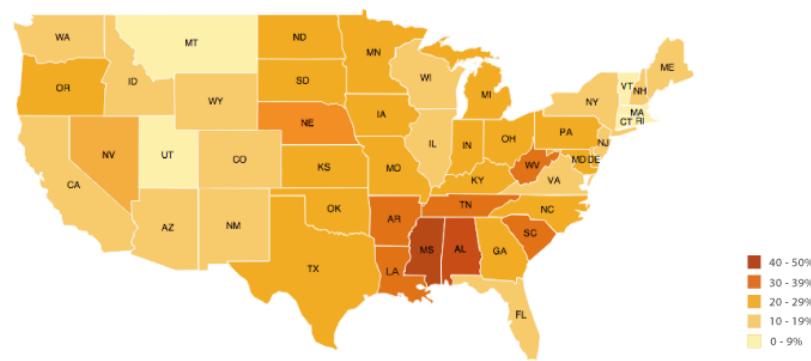
Variabel Visual: Posisi, hue, saturasi, dan ketebalan

Jenis visualisasi data merupakan perkembangan dari visualisasi *choropleth map*^{2.1.3}. Pada *choropleth map* warna diberikan pada sebuah unit geografis, hal ini memiliki kekurangan yaitu pada sebuah unit geografis belum tentu persebarannya merata. Oleh

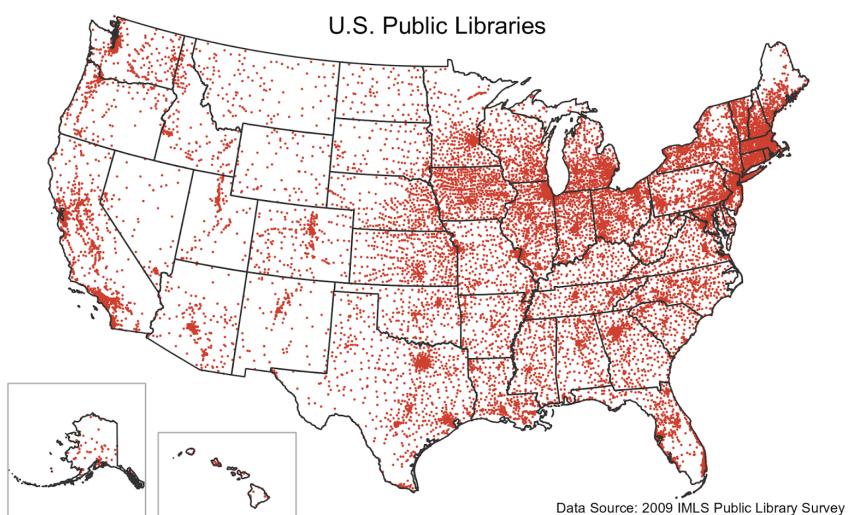
³⁷<https://www.github.com/ResidentMario/geoplot/issues/152> Diakses Desember 2022

³⁸<https://www.community.tableau.com/s/idea/0874T00000HAgfQAG/detail> Diakses Desember 2022

Gambar 2.35: Analogi Diagram Senar atau *Radial Network* ³⁴Gambar 2.36: Analogi Diagram Jaringan ³⁵



Gambar 2.37: Analogi *Choropleth map*³⁶



Gambar 2.38: Analogi *Dot Plot Map*³⁷

sebab itu visualisasi ini mengatasi kekurangan tersebut dengan saturasi warna atau tingkat kegelapan, jika tingkat kegelapan warnanya semakin gelap maka merepresentasikan data atau kepadatan yang lebih tinggi. Contoh visualisasi dapat dilihat pada gambar 2.40.

■ *Particle Flow Map*

Tipe data yang dibutuhkan: Banyak kuantitatif

Variabel Visual: Posisi, arah, ketebalan, kecepatan

Particle flow map merupakan visualisasi yang menggunakan animasi untuk menggambarkan pergerakan data geografis dari waktu ke waktu. Salah satu contoh penggunaan visualisasi yaitu menggambarkan gerakan arus di laut. Contoh visualisasi dapat dilihat pada gambar 2.41.

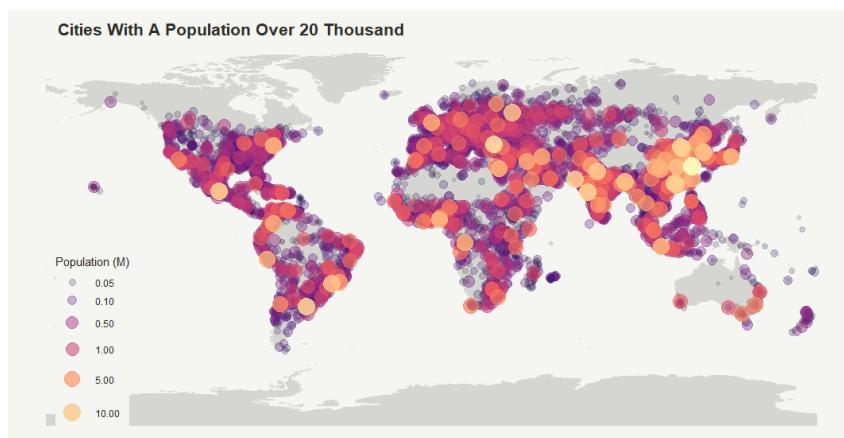
■ *Katogram (Cartogram)*

Tipe data yang dibutuhkan: 2 interval kuantitatif dan 1 rasio kuantitatif

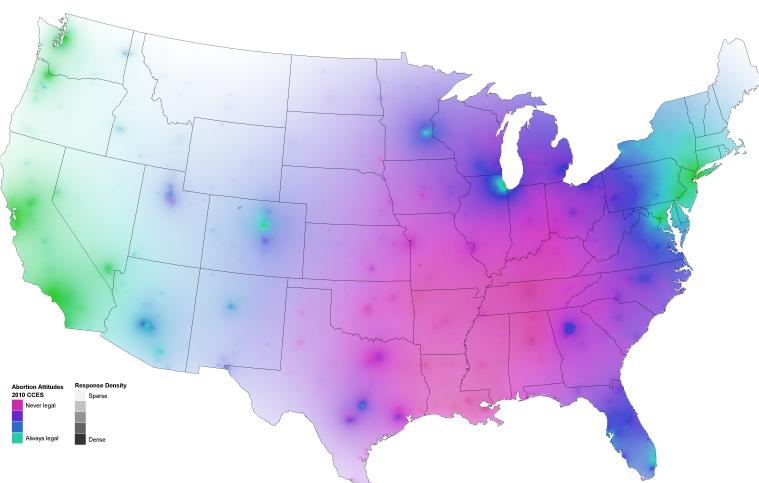
Variabel Visual: Posisi dan ukuran

³⁹<https://www.r-bloggers.com/2011/10/isarithmic-maps-of-public-opinion-data/> Diakses Desember 2022

⁴⁰Kirk, A. Data Visualization: a successful design process, 2st edition. (Birmingham: Packt Publishing,2012), hlm.



Gambar 2.39: Analogi *Bubble Plot Map*³⁸



Gambar 2.40: Analogi *Isarithmic Map*³⁹

Kartogram merupakan jenis visualisasi yang salah satunya digunakan pada atlas. Jenis visualisasi ini mengambil lokasi dan mengubah ukuran untuk merepresentasikan sebuah nilai. Oleh karena itu juga Kartogram tidak ditujukan untuk dibaca secara langsung karena hasilnya terdistorsi dan terdapat *skewed view* dari realitas yang ada. Tujuan dari visualisasi ini adalah untuk memberikan gambaran letak suatu negara, pulau atau geografis lainnya. Contoh visualisasi dapat dilihat pada gambar 2.42.

■ Kartogram Dorling (*Dorling Cartogram*)

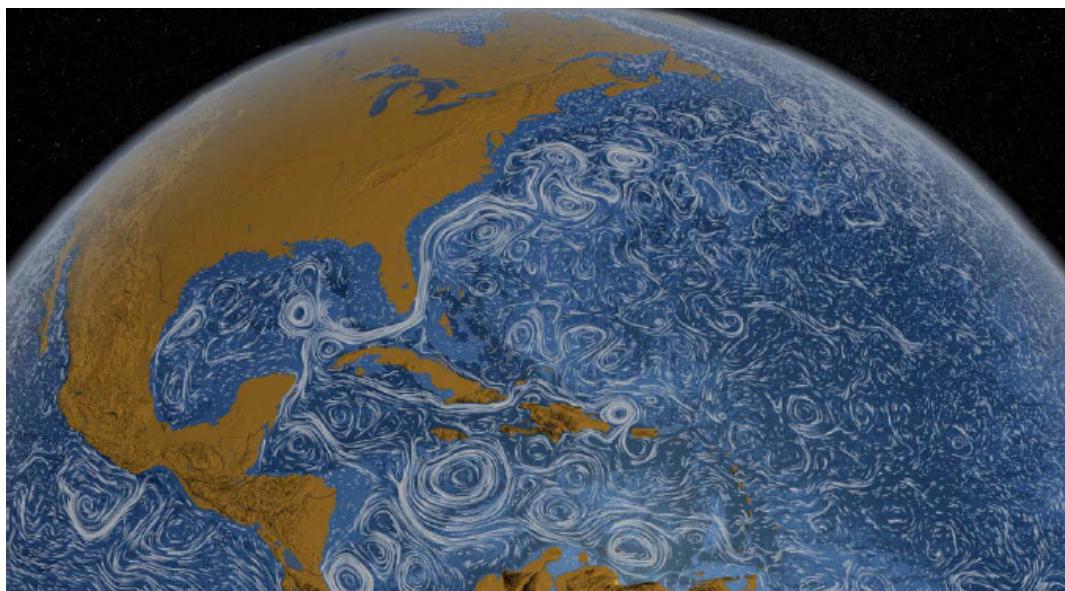
Tipe data yang dibutuhkan: 2 kategorikal dan 1 rasio kuantitatif

Variabel Visual: Posisi, size, hue

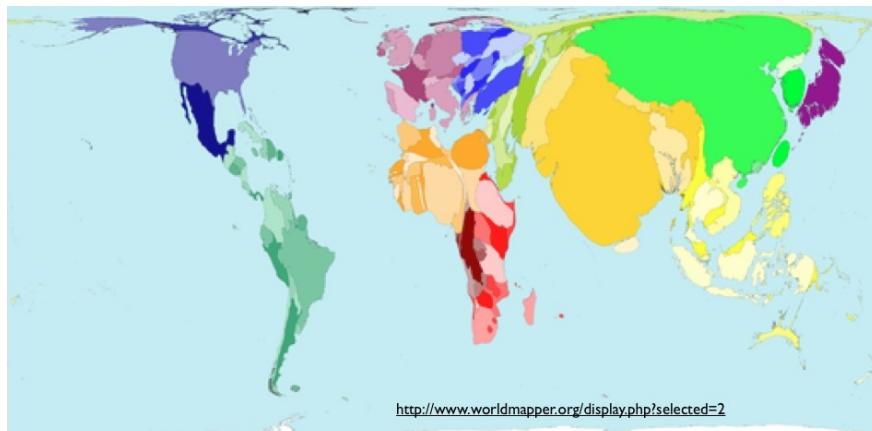
Kartogram Dorling menggunakan bentuk yang seragam (umumnya lingkaran) untuk merepresentasikan lokasi geografis dan memberikan ukuran lingkaran sesuai dengan nilai kuantitatif dari data. Contoh visualisasi dapat dilihat pada gambar 2.43.

⁴¹ <https://www.geocurrents.info/place/europe/microstates-in-cartograms> Diakses Desember 2022

⁴² <https://www.geocurrents.info/place/europe/microstates-in-cartograms> Diakses Desember 2022



Gambar 2.41: Analogi *Particle Flow Map*⁴⁰



Gambar 2.42: Analogi Katogram (*Cartogram*)⁴¹

2.1.4 Aplikasi Visualisasi Data pada WSN

Visualisasi data juga dapat diterapkan pada data yang dihasilkan oleh pengukuran pada WSN. Pada sub-bab ini memuat studi literatur ke hasil yang pernah dibuat oleh pihak lain, baik itu yang bersifat eksperimen atau aplikasi yang sudah diluncurkan.

2.1.4.1 Aplikasi yang bersifat eksperimen

Terdapat beberapa aplikasi yang sudah dibuat oleh pihak lain yang bersifat sebagai eksperimen atau penelitian. Berikut merupakan contoh aplikasi yang dibuat oleh pihak lain:

- Aplikasi yang digunakan untuk memantau suhu diberbagai ruangan menggunakan diagram luas (*area chart*)[5].
- Aplikasi yang digunakan untuk memantau kualitas udara secara *real-time* menggunakan diagram garis (*line chart*) untuk jenis data sensor kadar CO₂ pada udara[6].
- Aplikasi yang digunakan untuk prediksi cuaca, mempelajari iklim dan cuaca dari visualisasi data. Aplikasi ini menggunakan visualisasi data dengan jenis tulisan sederhana, *gauge chart*, dan diagram garis. Pada aplikasi ini jenis tulisan sedehana digunakan untuk jenis data yaitu suhu, curah hujan, kecepatan angin, kelembapan, dan tekanan. Aplikasi ini juga menggunakan



Gambar 2.43: Analogi Kartogram Dorling (*Dorling Cartogram*) ⁴²

diagram garis untuk jenis data sensor suhu, kelembapan, tekanan, kecepatan angin, dan curah hujan. Sedangkan untuk *gauge chart* digunakan untuk data kelembapan, suhu, dan tekanan^[7].

- Aplikasi yang digunakan untuk menampilkan visualisasi data dan mengaktifkan kipas pada kandang rusa. Visualisasi data yang digunakan adalah diagram garis untuk data kelembapan^[8].

2.1.4.2 Aplikasi yang Sudah Diluncurkan

Terdapat juga beberapa aplikasi sejenis yang sudah diluncurkan

- Aplikasi Weather oleh Apple⁴³

Aplikasi Weather merupakan aplikasi visualisasi data cuaca buatan Apple untuk perangkat buatan Apple dengan sistem operasi iOS atau iPadOS. Aplikasi Weather menampilkan visualisasi data sebagai berikut:

- *Isarithmic map* digunakan untuk menampilkan visualisasi data sensor curah hujan.
 - Diagram area digunakan untuk menampilkan visualisasi data sensor Kelembapan, suhu, tekanan, indeks UV, dan jarak pandang.
 - *Gauge chart* digunakan untuk menampilkan visualisasi data tekanan dan index UV.
 - *Floating bar* atau *Gantt chart* digunakan untuk menampilkan visualisasi data suhu.
 - Teks sederhana digunakan untuk menampilkan visualisasi data sensor curah hujan, indeks UV, kelembapan dan jarak pandang.
- The Weather Channel: forecast oleh The Weather Channel Interactive ⁴⁴
Aplikasi ini merupakan aplikasi yang dimiliki oleh TWC Product and Technology, LLC yang merupakan anak perusahaan dari IBM. Aplikasi ini dapat diunduh dan digunakan pada perangkat bergerak dengan sistem operasi iOS, iPadOS, macOS, tvOS, dan Android. Aplikasi ini menampilkan visualisasi data sebagai berikut:
 - Diagram Batang digunakan untuk menampilkan kemungkinan hujan.
 - Diagram luas digunakan untuk menampilkan visualisasi data sensor suhu.
 - *Isarithmic map* digunakan untuk menampilkan data sensor *dew point*, suhu, dan juga index UV.

⁴³<https://www.apps.apple.com/id/app/weather/id1069513131?l=en> Diakses November 2022

⁴⁴<https://apps.apple.com/id/app/the-weather-channel-forecast/id295646461?l=en> Diakses November 2022

- Diagram lingkaran digunakan untuk menampilkan visualisasi data sensor suhu, index UV, perkiraan hujan, dan untuk kondisi udara yang meliputi data sensor O₃, CO, PM10, SO₂, dan NO₂.
- Tabel digunakan untuk memvisualisasikan data prediksi hujan, suhu, kelembapan, dan kecepatan angin.
- CARROT Weather: Alerts & Radar oleh Grailr LLC⁴⁵
CARROT Weather: Alerts & Radar adalah aplikasi milik Glailr LLC. Aplikasi ini dapat diunduh dan digunakan pada perangkat bergerak dengan sistem operasi iOS, IPadOs, dan Android. Aplikasi ini menampilkan visualisasi data sebagai berikut:
 - Diagram batang digunakan untuk menampilkan data sensor curah hujan setiap satu jam, indeks UV tiap jam, kelembapan.
 - Diagram titik atau *dot plot* untuk data sensor suhu.
 - Diagram garis untuk menampilkan data sensor kecepatan angin.
- AccuWeather: Weather Radar oleh AccuWeather⁴⁶
AccuWeather: Weather Radar merupakan aplikasi yang dibuat oleh AccuWeather. AccuWeather sendiri merupakan sebuah perusahaan media memiliki fokus pada informasi-informasi cuaca / data kondisi cuaca di seluruh dunia. Aplikasi ini dapat digunakan disistem operasi iOS, IPadOs, macOS, tvOS, dan juga Android. Aplikasi ini menampilkan visualisasi data sebagai berikut:
 - Tabel digunakan untuk menampilkan data harian yang isinya ada prediksi hujan serta curah hujan jika hujan, suhu, dan suhu terasa. Selain itu tabel juga digunakan untuk menampilkan kondisi saat ini, di dalamnya terdapat data suhu, kecepatan angin, kecepatan angin, kelembapan, *dew point*, tekanan, dan lain sebagainya.
 - Diagram Batang untuk menampilkan data kondisi udara yang telah dikategorikan menjadi rendah, sedang, tinggi, sangat tinggi, dan ekstreme.

2.2 Wireless Sensor Network

Wireless sensor network(WSN) merupakan sekumpulan node sensor yang terhubung dalam satu jaringan komputer nirkabel. Setiap node sensor saling berkomunikasi antara satu dengan lainnya untuk memantau atau mengukur kondisi suatu lingkungan (sensing) sesuai dengan jenis sensornya. Data-data hasil pemantauan dari setiap sensor akan diteruskan menuju *base station* atau *sink*. Base station sendiri merupakan perangkat tempat data diproses lebih lanjut untuk diteruskan ke aplikasi yang menggunakan[9].

2.2.1 Node Sensor

Sensor didefinisikan oleh para ahli sebagai sebuah perangkat yang dapat menerima stimulus *input physical* dari lingkungan untuk diubah menjadi elemen atau energi lain yang terukur dan dapat diobservasi. Sebuah sensor bekerja dengan menerima *input physical* dari lingkungan kemudian diubah menjadi energi listrik oleh *transducer*[10].

2.2.1.1 Komponen Node Sensor

Sebuah *Node Sensor* biasanya terdiri dari empat bagian yaitu [10]

- *Sensing Unit*
Fungsi *Sensing unit* merupakan komponen yang terdiri dari konverter dari sinyal analog ke sinyal digital atau yang dikenal sebagai *analog-to-digital converters*(ADCs).

⁴⁵<https://play.google.com/store/apps/details?id=com.grailr.carrotweather> Diakses November 2022

⁴⁶<https://play.google.com/store/apps/details?id=com.accuweather.android> Diakses November 2022

- *Processing Unit*

Processing unit atau *embedded processor* biasanya pada WSN menggunakan *micro-controller* untuk melakukan penjadwalan, komputasi data, dan mengontrol fungsionalitas dari komponen lain pada *node sensor*.

- *Communication Unit*

Communication unit merupakan komponen yang bertugas untuk berkomunikasi dengan *node* lain ataupun *base station* menggunakan sinyal radio jarak pendek.

- *Power Unit*

Power unit merupakan komponen yang bertugas untuk memberikan daya yang dibutuhkan oleh komponen-komponen lainnya dalam *node sensor*. Pada lingkup WSN biasanya *power unit* merupakan sebuah baterai.

2.2.1.2 Jenis-jenis Sensor

Setiap sensor memiliki fungsi mendeteksi suatu parameter yang berbeda-beda. Tabel 2.2 menunjukkan contoh sensor dan jenis parameternya [11].

Tabel 2.2: Tabel Jenis Sensor

Jenis	Contoh Sensor
Suhu	Termistor, Termokopel
Tekanan	Pengukur tekanan, Barometer, Pengukur ionisasi
Optik	Fotodioda, Sensor inframerah, sensor CDC
Akustik	Resonator piezoelektrik, Mikrofon
Mekanik	Pengukuran regangan, Sensor Taktil, Kapasitif diafragma
Getaran	Accelerometers, Gyroscopes
Arus / Aliran	Anemometer, Sensor aliran massa udara
Posisi	GPS, Sensor berbasis ultrasound, Sensor inframerah
Kimiawi	Sensor pH, Sensor elektrokimia, Sensor gas inframerah
Kelembaban	Sensor kapasitif dan resistif, Higrometer, Sensor kelembapan berbasis MEMS
Radiasi	Deteksi ionisasi, Penghitungan Geiger-Mueller

2.2.2 Protokol Stack WSN

Protokol stack pada WSN terdiri dari lima buah *layer* dan tiga buah manajemen seperti pada gambar 2.44.

Tiga manajemen tersebut berguna untuk mengatur operasional kerja WSN. Setiap manajemen tersebut memiliki fungsinya masing-masing yaitu:

- *Power Management Plane*

Bertugas untuk mengatur energi yang dibutuhkan oleh *node sensor* dalam menjalankan fungsi-fungsinya seperti *sensing*, komputasi, transimisi sinyal, dan *reception*.

- *Connection Management Plane*

Bertugas untuk mengatur atau melakukan konfigurasi ulang koneksi antar *node sensor* pada saat terjadi perubahan seperti penambahan maupun pengurangan *node sensor*.

- *Task Management Plane*

Bertugas untuk membagi-bagi komputasi pada *node-node* sensor agar lebih efisien dalam penggunaan energi maupun waktu.

Lima *layer* tersebut berguna untuk mengatur komunikasi antar *node*. Setiap *layer* memiliki fungsinya masing-masing yaitu:

⁴⁷ <https://sindarku.wordpress.com/2010/08/13/protocol-stack-for-wireless-sensor-networks-wsns/> Diakses Desember 2022

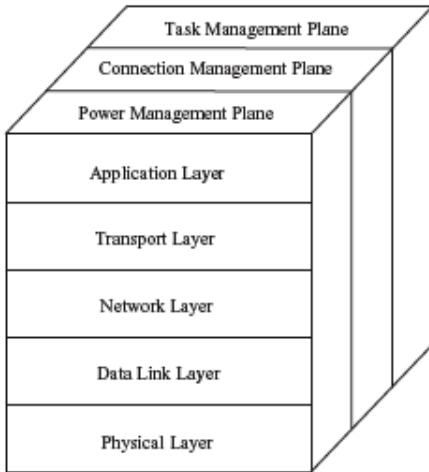


Fig. 2.8 Protocol stack for sensor networks.

Gambar 2.44: Protokol Stack WSN⁴⁷

1. *Physical Layer*

Berfungsi untuk mengubah bit *stream* agar dapat ditransmisikan sesuai dengan media komunikasi.

2. *Data Link Layer* Bertugas untuk menentukan bagaimana bit dikelompokkan menjadi *frame*, mengamankan atau mencegah terjadinya tabrakan paket, melakukan pengiriman ulang jika terindikasi kesalahan, serta MAC yang digunakan untuk menjamin agar *node-node* sensor bekerja dengan adil dan efisien.

3. *Network Layer*

Bertugas untuk membuat memetakan rute dari *node* sensor menuju *base station* atau sebaliknya. Pada *layer* ini juga termasuk menentukan apakah *single hop* atau *multi hop*.

4. *Transport Layer*

Bertugas dalam mengirimkan data di dalam WSN. Terdapat dua jenis pengiriman yang mungkin terjadi yaitu dari *base station* menuju *node* sensor yang disebut sebagai *downstream* dan kebalikannya yaitu dari *node* sensor menuju *base station* yang disebut sebagai *upstream*.

5. *Application Layer*

Bertugas untuk mengatur fungsi dari pengaplikasian *node* sensor seperti query dissemination, alokasi sensor, sinkronisasi waktu, dan keamanan jaringan.

2.2.3 Standar Protokol IEEE 802.15.4 dan Zigbee

WSN umumnya beroperasi dengan bertenaga energi listrik yang berasal dari baterai. Energi listrik yang terkandung dalam baterai terbatas sehingga baterai memerlukan pengisian ulang energi pada saat energi listrik yang disimpan pada baterai habis. Atas dasar tersebut dirancang sebuah protokol yang hemat energi. Standar protokol ini diberi nama atau kode IEEE 802.15.4. Pada standar protokol ini pengiriman data diatur memiliki kecepatan rendah yaitu 20 hingga 250 kilo bit per detik agar lebih ringan dalam beroperasi yang menyebabkan lebih hemat daya. [12, 13]

Zigbee merupakan standar protokol komunikasi yang berdiri diatas atau tambahan dari standar protokol IEEE802.15.4. Protokol IEEE 802.15.4 mengatur lapisan fisik (*physical layer*) dan lapisan MAC, sedangkan untuk protokol Zigbee ditambahkan lapisan jaringan dan lapisan aplikasi. Penggunaan Zigbee pada WSN memiliki beberapa keuntungan yaitu,

- Keuntungan hemat daya

Zigbee menghemat daya dirancang hemat daya dengan cara membatasi kecepatan yaitu 20 hingga 250 kilo dan lebih banyak dalam fase *sleep*. Dengan rancangan yang hemat daya tersebut

memungkinkan sebuah perangkat yang mengimplementasikan Zigbee untuk beroperasi hingga tahunan tanpa mengganti atau mengisi ulang baterai[12, 13].

- *Reliable*

Zigbee disebut *reliable* karena menggunakan standar protokol IEEE802.15.4 yang memiliki jalur cadangan sehingga jika satu jalur dianggap bermasalah maka ada jalur cadangan yang dapat digunakan. Selain itu Zigbee juga menggunakan CSMA-CA untuk menghindari terjadinya penggunaan jalur secara bersamaan yang dapat menyebabkan adanya kemungkinan paket yang hilang [12, 13].

- Rendah biaya

Zigbee disebut rendah biaya karena standar protokol ini hanya melakukan komputasi yang ringan sehingga dapat menggunakan chip *micro-controller* yang murah[12, 13].

- Aman

Zigbee disebut aman karena menggunakan algoritma kriptografi AES-128 untuk mengamankan pertukaran data[12, 13].

- Pengguna yang luas

Zigbee digunakan oleh 250 perusahaan lain dari berbagai negara sehingga Zigbee telah menjalani berhasil menangani berbagai skenario[12, 13].

2.2.4 Sistem Operasi Pada Wireless Sensor Network (WSN)

Sistem operasi pada WSN merupakan lapisan perangkat lunak yang menghubungkan antara *node* sensor dengan aplikasi agar memungkinkan pengembang perangkat lunak dapat melakukan pemrograman di *node* sensor. Sistem operasi pada WSN memiliki tujuan membangun interaksi antara perangkat lunak dengan sumber daya pada perangkat. Terdapat berbagai sistem operasi pada WSN. Berikut beberapa sistem operasi pada WSN dapat dilihat pada tabel 2.3 [14].

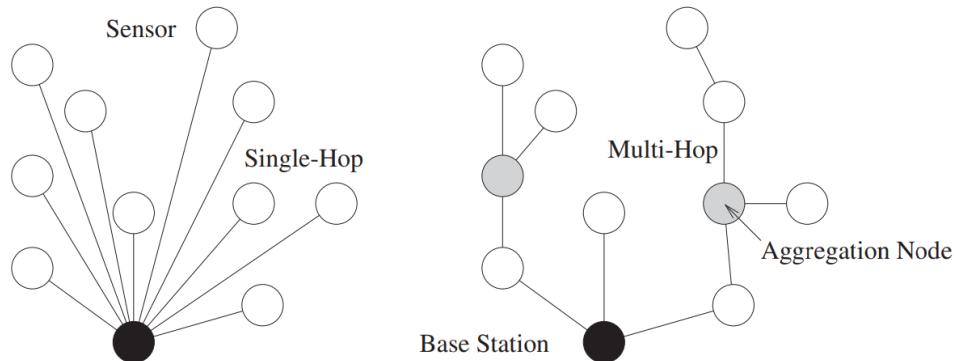
Tabel 2.3: Sistem Operasi pada WSN

Sistem operasi	Paradigma pemrograman	Penjadwalan	Alokasi memori	System call	Pemrograman dinamis	Portabilitas
TinyOS	Berdasarkan kejadian	FIFO	Statis	Tidak tersedia	Memerlukan perangkat lunak tambahan	Tinggi
SOS	Berdasarkan kejadian	FIFO	Dinamis	Tidak tersedia	Tersedia	Menengah ke bawah
Contiki	Berdasarkan kejadian dan dapat menjalankan banyak thread	FIFO	Dinamis	Dengan library	Tersedia	Menengah
LiteOS	Berdasarkan thread	Berdasarkan prioritas seperti round-robin	Dinamis	Tersedia	Tersedia	Rendah

2.2.5 Protokol Komunikasi

Terdapat dua jenis protokol komunikasi yang digunakan pada WSN, yaitu *single-hop* dan *multi-hop*. Perbedaan jenis protokol komunikasi ini dibedakan berdasarkan jumlah *hop* atau langkah untuk mencapai *base station*. Pada protokol *single-hop* atau satu langkah maka setiap *node* sensor

terhubung langsung dengan *base station*. *Single-hop* cocok digunakan pada saat lebar luas daerah yang *sensing* sempit karena jika lebar luas daerah terlalu besar atau jauh maka pengiriman data bisa terganggu seperti lambat, kehilangan koneksi antara *base station* dengan *node* sensor, dan paket hilang. Pada protokol *multi-hop* atau banyak langkah maka ada *node* sensor yang tidak terhubung langsung dengan *base station*. *Node* sensor yang tidak terhubung langsung dengan *base station* mengirimkan paket melalui *node* atau *node-node* tetangganya. Protokol *multi-hop* memiliki kekurangan yaitu memerlukan energi yang banyak/besar, sehingga protokol komunikasi ini harus dipertimbangkan dalam membangun perangkat lunak yang melibatkan WSN [14]. Analogi Protokol komunikasi dapat dilihat pada gambar 2.45.



Gambar 2.45: Protokol komunikasi *Single-hop* (kiri) dan *Multi-hop* (kanan) ⁴⁸

2.2.6 Topologi Wireless Sensor Network (WSN)

WSN memiliki berbagai macam topologi. Setiap topologi memiliki ciri khas, keuntungan masing-masing. Terdapat beberapa topologi jaringan yang umum digunakan antara lain:

1. Poin ke poin

Pada topologi ini hubungan antar *node* dengan *node* lain dilakukan secara langsung, tanpa melalui perantara dan menggunakan sebuah jalur yang hanya terdapat dua buah node.

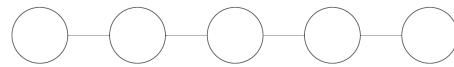


Gambar 2.46: Topologi Poin ke Poin

2. Linear

Topologi linear mirip dengan topologi poin ke poin, hanya saja pada topologi ini hubungan tidak terbatas hanya dua buah *node*. antar *node* dengan *node* lainnya bisa melalui *node* lainnya terlebih dahulu.

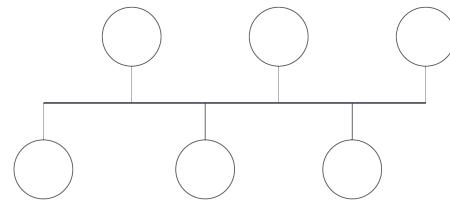
⁴⁸Dargie, W. dan Poellabauer, C. Fundamentals of Wireless Sensor Networks: Theory and Practice, 1st edition. (Chichester:A John Wiley and Sons, Ltd.,2010), hlm. 9



Gambar 2.47: Topologi Linear

3. Bus

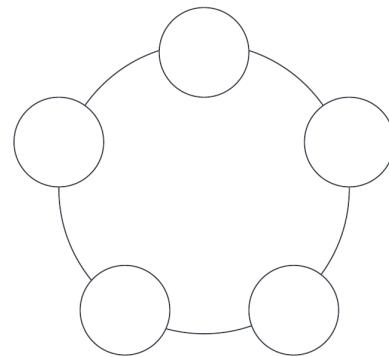
Topologi ini memiliki ciri khas yaitu hanya terdapat satu buah jalur yang menghubungkan banyak *node* sensor. Sebuah sinyal akan dikirimkan ke kedua arah hingga mencapai tujuan. Pada topologi ini harus memiliki "sistem" yang menangani kemungkinan terjadinya tabrakan atau terdapat setidaknya dua buah *node* yang menggunakan jalur bersama-sama.



Gambar 2.48: Topologi Bus

4. Ring

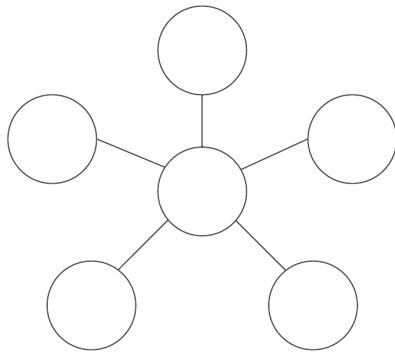
Topologi ring mirip dengan topologi linear tetapi tidak memiliki ujung. Kedua ujung pada topologi linear disambungkan sehingga jika digambarkan akan berbentuk seperti ring.



Gambar 2.49: Topologi Ring

5. Bintang

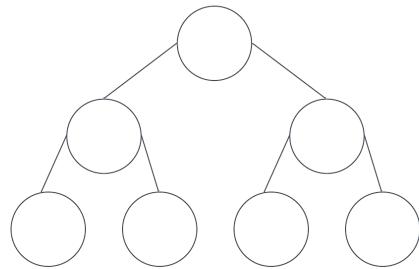
Topologi bintang memiliki sebuah ciri khas yaitu terdapat sebuah *node* tengah yang bertindak sebagai penghubung *node-node* lainnya. Pada topologi ini setiap data harus dikirimkan ke *node* tengah terlebih dahulu, oleh karena itu dibutuhkan *node* tengah yang lebih kuat komputasi. Topologi ini umum digunakan untuk jaringan nirkabel termasuk WSN.



Gambar 2.50: Topologi Bintang

6. Tree(Pohon)

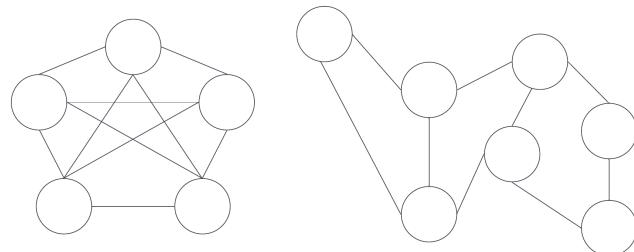
Topologi *tree* merupakan topologi yang memiliki sifat hierarki. Setiap *node* memiliki hierarki tingkatan hierarkinya masing-masing. Topologi ini memiliki keuntungan dalam ekspansi *node* baru dan melakukan isolasi terhadap *node* yang bermasalah. Keuntungan topologi ini dalam ekspansi *node* baru menyebabkan *node* yang memiliki hierarki lebih tinggi cenderung menjadi lebih boros daya.



Gambar 2.51: Topologi Pohon

7. Mesh

Pada topologi mesh setiap *node* terhubung dengan satu atau lebih *node* lainnya. Jika setiap *node* terhubung ke seluruh *node* lainnya disebut dengan *fully connected mesh*, sedangkan jika tidak setiap *node* terhubung dengan seluruh *node* lainnya maka disebut dengan *partially connected mesh*.



Gambar 2.52: Topologi *Fully Connected Mesh* (kiri) dan *Partially Connected Mesh* (kanan)

2.3 Preon32

Preon32 adalah sensor yang dibuat oleh perusahaan asal Berlin, Jerman yaitu Virtenio GmbH. Preon32 menggunakan *virtual machine* khusus yang dibuat perusahaan Virtenio bernama PreonVM. PreonVM memungkinkan pemrogram untuk membuat program dengan bahasa yang berorientasi objek yaitu Bahasa Java. Spesifikasi Perangkat Keras Preon32 dapat dilihat pada tabel 2.4-2.7.

Tabel 2.4: Tabel Spesifikasi *Micro-Controller* Preon32 [3]

<i>Micro controller</i>	
<i>Micro-controller Core</i>	32 Bit Cortex M3 Microcontroller, RISC
<i>Clock Frequency</i>	Hingga 72 MHZ
Memori Sistem	64 kByte SRAM
Memori Data	Hingga 512kByte flash
Konsumsi Arus	3,7 mA (Hidup, 8 MHz), 28,3 mA (Hidup, 72 MHz), 1,3 mA (<i>sleep, no peripherals active</i>), 26 A (Mati total)

Tabel 2.5: Tabel Spesifikasi *High Frequency Transceiver* Preon32 [3]

<i>High Frequency Transceiver</i>	
Pita Frekuensi	2400.0 2483.5 MHz
Saluran	16
Interval Saluran	5 MHz
Lebar Pita Data	250 Kbit/s sampai 2Mbit/s
Output Power	-17 dBm sampai 3 dBm
Sensitivitas	-101 dBm
Jangkauan	Hingga 250 m luar ruangan Hingga 30 m di dalam ruangan
Konsumsi Arus	Sleep mode: 20 nA receive mode: 12.3 mA transmit mode: 14 mA

Tabel 2.6: Tabel Spesifikasi *Interface* Preon32 [3]

Interface	
Serial Interface	1x USB 2.0 Full-speed controller 1x CAN interface (2.0B Active) 3x SPIs, up to 18Mbit.s 3x USARTs, up to 4.5 Mbit/s 1x I ² C interface, up to 400 Kbit/s
Analog interfaces:	2x 12 Bit analog to digital converters 1 MSPS, 15 channels 2x 12Bit digital to analog converters
Other interfaces:	37x digital input or output pins (GPIOs) 15x external interrupt inputs
Data memory (tambahan)	External 8 Mbit serial Flash

Tabel 2.7: Tabel Spesifikasi *General* Preon32 [3]

Umum	
Dimensi	27,5 x 19 x 3,3 mm
Suhu kerja	-40°C hingga +85°C
Tegangan kerja	2,7 - 3,6 V

2.3.1 Spesifikasi Jenis-Jenis Sensor Modul Preon32

Modul Preon32 memiliki empat buah sensor yaitu suhu, tekanan udara, kelembapan, dan akselerasi. Berikut spesifikasi sensor-sensor Preon32:

- Suhu [15]
 - Pembuat : Analog Devices
 - Model :ADT7410
 - Antarmuka :digital, I2C
 - Resolusi :16-Bit
 - Rentang :-40C sampai +105C
 - Akurasi:±0,5C
- Tekanan [15]
 - Pembuat :Freescale
 - Model :MPL115A2
 - Antarmuka :digital, I2C
 - Resolusi :0,15 kPa
 - Rentang :50 kPa sampai 115 kPa
 - Akurasi:±1,0 kPa
- Kelembapan [15]
 - Pembuat :Sensirion
 - Model :SHT21
 - Antarmuka :digital, I2C
 - Resolusi :12-Bit
 - Rentang :0%RH sampai 100%RH
 - Akurasi:±2,0 %RH
- Getaran [15]
 - Pembuat :Analog Devices
 - Model :ADXL345
 - Antarmuka :digital, SPI
 - Resolusi :13-Bit per axis
 - Rentang :±16 g, 3 axis
 - Akurasi:3,9 mg/LSB

2.3.2 PreonVM

PreonVM merupakan *virtual machine* buatan Virtenio yang dirancang khusus agar dapat dijalankan pada perangkat dengan sumber daya terbatas sehingga PreonVM dapat dijalankan pada sensor Preon32. PreonVM dirancang menggunakan Bahasa Java sehingga memiliki beberapa keuntungan yaitu mendukung tipe data primitif Java, dapat menggunakan banyak thread beserta operasi-operasi thread Java, memiliki Garbage Collection, mendukung *exception handling*, dan terdapat *system properties* untuk mengatur Aplikasi.

2.4 HTTP

Hypertext Transfer Protocol —HTTP— adalah sebuah protokol komunikasi yang berada di level aplikasi yang mengatur pengiriman dan penerimaan informasi. Dalam HTTP, *client* selalu membuka koneksi untuk melakukan permintaan ke *server*, setelah melakukan permintaan *client* menunggu untuk mendapatkan balasan dari server[16].

2.4.1 HTTP Methods

HTTP memiliki beberapa perintah yang setiap perintahnya memiliki tujuannya masing-masing. Berikut adalah perintah-perintah yang didukung oleh HTTP[16]:

- GET
Sebuah perintah yang digunakan untuk mengirimkan *resource* dari *server* ke *client*.
- PUT
Sebuah perintah yang digunakan untuk menyimpan data yang diberikan *client* ke *server*.
- DELETE
Sebuah perintah yang digunakan untuk menghapus sebuah *resource* di *server*.
- POST
Sebuah perintah yang digunakan untuk mengirimkan data *client* ke *server*.
- HEAD
Sebuah perintah yang digunakan untuk mengirimkan sebuah balasan *headers* HTTP.
- OPTION
Sebuah perintah yang digunakan untuk menentukan *methods* yang dapat diterima dan diproses oleh server pada URL tersebut.

2.4.2 Kode Status

Setiap balasan HTTP dari server memiliki kode status yang fungsinya untuk memberitahu *client*. Kode status HTTP terdiri dari 3 digit angka. Kode status HTTP secara umum dibagi menjadi 5 bagian yaitu[16]:

- Kode 100-199.
Kode ini digunakan untuk mengirimkan informasi-informasi seperti, masih memproses, meminta untuk beralih ke versi HTTP yang lain, dan lain sebagainya.
- Kode 200-299.
Kode digunakan untuk memberitahu *client* bahwa *server* berhasil memahami dan menyetujui permintaan.
- Kode 300-399
Kode ini digunakan untuk memberitahu *client* bahwa *server* memerlukan tindakan tambahan untuk menjawab permintaan karena adanya perubahan pada *resource* yang diminta *client*.
- Kode 400-499
Kode ini digunakan untuk memberitahu *client* bahwa *server* tidak dapat memenuhi permintaan *client* karena masalah seperti sintaks yang buruk, tidak memiliki atau tidak memberikan kredensial yang valid, tidak ditemukan, dan lain sebagainya.
- Kode 500-599
Kode ini digunakan untuk memberitahu *client* bahwa *server* tidak dapat memenuhi permintaan *client*. Perbedaan kode ini dengan kode 400-499 adalah untuk kode 400-499 kesalahan terjadi di sisi *client* sedangkan kode 500-599 kesalahan terjadi pada sisi *server*.

2.4.3 HTTP Messages

Pesan HTTP secara sederhana dapat dibagi menjadi tiga buah bagian yaitu *status line* atau *start line* yang berisi method, versi http, dan kode status. Bagian kedua yaitu *header* yang berisikan informasi-informasi *header* seperti tanggal, panjang pesan, kapan terakhir diubah, tipe konten , dan lain sebagainya. Bagian ketiga yaitu *body* yang berisikan pesan permintaan atau balasan^[16].

2.5 Application Programming Interface

Application programming interface(API) merupakan antarmuka atau perantara penghubung antara satu aplikasi dengan aplikasi-aplikasi lainnya. Aplikasi yang mengirimkan permintaan atau *request* disebut sebagai klien sedangkan yang melayani permintaan disebut sebagai *server*⁴⁹. API memiliki berbagai arsitektur antara lain,

- SOAP
Arsitektur API soap memiliki ciri utama yaitu selalu mengirimkan balasan permintaan dengan format *Extensible Markup Language* (XML).
- REST
Arsitektur API REST memiliki ciri yaitu fleksibel karena arsitektur ini mendukung semua HTTP method dan dapat mengembalikan balasan dalam berbagai format.
- RPC
Arsitektur API RPC memiliki ciri utama yaitu berorientasikan aksi. Pada arsitektur ini memungkinkan antara komputer untuk menjalankan aksi di komputer lain.

2.6 Library dan Framework

Library adalah kumpulan kode atau modul yang dapat dipakai di kode lain untuk membantu menyelesaikan tugas yang umum. sedangkan *framework* adalah sebuah kerangka kerja yang digunakan untuk membangun sebuah aplikasi.

2.6.1 Bootstrap

Bootstrap adalah *framework* yang paling banyak digunakan berdasarkan survei yang dilakukan oleh Matthias Gelbmann pada 3 Maret 2022 ⁵⁰. Bootstrap umumnya digunakan untuk mempercepat proses pembuatan atau pengembangan website ataupun untuk mempercantik tampilan website. Bootstrap memiliki beberapa keunggulan utama yaitu responsif, memiliki komponen UI yang sudah dirancang menarik, dan lain sebagainya⁵¹.

2.6.2 Retrofit

Retrofit adalah library yang dapat digunakan untuk melakukan HTTP *request* pada bahasa pemrograman Java. Retrofit memiliki beberapa keunggulan yaitu mendukung berbagai HTTP *methods*, dapat memanipulasi *Header HTTP*, bekerja secara *asynchronous*, konversi permintaan dan balasan secara otomatis, dan lain sebagainya⁵².

⁴⁹<https://aws.amazon.com/id/what-is/api/> Diakses Juni 2023

⁵⁰https://w3techs.com/blog/entry/new_survey_on_css_frameworks Diakses Juni 2023

⁵¹<https://getbootstrap.com/docs/5.3/getting-started/introduction/> Diakses Juni 2023

⁵²<https://github.com/square/retrofit> Diakses Juni 2023

2.6.3 Plotly

Plotly khususnya plotly.js adalah library yang dapat digunakan untuk menampilkan visualisasi data dalam bentuk grafik. Plotly memiliki beberapa keunggulan yaitu interaktif dan dinamis yang artinya dengan menggunakan Plotly, pengguna dapat berinteraksi dengan grafik. Interaksi yang mungkin dilakukan dengan plotly yaitu *hover*, *zoom*, geser, dan bisa *realtime*. Selain itu Plotly juga mendukung berbagai bahasa pemrograman seperti Python, R, dan lain sebagainya ⁵³.

2.6.4 Fast API dan Uvicorn

Fast API adalah sebuah *framework* untuk membuat API dengan bahasa Python. Fast API memiliki beberapa keunggulan antara lain memerlukan waktu yang singkat untuk menuliskan kode, menurunkan kemungkinan kesalahan developer, digunakan oleh perusahaan besar seperti Microsoft dan Netflix⁵⁴.

Uvicorn merupakan sebuah antarmuka standar untuk komunikasi antara web server dan aplikasi dengan bahasa Python. Uvicorn memiliki beberapa keunggulan antara lain memiliki performa tinggi karena library uicorn dapat memproses permintaan secara asinkron, terdapat pencatatan dan penanganan error, dan dapat dikonfigurasi sesuai kebutuhan seperti jumlah *threads* yang digunakan, *port* yang dipakai, dan lain sebagainya⁵⁵.

2.6.5 Laravel

Laravel adalah sebuah framework PHP populer yang digunakan untuk membuat sebuah aplikasi berbasis website. Laravel memiliki beberapa kelebihan yaitu mempercepat waktu pengembangan aplikasi, pemrogram menuliskan kode uang lebih sedikit, aplikasi yang lebih aman, adanya arsitektur Model-View-Controller(MVC,dan keunggulan-keunggulan lainnya ⁵⁶). Laravel memiliki beberapa fitur unggulan yaitu:

- *Routing*

Laravel memiliki sistem routing yang simple tanpa memerlukan pemrogram secara manual membuat file-file konfigurasi *routing*.

- Sistem *Template*

Laravel memiliki sistem *template* yang disebut sebagai Blade. Blade memungkinkan pemrogram menuliskan kode secara tidak berulang-ulang, selain itu Blade juga mempermudah penulisan kode seperti pengkondisian, autentikasi, dan lain sebagainya.

- Menyediakan berbagai *Library*

Laravel menyediakan banyak *library* seperti library yang dapat digunakan untuk pengiriman email, yang menghubungkan antara aplikasi dengan basis data dan lain sebagainya.

- Sistem cache

Laravel menyediakan fitur *caching* yang dapat digunakan untuk membantu mempercepat proses pemrosesan. *Cache* dalam Laravel dapat diatur secara manual mengenai apa saja file atau data yang hendak dilakukan maupun tidak dilakukan *cache*.

- *Error handling* dan *logging*

Laravel menyediakan *error handling* dalam artian jika terjadi *error* Laravel akan menampilkan laporan *error* yang jelas. Selain itu Laravel juga akan mencatat segala aktivitas.

2.6.6 Fetch

Fetch API adalah antarmuka yang dapat digunakan untuk mengirimkan permintaan atau *request* HTTP ke server. Fetch API memiliki beberapa kelebihan yaitu, berjalan secara asinkronus sehingga

⁵³<https://github.com/plotly/plotly.js/> Diakses Juni 2023

⁵⁴<https://fastapi.tiangolo.com/lo/> Diakses Juni 2023

⁵⁵<https://www.uicorn.org/> Diakses Juni 2023

⁵⁶<https://laravel.com/docs/10.x> Diakses Juni 2023

tidak terjadi *blocking* sepanjang melakukan permintaan ke server, lebih mudah karena fetch API sudah secara otomatis mengelola *header*, lebih mudah dalam menangani balasan *server* karena Fetch api menyediakan fungsi yang dapat mengkonversi format-format yang umum seperti JSON, *formData*, *text*, dan lain sebagainya⁵⁷.

2.7 Bahasa Pemrograman

Bahasa Pemrograman merupakan sekumpulan instruksi komputer yang dapat komputer jalankan. Bahasa pemrograman dibagi menjadi tiga tingkatan yaitu level rendah, menengah, dan tinggi. Level rendah atau yang dikenal sebagai bahasa mesin *machine language* merupakan bahasa yang hanya mengenal angka biner saja untuk menjalankan operasi komputer. Level menengah adalah level dimana instruksi sudah terdapat simbol-simbol seperti huruf, angka dan karakter spesial. Kemudian ada level tinggi yaitu level dimana bahasa pemrograman sudah dapat dibaca dan dimengerti oleh manusia karena sudah cukup mirip dengan bahasa manusia[17].

2.7.1 Bahasa Pemrograman Java

Secara umum, Bahasa Java dirancang berorientasi objek, sederhana, memiliki keamanan tinggi, dan dapat bekerja di berbagai tipe prosesor. Java saat ini masih banyak digunakan diberbagai Organisasi antara lain Uber, Google, Netflix, Instagram, Spotify, Amazon, Slack, dan lain sebagainya. Bahasa Java memiliki banyak keuntungan antara lain sebagai berikut [18, 17]:

1. Sekali buat, jalankan dimanapun. Java dapat berjalan dengan baik diberbagai perangkat atau sistem operasi.
2. Java didesain memiliki keamanan yang baik. Java memisahkan sistem *host* dengan sistem Java sehingga jika terjadi sesuatu dengan sistem Java maka sistem *host* tidak berpengaruh.
3. Java didesain *network-centric*. Java didesain agar pemrogram dapat membuat program yang *network-based applications* dengan mudah.
4. Java didesain dinamis. Java didesain berorientasi objek yang disebut *clases*. Setiap *clases* disimpan dalam file yang terpisah. Hal ini memungkinkan Java untuk hanya menggunakan file-file yang dibutuhkan saja.
5. Java menggunakan format internasional. Java dari awalnya sudah didesain menggunakan 16-bit Unicode sehingga Java bisa mewakili lebih banyak karakter jika dibandingkan dengan bahasa yang pada saat Java dibuat umumnya menggunakan 8-bit.
6. Java didesain agar pemrogram dapat membuat kode lebih cepat. Java didesain memiliki struktur sederhana tetapi tetap tidak meninggalkan aspek elegant dan fungsionalitasnya.
7. Java memiliki memori manajemen. Java memiliki sebuah memori manajemen yang disebut Garbage Collection atau terkadang disebut Garbage Collector. Garbage Collection ini memiliki fungsi untuk mengurangi penggunaan memori dengan cara melepaskan objek yang sudah tidak digunakan lagi.

2.7.2 Bahasa Pemrograman Python

Bahasa Python dibuat oleh Guido van Rossum pada tahun 1989. Bahasa Python pertama kali dikembangkan karena bahasa-bahasa pemrograman yang ada seperti C, C++, Lisp, Java, dan Perl tidak memiliki *library* yang cukup untuk memudahkan pembuat program. Sehingga Bahasa Python memiliki beberapa fitur yaitu *high level*, berorientasi objek, *reuse*, *extensible*, dapat berjalan di berbagai perangkat, mudah dibaca, mudah dipelihara, kokoh, dan lain sebagainya[19].

⁵⁷https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API Diakses Juni 2023

BAB 3

ANALISIS

Pada bab ini menjelaskan mengenai analisis perangkat lunak yang terdiri dari deskripsi perangkat lunak, analisis fungsi perangkat lunak, analisis kelas diagram untuk *node sensor*, analisis kelas diagram *base station*, analisis kelas diagram server, analisis kelas diagram aplikasi. analisis struktur basis data, dan analisis format pertukaran data.

3.1 Deskripsi Perangkat Lunak

Berdasarkan permasalahan yang terdapat pada latar belakang maka perangkat lunak yang dibangun memiliki fungsi untuk menampilkan visualisasi data sensor yang sesuai antara kebutuhan pengguna, jenis data, dan lokasi WSN berada atau tersedia. Dari visualisasi yang sesuai antara kebutuhan pengguna, jenis data, dan lokasi diharapkan pengguna dapat mendapatkan *insight* atau memenuhi kebutuhan pengguna.

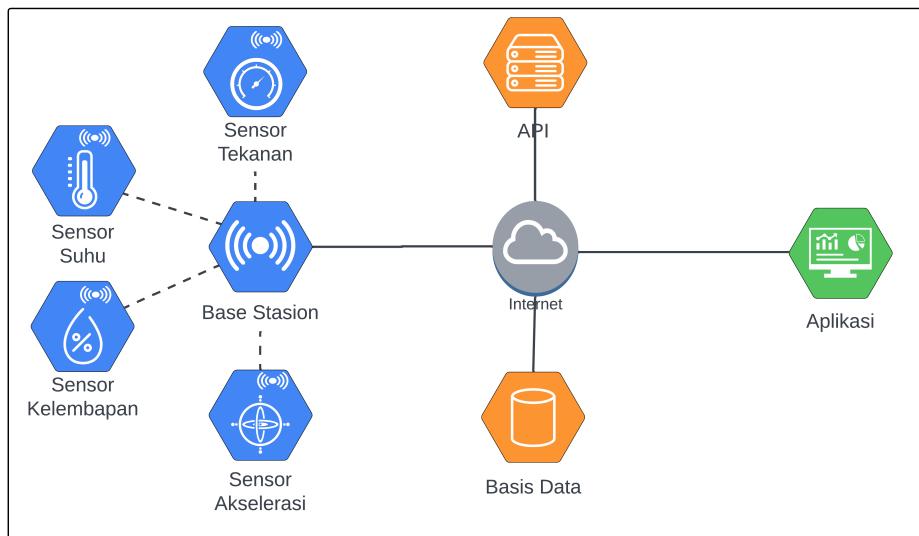
Kondisi suatu lingkungan satu lokasi yang tidak luas, seperti area kantor, rumah, dan universitas memiliki kecenderungan tidak berebeda. Dengan anggapan tersebut perangkat lunak yang dibuat setiap lokasi WSN hanya akan menggunakan hanya satu *node sensor* dengan empat jenis sensor. Namun tidak menutup kemungkinan dalam satu lokasi terdapat dua atau lebih WSN untuk menangani hal seperti kebutuhan untuk mengamati kondisi lingkungan di dalam dan di luar sebuah gedung yang sama.

Perangkat Lunak akan dibagi menjadi setidaknya tiga bagian besar yaitu bagian WSN, bagian *server* ,dan tampilan visualisasi data. Berikut penjelasan singkat perangkat lunak yang dibutuhkan:

- Bagian WSN
 - Bagian *node sensor* merupakan bagian yang melakukan *sensing*, kemudian mengirimkan data *sensing* ke *base station* menggunakan radio dengan protokol Zigbee. Bagian ini diunggah ke *node sensor*.
 - Bagian *base station* merupakan bagian yang bertugas untuk menerima atau mengirimkan pesan dari atau ke radio dengan protokol Zigbee, kemudian diteruskan ke komputer yang terhubung. Bagian ini diunggah ke *base station*.
 - Bagian *base station* pada komputer merupakan bagian yang bertugas untuk mengaktifkan program perangkat lunak kedua, berkomunikasi atau menerima pesan dari *node sensor*, memberikan informasi waktu, dan mengirimkan data *sensing* ke bagian *server*.
 - Bagian *node sensor* merupakan bagian yang bertugas untuk mengaktifkan *node sensor*.
- Bagian *server* merupakan bagian yang bertugas untuk menerima dan menyimpan data *sensing*, memberikan data yang dibutuhkan untuk visualisasi, mengatur pendaftaran dan konfigurasi *base station*, pendaftaran lokasi baru, dan lain sebagainya.
- Bagian tampilan visualisasi data merupakan bagian yang bertugas untuk meminta data ke bagian *server* dan menampilkan data ke dalam suatu bentuk visualisasi data yang telah dipilih oleh pengguna.

3.1.1 Analisis Desain Sistem

Sesuai dengan yang telah dijelaskan pada 3.1 maka dibuatlah desain sistem. Desain sistem yang dibangun memiliki tiga buah bagian utama yaitu WSN, *server*, dan aplikasi klien. Penjelasan setiap bagian akan dibahas pada subbab 3.1.4 hingga 3.1.6. Gambar desain sistem dapat dilihat pada gambar 3.1.



Gambar 3.1: Skema Desain Sistem

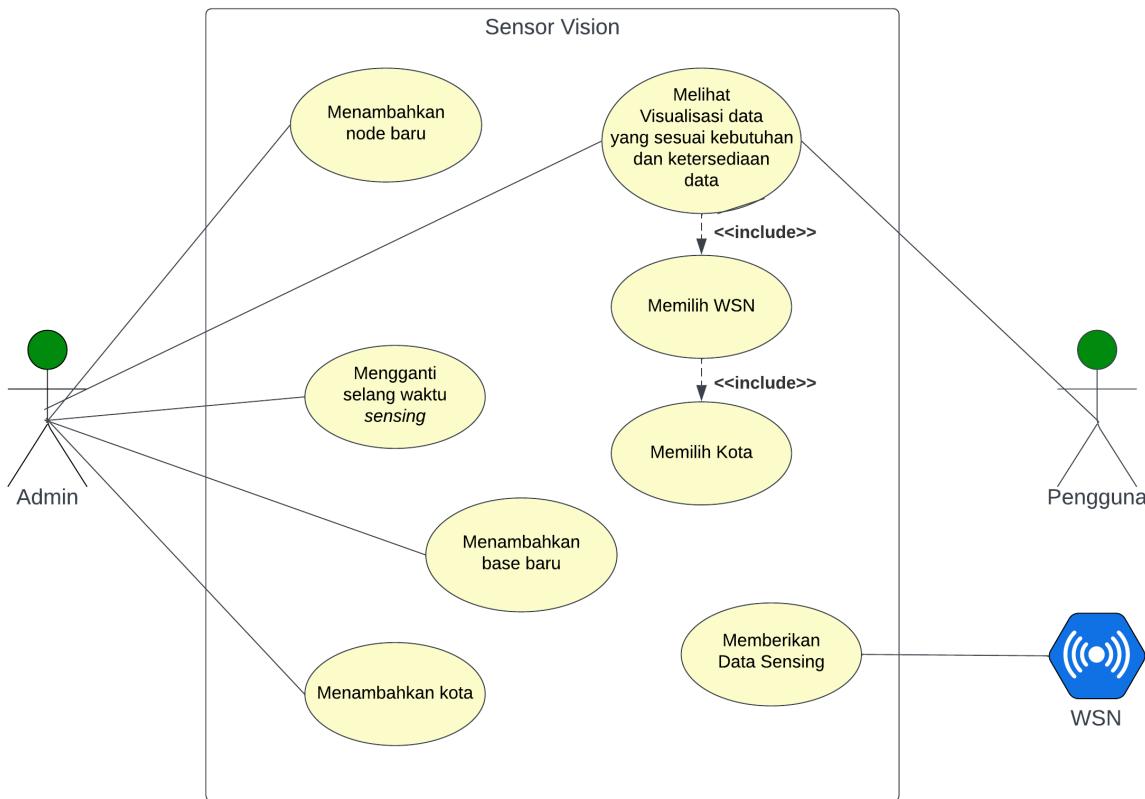
3.1.2 Analisis Fungsi Perangkat Lunak

Berdasarkan deskripsi perangkat lunak pada bagian 3.1 maka perangkat lunak setidaknya memiliki fungsi-fungsi sebagai berikut:

1. Memilih sebuah *node sensor* yang terdaftar untuk divisualisasikan.
2. Memilih jenis data yang hendak di visualisasikan.
3. Memilih jenis visualisasi data yang sesuai dengan kebutuhan pengguna.
4. Menambahkan WSN pada suatu lokasi.
5. Menambahkan *node sensor* pada suatu *base station*.
6. Melihat status pada *node sensor*.
7. Mengatur selang waktu *sensing* sebuah *node sensor*.
8. Menambahkan kota baru.
9. *login*, *logout*, dan daftar.

3.1.3 Use Case Diagram dan Use Case Skenario Website

Berdasarkan analisis fungsi yang dijabarkan pada bagian 3.1.2, maka *use case* diagram perangkat lunak website untuk admin dan pengguna dapat dilihat pada gambar 3.2 . Kemudian untuk skenario-skenario yang terdapat pada *use case* dapat dilihat pada tabel 3.6 sampai 3.5 .



Gambar 3.2: Diagram Use Case Perangkat Lunak

Tabel 3.1: Tabel Use Case Skenario Melihat Visualisasi Data

Nama	Melihat Visualisasi data yang sesuai kebutuhan dan ketersediaan data
Deskripsi	Pengguna memilih dan melihat visualisasi data yang sesuai dengan ketersediaan data dan kebutuhan
Aktor	Pengguna
Prekondisi	Pengguna sudah melakukan <i>login</i>
Prakondisi	Pengguna ditampilkan Visualisasi data yang sesuai dengan ketersediaan data dan kebutuhan pengguna
Alur Skenario	<ol style="list-style-type: none"> 1. Pengguna memilih kota. 2. Pengguna memilih lokasi sensor. 3. Pengguna memilih jenis data yang tersedia di tempat yang sudah dipilih sebelumnya. 4. Pengguna memilih jenis visualisasi data yang sesuai dengan kebutuhan pengguna. 5. Pengguna menekan tombol untuk memvisualisasikan data.

Tabel 3.2: Tabel Use Case Skenario Menambahkan Kota

Nama	Menambahkan Kota
Deskripsi	Admin menambahkan kota baru
Aktor	Admin
Prekondisi	Admin sudah melakukan dan sedang <i>login</i>
Prakondisi	Admin berhasil menambahkan kota baru
Alur Skenario	<ol style="list-style-type: none"> 1. Admin memilih tambah kota baru pada bagian menu. 2. Sistem menampilkan halaman formulir kota baru yang akan ditambahkan. 3. Admin mengisi halaman formulir dan menekan lanjutkan. 4. Sistem menampilkan kembali halaman utama admin

Tabel 3.3: Tabel Use Case Skenario Menambahkan *Base Station* Baru

Nama	Menambahkan <i>base station</i> baru
Deskripsi	Admin menambahkan <i>base station</i> baru pada suatu lokasi
Aktor	Admin
Prekondisi	Admin sudah melakukan dan sedang <i>login</i>
Prakondisi	Admin berhasil menambahkan <i>base station</i> baru pada suatu lokasi
Alur Skenario	<ol style="list-style-type: none"> 1. Admin menekan tombol Tambah <i>Base Station</i>. 2. Sistem menampilkan halaman formulir <i>base station</i> baru yang akan ditambahkan. 3. Admin mengisi formulir yaitu nama <i>base station</i>, posisi <i>latitude</i>, dan posisi <i>longitude</i>. Kemudian admin menekan kirim. 4. Sistem menampilkan kembali halaman utama admin

Tabel 3.4: Tabel Use Case Skenario Mengganti Selang Waktu *Sensing*

Nama	Mengganti selang waktu <i>sensing</i>
Deskripsi	Admin mengganti selang waktu <i>sensing</i>
Aktor	Admin
Prekondisi	Admin sudah melakukan dan sedang <i>login</i>
Prakondisi	Selang waktu <i>sensing</i> WSN yang dipilih berubah
Alur Skenario	<ol style="list-style-type: none"> 1. Admin menekan tombol <i>update Interval</i>. 2. Admin memilih <i>node</i> yang ingin dilakukan <i>update</i>. 3. Admin mengisi selang waktu baru dalam detik dan menekan tombol kirim. 4. Sistem menampilkan kembali halaman utama admin.

Tabel 3.5: Tabel Use Case Skenario Menambahkan *Node* Sensor Baru

Nama	Menambahkan <i>node</i> sensor baru pada suatu lokasi yang terdapat <i>base station</i>
Deskripsi	Admin menambahkan <i>node</i> sensor baru pada suatu lokasi yang terdapat <i>base station</i>
Aktor	Admin
Prekondisi	Admin sudah melakukan dan sedang <i>login</i>
Prakondisi	Admin berhasil menambahkan <i>node</i> sensor baru pada suatu lokasi yang terdapat <i>base station</i>
Alur Skenario	<ol style="list-style-type: none"> 1. Admin menekan tombol Tambah <i>Node Sensor</i>. 2. Sistem menampilkan halaman formulir <i>node</i> sensor yaitu memilih kota, memilih <i>base station</i>, mengisi nama lokasi, mencentang <i>indoor</i> bila posisi <i>node</i> di dalam ruangan, mengisi identifier, interval, dan mencentang sensor-sensor yang tersedia. 3. Admin menekan kirim. 4. Sistem menampilkan kembali halaman utama admin.

Tabel 3.6: Tabel Use Case Skenario Memberikan Data *Sensing*

Nama	Memberikan Data <i>Sensing</i>
Deskripsi	Sensor Melakukan <i>Sensing</i> setiap interval
Aktor	<i>Node</i> sensor sudah aktif
Prekondisi	<i>Node</i> sensor sudah aktif
Prakondisi	Data hasil <i>sensing</i> sampai ke API
Alur Skenario	<ol style="list-style-type: none"> 1. Setiap interval tertentu <i>node</i> sensor melakukan <i>sensing</i>. 2. Mengirimkan data <i>sensing</i> ke <i>base station</i>.

3.1.4 Wireless Sensor Network (WSN)

Bagian WSN memiliki fungsi memberikan data setiap selang waktu tertentu. Bagian WSN terbagi menjadi dua yaitu bagian bagian yaitu bagian *base station* dan bagian *node* sensor. Pada bagian *base station* akan dijelaskan lebih detil di bagian 3.1.4.1, sedangkan untuk bagian *node* sensor akan dijelaskan pada bagian 3.1.4.2.

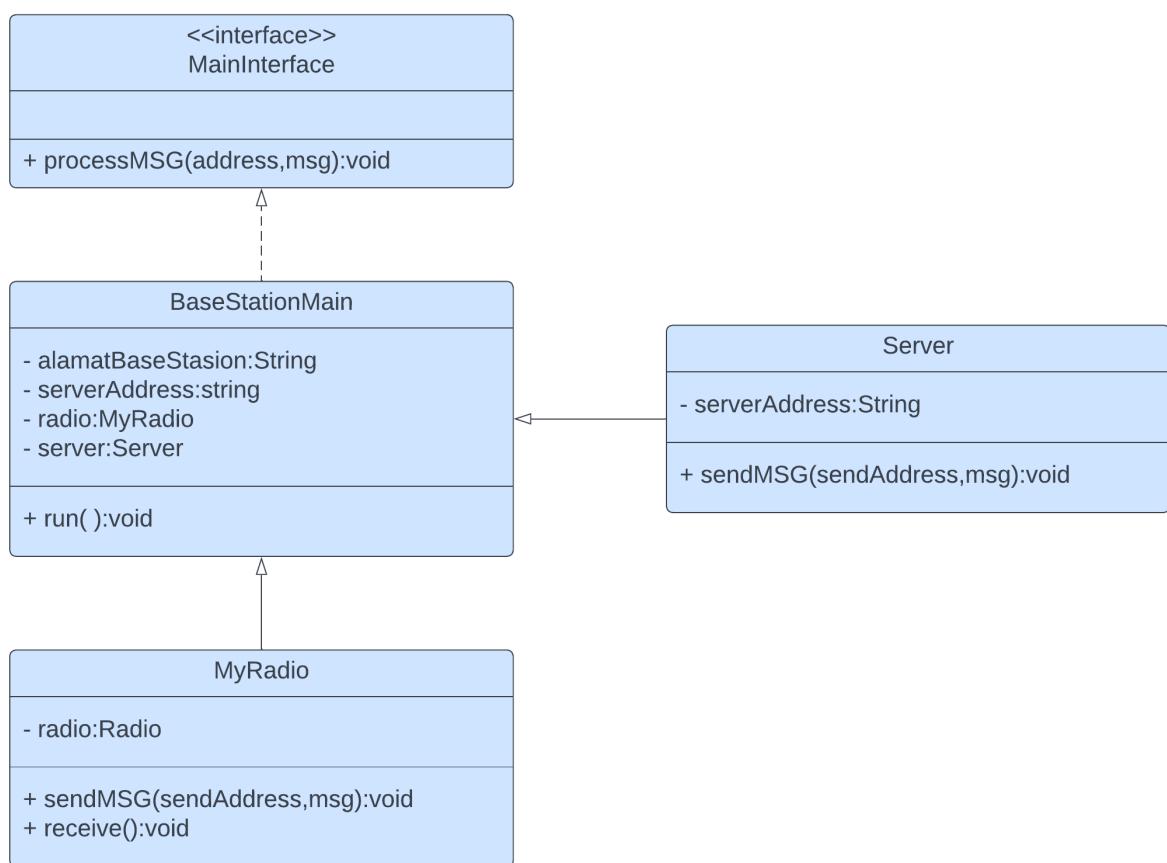
3.1.4.1 Base Station di Preon32

Bagian *base station* merupakan bagian yang bertugas untuk berkomunikasi langsung dengan server, mengatur *node* sensor seperti selang waktu sensing dan inisialisasi awal. Berikut diagram kelas perangkat lunak yang terdapat pada *base station* dapat dilihat pada gambar 3.3.

Pada kelas diagram perangkat lunak pada *base station* terdapat tiga buah kelas yaitu Server, Main, Node. Berikut adalah penjelasan kelas-kelas tersebut.

- BaseStaTionMain

Kelas BaseStaTionMain merupakan sebuah kelas yang akan dijalankan pertama kali pada saat *base station* diberi daya karena di dalamnya terdapat fungsi main yang merupakan fungsi pertama yang akan dijalankan pada pemrograman Java. Di dalam Kelas Main terdapat atribut dan fungsi lain yaitu:

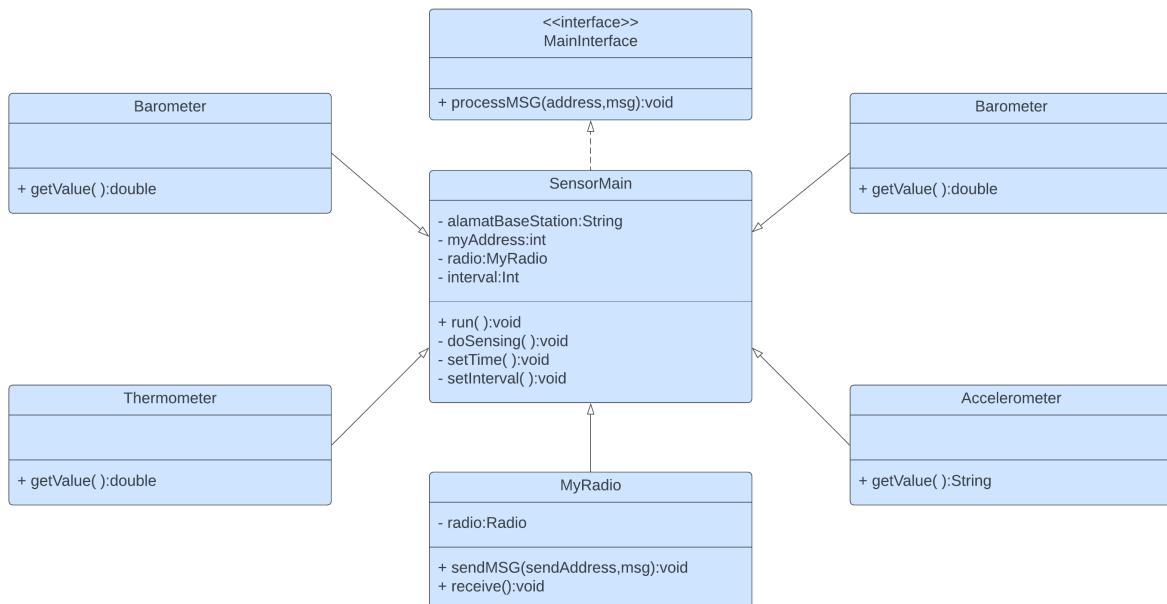


Gambar 3.3: Diagram Kelas Perangkat Lunak pada *Base Station*

- Variabel alamatBaseStasion
Variabel untuk menyimpan alamat radio *base station*
- Variabel server
Variabel untuk menampung objek Server yang akan dikirimkan data hasil *sensing* dari *node* sensor.
- Fungsi run
Fungsi yang bertugas untuk melakukan inisialisasi awal.
- Fungsi ProcessMSG
Fungsi yang bertugas untuk menentukan sebuah pesan hendak dikirim ke *server* atau ke radio untuk diteruskan ke *node*.
- Server merupakan kelas yang digunakan untuk mengatur komunikasi dengan server. Di dalam Kelas Server terdapat atribut dan fungsi lain yaitu:
 - Variabel serverAddress
Variabel yang digunakan untuk menyimpan protokol, alamat dan port dari *server*..
 - Fungsi send bertugas secara langsung mengirimkan data ke server.
- MyRadio merupakan kelas yang digunakan untuk berhubungan dengan *node*, seperti pengiriman data *sensing*, mengubah selang waktu *sensing*, insialisasi dan sinkronisasi waktu. Di dalam Kelas Node terdapat atribut dan fungsi lain yaitu:
 - Variabel radio
Variabel merupakan sebuah objek radio yang merupakan *library* penggunaan radio.
 - Fungsi sendMSG
Sebuah fungsi yang digunakan untuk mengirimkan suatu pesan ke *node* sensor tujuan.
 - Fungsi reveice Sebuah fungsi yang digunakan untuk menerima pesan dari *node* sensor.

3.1.4.2 Node Sensor

Bagian *node* sensor merupakan bagian yang melakukan *sensing* kondisi lingkungan dan memberikan hasil *sensing* kepada *base station* dalam suatu format data yang dibahas pada 3.3.1. Berikut diagram kelas perangkat lunak yang terdapat pada *node* sensor dapat dilihat pada gambar 3.4.



Gambar 3.4: Diagram Kelas Sederhana Perangkat Lunak pada *Node Sensor*

Pada kelas diagram perangkat lunak pada *node* sensor terdapat tujuh buah kelas yaitu MyBaseStation, Main, Sensing, Suhu, Kelembapan, Tekanan, dan Getaran. Berikut adalah penjelasan kelas-kelas tersebut.

- SensorMain

Kelas SensorMain merupakan sebuah kelas yang akan dijalankan pertama kali pada saat *node* sensor diberi daya karena di dalamnya terdapat fungsi main yang merupakan fungsi pertama yang akan dijalankan pada pemrograman Java. Di dalam Kelas Main terdapat atribut dan fungsi lain yaitu:

- Variabel alamatBaseStation
Variabel yang menyimpan alamat radio *base station*.
 - Variabel myAddress
Variabel yang menyimpan alamat radio *node*.
 - Variabel radio
Variabel objek sebuah radio yang digunakan untuk berkomunikasi dengan *base station*.
 - Variabel interval
Variabel yang menyimpan selang waktu untuk *sensing*.
 - Fungsi ProcessMSG
Fungsi yang bertugas untuk memisahkan memecah dan menjalankan pesan perintah yang dikirimkan *base station*
 - Fungsi run
Fungsi yang bertugas untuk melakukan inisialisasi awal dan menjalankan fungsi *sensing*.
 - Fungsi doSensing
Fungsi yang bertugas untuk memerintahkan sensor mengambil kondisi lingkungan sesuai dengan tipe sensor.
 - Fungsi setTime
Fungsi yang bertugas untuk menganti waktu pada *node sensor*
 - Fungsi setInterval
Fungsi yang bertugas untuk menganti interval. [3.3.1](#).
- kelas MyRadio merupakan kelas yang digunakan untuk berhubungan dengan *base station*, seperti pengiriman data *sensing*, mengubah selang waktu *sensing*, dan lain sebagainya. Di dalam Kelas Node terdapat atribut dan fungsi lain yaitu:
 - Variabel radio
Variabel merupakan sebuah objek radio yang merupakan *library* penggunaan radio.
 - Fungsi sendMSG
Sebuah fungsi yang digunakan untuk mengirimkan suatu pesan ke *base station* tujuan.
 - Fungsi reveive Sebuah fungsi yang digunakan untuk menerima pesan dari *base station*.
 - kelas Barometer, Thermometer, Higrometer, dan Akselerometer merupakan kelas objek sensor. Pada kelas ini terdapat satu buah fungsi yaitu getValue yang digunakan untuk melakukan *sensing* dan mengambil nilainya.

3.1.5 API

Bagian API merupakan bagian yang menghubungkan antara bagian website, WSN, dan basis data. Pada skripsi ini penulis memilih menggunakan REST API dikarenakan tidak perlunya dua buah jalur komunikasi. Dua jalur komunikasi yang dimaksud yaitu satu jalur untuk data *realtime* yang terhubung ke WSN dan satu jalur lagi untuk menyimpan data. Untuk menghubungkan antar bagian-bagian yang ada maka dibutuhkan sebuah fungsi sebagai berikut:

- insertKota

Untuk menyimpan informasi kota ke basis data.

- **getKota**
Untuk mengambil detail informasi sebuah kota dari basis data.
- **getBS**
Untuk mengambil detail informasi mengenai sebuah *base station*.
- **insertBS**
Untuk menyimpan informasi sebuah *base station* ke basis data.
- **getNodeSensor**
Untuk mengambil detail informasi mengenai sebuah *node sensor*.
- **insertNodeSensor**
Untuk menyimpan informasi sebuah *node sensor* ke basis data.
- **insertTipeSensor**
Untuk menyimpan informasi tipe sensor apa saja yang terdapat pada sebuah *node sensor* ke basis data.
- **update**
Untuk menyimpan sebuah perintah ke basis data.
- **sensing**
Untuk menyimpan sebuah nilai pengukuran ke basis data.
- **getData**
Untuk mengambil nilai pengukuran pada selang waktu tertentu dari basis data.
- **getDataRealTime**
Untuk mengambil data realtime langsung dari sebuah variabel.
- **getInterval**
Untuk mengambil nilai interval sebuah *node sensor*.

3.1.6 Basis Data

Perangkat Lunak Visualisasi data sensor tentunya memerlukan basis data untuk menyimpan data hasil *sensing* WSN agar dapat memenuhi kebutuhan visualisasi data pengguna. Oleh karena itu dibuat rancangan skema basis data dalam bentuk diagram *entity relationship*. Diagram *entity relationship* dapat dilihat pada gambar 3.5.

Diagram *entity relationship* pada gambar 3.5 memiliki rancangan bentuk fisik sebagai berikut:

- Entitas Base Station

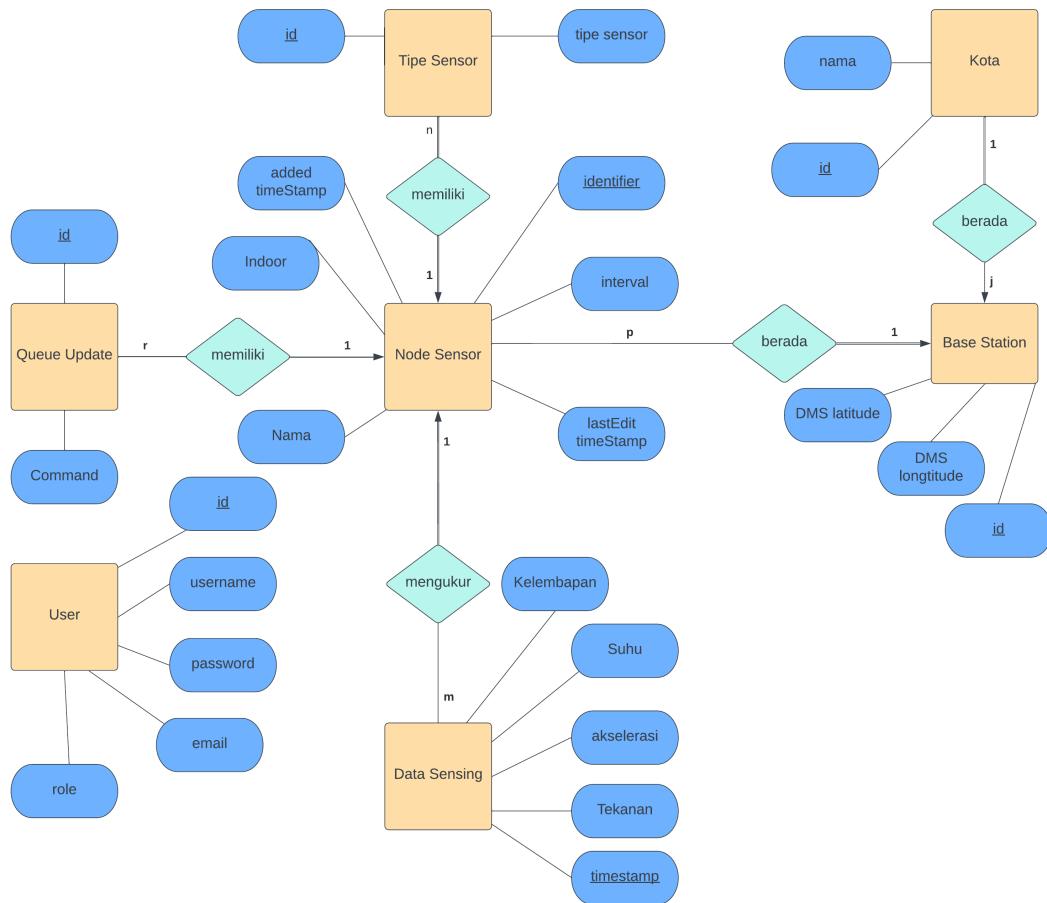
Tabel 3.7: Rancangan Fisik Entitas Base Station

Base Station			
Tipe	Nama Atribut	Tipe Data	Keterangan
PK	id	int	id dari Base Station
	DMS latitude	varchar(20)	digunakan untuk menyimpan posisi longitude WSN
	DMS latitude	varchar(20)	digunakan untuk menyimpan posisi latitude WSN

- Entitas Kota

Tabel 3.8: Rancangan Fisik Entitas Kota

Kota			
Tipe	Nama Atribut	Tipe Data	Keterangan
PK	id	int	id dari Kota
	nama	varchar(50)	digunakan untuk menyimpan nama kota



Gambar 3.5: Diagram Relasi entitas Basis Data

- Entitas Tipe Sensor

Tabel 3.9: Rancangan Fisik Tipe Sensor

Kota			
Tipe	Nama Atribut	Tipe Data	Keterangan
PK	id	int	id dari Kota
	tipeSensor	varchar(10)	digunakan untuk menyimpan tipe sensor

- Entitas Queue Update

Tabel 3.10: Rancangan Fisik Queue Update

Kota			
Tipe	Nama Atribut	Tipe Data	Keterangan
PK	id	int	id dari Kota
	command	varchar(50)	digunakan untuk perintah yang akan dikirimkan ke node

- Entitas Node Sensor

Tabel 3.11: Rancangan Fisik Entitas Node Sensor

Node Sensor			
Tipe	Nama Atribut	Tipe Data	Keterangan
PK	nama	varchar(50)	nama dari node sensor
	identifier	varchar(4)	identifier atau pengenal dari node sensor yang dikirimkan pada saat mengirim atau meminta data
	interval	Long	digunakan untuk mencatat selang waktu melakukan sensing
	indoor	tinyint	digunakan untuk menyimpan
	addedTimeStamp	datetime	digunakan untuk waktu node disaat node disimpan ke basis data
	lastEditTimeStamp	datetime	digunakan untuk waktu node di saat node ada perubahan seperti perubahan interval

- Entitas User

Tabel 3.12: Rancangan Fisik Entitas User

User			
Tipe	Nama Atribut	Tipe Data	Keterangan
PK	id	int	id dari sebuah user
	Username	varchart(50)	digunakan untuk menyimpan username user
	Password	varchart(100)	digunakan untuk menyimpan password user
	email	varchart(50)	digunakan untuk menyimpan email user
	role	tinyint	digunakan untuk role dari user. 0 untuk pengguna 1 untuk admin

Entitas Data *Sensing*

Entitas Data Sensing merupakan tabel yang akan dibuat oleh server setiap tanggal 1. Selain itu entitas ini juga akan dibedakan tabel antar *node sensor*. Kedua hal ini bertujuan untuk mengurangi dan membatasi beban dari mengurutkan data setiap kali WSN memberikan data ke bagian server. Sehingga format nama pada entitas ini akan menjadi **[IDENTIFIER Sensor]-[bulan]-[tahun]**.

Tabel 3.13: Rancangan Fisik Entitas Data Sensing

Data Sensing			
Tipe	Nama Atribut	Tipe Data	Keterangan
PK	timestamp	DATE	primary key Data Sensing
	kelembapan	varchart(50)	digunakan untuk menyimpan kelembapan
	suhu	varchart(100)	digunakan untuk menyimpan suhu
	tekanan	varchart(50)	digunakan untuk menyimpan tekanan
	akselerasi	bit	digunakan untuk menyimpan akselerasi terhadap sumbu x

Setiap entitas memiliki relasi dengan entitas lain, berikut merupakan hubungan antar entitas:

- Entitas Base Station mempunya relasi 'memiliki' dengan entitas Node Sensor. Relasi ini memiliki kardinalitas satu ke banyak yang berarti satu buah *base station* bisa memiliki banyak *node sensor*.
- Entitas Base Station mempunyai relasi 'mengukur' dengan entitas Data Sensing. Relasi ini memiliki kardinalitas satu ke banyak yang berarti satu buah *base station* mengukur lebih dari satu kali.
- Entitas Base Station mempunyai relasi 'berada' dengan entitas Lokasi. Relasi ini memiliki kardinalitas satu ke satu yang berarti satu buah *base station* hanya bisa berada pada satu wilayah saja.
- Entitas Lokasi mempunyai relasi 'berada' dengan entitas Relasi ini memiliki kardinalitas banyak ke satu yang berarti banyak lokasi dapat berada di satu kota yang sama.

3.2 Analisis Tujuan Pembuatan Visualisasi Data Sensor

Pada skripsi ini penulis menggunakan 4 buah jenis sensor yaitu sensor kelembapan, tekanan, suhu, dan akselerasi. Keempat buah sensor ini memiliki data yang bersifat kuantitatif yang dihasilkan dari pengukuran sensornya. Berdasarkan teori yang dijabarkan pada bagian 2.1.2 maka penulis mendapatkan kandidat visualisasi yang sesuai yaitu visualisasi yang dapat menampilkan data bersifat kuantitatif. Kemudian penulis juga melakukan pengamatan kepada aplikasi-aplikasi sejenis yang sudah diluncurkan maupun bersifat eksperimen, yang dijabarkan pada bagian 2.1.4. Berdasarkan teori dan pengamatan penulis mendapatkan jenis-jenis visualisasi data yang sesuai seperti diagram titik, diagram garis, diagram luas, dan lain sebagainya.

Setelah didapatkan jenis-jenis visualisasi yang umum digunakan penulis melakukan analisa untuk melakukan penentuan jenis visualisasi yang sesuai. Penentuan ini dilakukan agar pembuat visualisasi dapat menentukan jenis visualisasi 2.1.2 yang sesuai agar informasi dapat disampaikan dengan baik dan tidak terjadi bias. Berikut akan dibahas tujuan-tujuan utama dari pembuatan visualisasi data.

3.2.1 Tujuan Pemantauan (*Monitoring*)

Pembuatan visualisasi data dapat bertujuan untuk pemantauan secara langsung *realtime* terhadap kondisi lingkungan di tempat sensor berada. Poin utama dari tujuan ini adalah mengamati kondisi saat ini dari suatu lingkungan. Namun terkadang ada juga kebutuhan mengetahui data yang sesaat sebelumnya, tujuannya seperti untuk mencari peningkatan ataupun penurunan yang signifikan agar dapat menentukan tindakan berikutnya..

Visualisasi data yang cocok untuk mengamati suatu lingkungan adalah visualisasi data yang sederhana dalam artian pembaca tidak perlu untuk melihat apa yang berada di sumbu x maupun y agar dapat dengan mudah dan cepat dipahami. Sehingga visualisasi data yang sesuai untuk yang *realtime* adalah teks sederhana dan *gauge chart*. Namun penggunaan *gauge chart* tidak bisa untuk semua jenis data, karena perlu ditentukan terlebih batas bawah dan batas atasnya. Jika tidak menentukan batas atas atau bawah dapat membuat masalah seperti data baru yang berada di luar bawah maupun atas. Penyelesaian teknis cukup mudah yaitu batas atas dan bawah disesuaikan dengan batas atas dan bawah sensor. Penyesuaian ini kurang baik dipakai karena ada kecenderungan sensor memiliki rentang nilai yang besar, sehingga peningkatan atau penurunan tidak secara jelas terlihat, sehingga ada kecenderungan *gauge chart* selalu menampilkan rendah atau tinggi. Penyelesaian kedua yaitu *update* saja batas atas atau bawahnya jika ada data yang di luar batas. Namun hal ini menyebabkan masalah yaitu terkadang pembaca hanya melihat posisi penunjuknya (jarum) seperti penunjuknya ditengah-tengah. Masalah yang muncul yaitu jika batas bawah atau atasnya diganti atau tergantikan maka penunjuknya tidak lagi berada ditengah-tengah melainkan lebih mendekati batas atas atau lebih mendekati batas bawah. Hal ini menyebabkan

terjadinya kesalahan interpretasi dalam pembacaan, oleh karena itu pada skripsi ini *gauge chart* hanya akan dipakai pada satu jenis data yang sudah jelas batas atas maupun bawahnya yaitu kelembapan relatif. Alasan kelembapan relatif dapat menggunakan *gauge chart* adalah kelembapan relatif memiliki rentang yang jelas yaitu 0% (tidak mengandung uap air) sampai dengan 100% (tidak dapat menyerap uap air).

Kemudian kebutuhan berikutnya yaitu mengetahui data yang sesaat sebelumnya, visualisasi data yang cocok untuk kebutuhan ini adalah visualisasi yang memperlihatkan data terbaru berkaitan dengan data sebelumnya. Banyak diagram atau plot yang dapat menampilkan waktu dengan nilai. Namun dari diagram yang dapat waktu dengan nilai hanya beberapa yang dapat menunjukkan adanya keterkaitan antara nilai dengan nilai sebelumnya sehingga sebagian dari diagram tersebut tidak memenuhi syarat. Salah satu diagram yang tidak memenuhi syarat yaitu diagram titik. Pada diagram titik representasi antar data tidak memunculkan kesan data tersebut terhubung antara data dengan data berikutnya atau sebelumnya. Kemudian contoh berikutnya yang tidak memenuhi yaitu diagram batang dan histogram. Pada diagram batang terdapat pemisah yang menyebabkan adanya kesan antar data terpisah atau tidak berkaitan dengan data sebelumnya. Diagram batang bisa tidak diberikan jarak, diagram ini disebut dengan histogram, pada histogram juga terdapat masalah yaitu adanya kesan data seolah-olah berubah hanya pada saat pengukuran (*sensing*) atau tidak secara bertahap. Sehingga visualisasi data yang cocok untuk kebutuhan ini adalah diagram garis dan diagram luas. Kedua diagram ini menampilkan data dengan waktunya, kemudian setiap data dihubungkan dengan garis gradien dengan data berikutnya. Garis gradien ini yang membuat kedua diagram ini layak karena garis menimbulkan kesan data tersebut terhubung dan berkaitan, kemudian gradien yang menandakan data tersebut berubah secara perlahan.

3.2.2 Tujuan Melihat Data Lampau

Visualisasi data tidak hanya menampilkan data yang *realtime* saja tetapi ada kebutuhan melihat data lampau. Tujuan dari melihat data lampau dapat beragam seperti melihat tren yang cenderung naik, turun atau datar. Kemudian tujuan melihat data lampau dapat juga untuk melakukan perbandingan wilayah yang memiliki nilai data lebih tinggi atau rendah. Ataupun tujuan bisa untuk mencari kejanggalan seperti pada siang hari suhu dingin atau pada malam hari suhu sangat panas tanpa diikuti oleh data sebelum dan sesudahnya. Tujuan melihat data lampau sangat beragam, namun ada satu hal yang tetap menjadi poin utama yaitu kondisi suatu lingkungan umumnya tidak berubah secara signifikan dan tanpa diikuti oleh sebelum dan sesudahnya, seharusnya masih ada keterkaitan dengan data sebelum dan sesudahnya oleh karena itu tipe diagram yang cocok untuk kebutuhan ini adalah diagram garis dan diagram luas. Alasan kedua diagram tersebut cocok, dapat dibaca pada sub-bab 3.2.1.

3.2.3 Tujuan Melihat Statistik

Tidak semua kebutuhan dapat dicapai dengan melihat data lampau semata saja, terkadang data lampau butuh diproses lebih lanjut seperti di *plot* antar jenis data, dicari nilai terendah dan tertingginya, ataupun dihitung nilai-nilai statistik seperti kuartil, median, dan lain sebagainya. Umumnya untuk menampilkan statistik, diagram yang digunakan untuk sudah spesifik seperti jika ingin mencari hubungan antar kategori sudah spesifik untuk menggunakan *scatter plot*. Untuk mencari statistik seperti kuartil, pencilan, dan lain sebagainya dapat digunakan *box plot*.

Untuk kebutuhan mencari korelasi atau hubungan antar jenis data dapat digunakan scatter plot karena scatter plot memberikan sebuah penanda berupa titik ditarik dan kolom yang bersesuaian, dari titik titik yang terbentuk dapat dilihat polanya. Untuk menghitung statistik dan mencari pencilan dapat menggunakan box plot.

Tetapi tetap ada beberapa jenis statistik yang dapat divisualisasikan dengan cara berbeda seperti untuk menampilkan jumlah dalam suatu rentang, misalkan jumlah data yang memiliki suhu direntang 1-10, 11-20, dan seterusnya dapat digunakan diagram lingkaran, diagram batang baik

yang bertumpuk maupun tidak, dan histogram. Untuk jumlah disuatu rentang tentunya tidak menggunakan diagram garis ataupun area karena antar datanya tidak memiliki hubungan juga tidak seharusnya bergradien.

Pemilihan antara ketiga jenis visualisasi dapat didasari oleh berapa banyak yang ingin divisualisasikan, jika lebih dari satu sensor maka sebaiknya menggunakan diagram batang. Sedangkan jika hanya satu sensor maka bisa menggunakan diagram lingkaran atau histogram. Tetapi penulis tetap menyarankan menggunakan histogram karena dengan jika menggunakan diagram lingkaran akan sukar dibaca dan dimengerti, jika visualisasi data yang dicari memiliki jumlah kategori yang banyak. Dengan kategorinya yang banyak maka diagram lingkaran akan menampilkan banyak kategori dengan sudut yang hampir-hampir mirip, hal ini menyebabkan menjadi sukar dibaca.

Pemilihan diagram batang untuk yang memiliki setidaknya dua buah sensor didasari karena, jika menggunakan histogram seluruh batang akan saling bersebelahan tanpa jarak hal ini membuat grafik menjadi belang-belang yang menyebabkan menjadi tidak nyaman untuk dilihat dan dibaca. Untuk visualisasi setidaknya dua buah sensor sebaiknya menggunakan diagram batang karena ada jarak kosong yang memisahkan antar kategori sehingga lebih nyaman dilihat dan dibaca. Namun perlu dipertimbangkan, jika kategorinya memiliki jumlah yang banyak. Karena jika kategorinya memiliki jumlah yang banyak maka lebar batang akan menjadi kecil dan akan semakin kecil karena diberi jarak antar kategorinya.

3.3 Format Pertukaran Data

Format pertukaran data menjadi hal yang penting dalam komunikasi agar tidak terjadi kesalahpahaman dalam komunikasi, oleh karena itu format pertukaran data perlu dipilih atau dirancang. Perancangan sebuah perangkat lunak perlu untuk mempertimbangkan format pertukaran data yang akan digunakan agar perangkat lunak menjadi efisien dalam ukuran data dan performa komputasi. Terdapat tiga jenis format data yang umum digunakan yaitu XML, JSON, dan YAML.

3.3.1 Analisa Format Data *Node Sensor* ke *Base Station*

Format data untuk pengiriman dari *node sensor* ke *base station* ataupun sebaliknya dirancang agar penggunaan datanya minimal mungkin. Salah satu cara membuat mengurangi penggunaan data adalah dengan merepresentasikan data yang sering dikirim seperti Tipe sensor dengan sebuah karakter atau angka. Oleh karena itu format data dibuat key-pair value, dimana key sebagai sebuah karakter atau karakter-karakter yang merepresentasikan tipe sensor dan *value* merupakan hasil pengukuran. Pada tabel 3.14 sebuah tipe sensor yang semula menggunakan banyak karakter akan menjadi menggunakan karakter yang lebih sedikit yaitu satu sampai 2. Pemilihan karakter didasari oleh simbol pada satuan internasional, namun terdapat beberapa jenis data yang tidak dapat direpresentasikan oleh utf-8 seperti α dan ada juga jenis sensor yang tidak terdapat simbol pada Satuan internasional yaitu kelembapan. Oleh sebab itu untuk α dikarenakan pada satuan internasional akselerasi diberi simbol α dengan huruf kecil maka diganti ke huruf latin 'a' sedangkan untuk kelembapan karena tidak ada simbolnya maka diberi singkatan bahasa Inggris yaitu rh dari *relative humidity*.

Tabel 3.14: Tabel format Data Tipe Sensor

Representasi	Jenis Sensor
T	Suhu
Pa	Tekanan
rh	Kelembapan relatif
a	Akselerasi

3.3.2 Analisa Format Data Base Station ke Server

Pada *base station* akan ditambahkan waktu lokal karena data hasil *sensing* WSN berkaitan erat terhadap suatu waktu. Selain itu juga data WSN memiliki kaitan terhadap lokasi *node sensor* melakukan *sensing*. Oleh karena itu akan ditambahkan waktu lokal dan id dari *base station*.

3.3.2.1 Analisa Format Data dan Pemilihan Format Data

```
<root>
<time>yyyy-MM-dd-HH-mm-ss</time>
<baseld>AAAA</baseld>
<result>
<rh>80</rh>
<T>32</T>
<Pa>60</Pa>
<a>0.04</a>
<a>0.01</a>
<a>0.97</a>
</result>
</root>
```

```
{
  "time": "yyyy-MM-dd-HH-mm-ss",
  "baseld": "AAAA",
  "result": {
    "rh": 80,
    "T": 32,
    "pa": 60,
    "a": [0.04, 0.01, 0.97]
  }
}
```

```
---
time: yyyy-MM-dd-HH-mm-ss
baseld: 'AAAA'
result:
  'rh': 80
  'T': 32
  'Pa': 60
  'a':
    - 0.04
    - 0.01
    - 0.97
```

Gambar 3.6: Data yang di format dengan XML

Gambar 3.7: Data yang di format dengan JSON

Gambar 3.8: Data yang di format dengan YAML

Terdapat beberapa format data yang umum digunakan yaitu XML, JSON, dan YAML. Data yang sudah diformat XML dapat dilihat pada gambar 3.6, format JSON dapat dilihat pada gambar 3.7, dan dapat format YAML dilihat pada gambar 3.8. Ada beberapa pertimbangan yang dapat dilakukan saat menentukan format pertukaran data yang akan digunakan pada perangkat lunak sebagai berikut:

- Ukuran data

Ukuran data menjadi poin yang cukup penting agar data yang dikirim lebih ringkas agar data dapat dikirimkan dengan lebih cepat. Format XML memerlukan 183 karakter, sedangkan JSON memerlukan 101 karakter dan YAML membutuhkan 8 karakter. Dari ketiga format tersebut diketahui JSON menggunakan lebih sedikit karakter. Hal tersebut dikarenakan pada format XML *key* dituliskan dua kali yaitu pada saat pembuka dan penutup, sedangkan untuk YAML memerlukan *white space* berupa spasi atau tab.

- Performa

Berdasarkan penelitian berjudul "Comparison between JSON and YAML for data serialization" yang dilakukan oleh Malin Eriksson dan Victor Hallberg dari KTH Royal Institute of Technology didapatkan hasil JSON lebih cepat dibandingkan dengan YAML baik untuk *serialization* maupun *deserialization*.[20].

Berdasarkan kedua poin yang dijabarkan pada 3.3.2.1 didapatkan hasil yaitu perangkat lunak akan lebih baik jika menggunakan JSON karena ukuran data yang paling kecil, serta kemampuan komputasi yang lebih baik dibandingkan dengan YAML.

3.3.3 Analisa Format Dari Base station ke Node Sensor

Pada saat *node sensor* dinyalakan dibutuhkan proses sinkronisasi waktu dengan *base station* atau pada saat ingin mengganti selang waktu *sensing* dibutuhkan komunikasi dari *base station* ke *node sensor* untuk memberi perintah berupa pesan. Berikut merupakan isi data atau pesan yang dikirimkan oleh *base station*:

Perintah	Keterangan
setTime:<time>	Untuk mengatur waktu WSN. <time>merupakan nilai waktu berupa jumlah milidetik dari epoch time (1 Januari 1970 00:00:00)
setInterval:<interval>	untuk mengatur selang waktu sensing <interval>merupakan nilai waktu dalam waktu milidetik

3.3.4 Analisa Format Dari Node Sensor ke Base station

Pada saat sebuah *node* sensor dinyalakan sebuah node akan meminta informasi waktu dan interval. Selain itu *node* sensor juga akan mengirimkan data *sensing* setiap selang waktu. Sebuah *node* sensor mengirimkan data selalu diawali sebuah prefiks 'server' ataupun 'bs' sebagai penanda kepada siapa *node* sensor berkomunikasi, hal ini bertujuan agar kerja *base station* karena jika prefiks 'server' maka *base station* akan meneruskan pesan ke komputer. sedangkan jika prefiks 'bs' maka akan diproses di *base station*. Berikut merupakan isi atau pesan yang dikirimkan oleh *node* sensor

- Pesan data *sensing*

Data hasil *sensing* akan dikirimkan sebagai berikut,

```
server:"time":<waktu milidetik>,
"idBS": "<identifier>","T":<nilai>,"rh":<nilai>,
"Pa":<nilai>,"a": [<nilai>,<nilai>,<nilai>]
```

Pesan hasil *sensing* mengirimkan waktu dalam bentuk milidetik, karena jika dibandingkan dengan format waktu yang umumnya dipakai yaitu 'yyyy-MM-dd hh:mm:ss' maka akan menggunakan 19 karakter sedangkan untuk milidetik dari *epoch time* misalkan '253384993031' (2023-06-14 15:42:02 maka hanya akan menggunakan 12 karakter, dengan catatan format umum menghilangkan informasi milidetik. Sedangkan untuk nilai hasil *sensing* dikirimkan dengan desimal presisi dua.

- Pesan pada saat *node* baru diaktifkan.

```
"bs:intervalRequest,"node":<identifier>
```

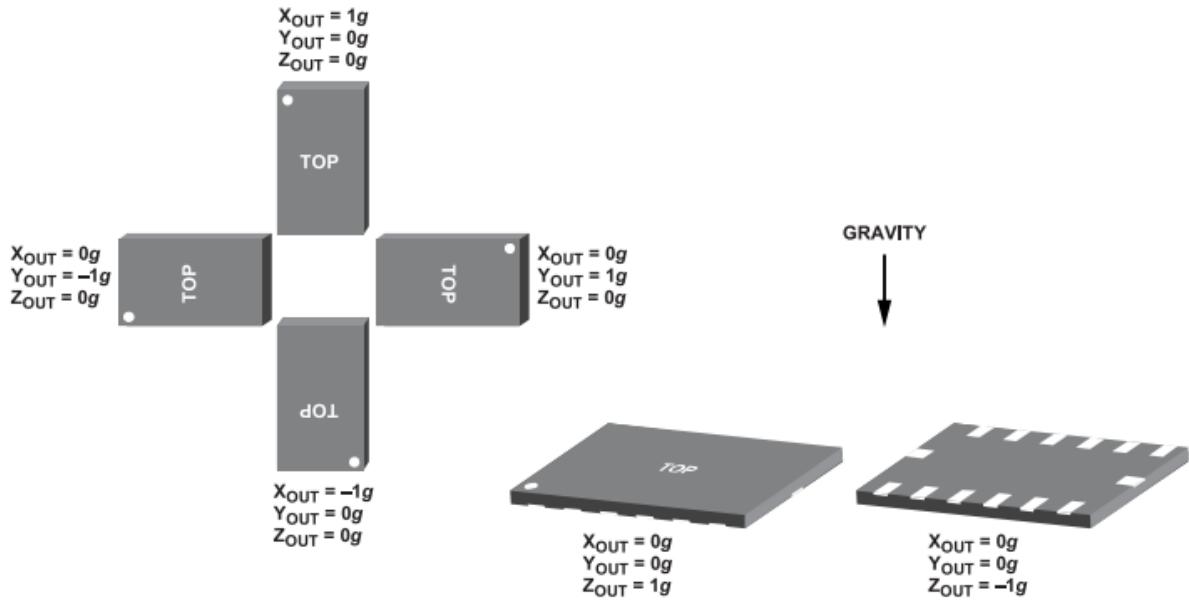
Pesan pada saat *node* baru diaktifkan memiliki prefiks 'bs' karena pada awalnya *node* perlu untuk mengambil waktu dan mengambil interval *sensing*.

3.4 Analisis Masalah Pada WSN

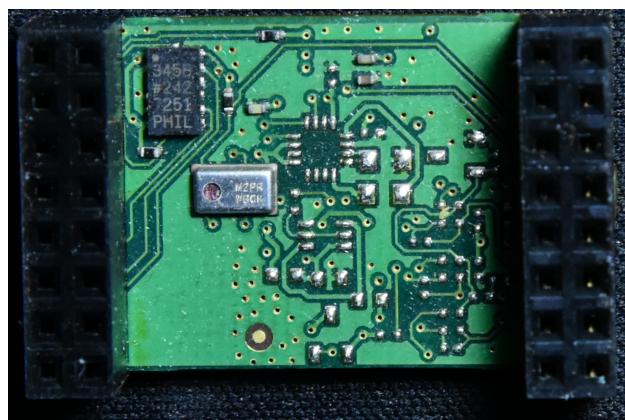
Pada pembuatan visualisasi data khususnya data sensor, pembuat juga perlu menangani masalah yang terjadi di bagian WSN. Masalah pada bagian WSN, dianalisis dan ditangani pada bagian ini.

3.4.1 Penempatan Node Sensor

Pada Preon32 terdapat sensor akselerometer. Akselerometer ini mengukur percepatan atau akselerasi, akselerasi merupakan besaran vektor atau besaran yang memiliki arah, karena itu *node* sensor perlu ditetapkan suatu aturan peletakan sensor agar nilai nya memiliki suatu arah yang tetap. Peletakan satu *node* sensor agar seragam dengan *node* sensor dapat dilakukan dengan seluruh *node* sensor menghadap ke suatu objek seperti kutub utara magnetis bumi, bangunan ikonis, dan lain sebagainya. Salah satu arah yang sering dijadikan acuan adalah menggunakan arah mata angin yang menjadikan kutub utara magnetis bumi sebagai acuan arah utara dan kutub selatan magnetis bumi sebagai arah selatan.

Gambar 3.9: Diagram Akselerasi ADXL 345 ¹

Pada koordinat kartesian tiga dimensi terdapat tiga buah sumbu yaitu Sumbu X, Y, dan Z. Sumbu Z dikenal sebagai sumbu vertikal. Sumbu Z biasanya tegak lurus terhadap sumbu X dan sumbu Y. Sumbu Z- biasanya menandakan arahnya ke bawah sedangkan Z+ menandakan arahnya ke atas. Gravitasi merupakan percepatan menuju inti bumi, secara relatif manusia percepatan ke bawah, oleh karena itu penulis menetapkan gravitas bumi mengambil Z-. Kemudian pada kartesian jika dua dimensi maka sumbu Y merupakan sumbu vertikal, karena itu penulis menetapkan Y+ sebagai arah utara. Berdasarkan gambar 3.9, jika ditetapkan penulis Y+ sebagai percepatan ke arah utara maka X+ positif akan menjadi ke arah kanan atau timur. hal ini sesuai dengan koordinat kartesian yang digambarkan pada umumnya.

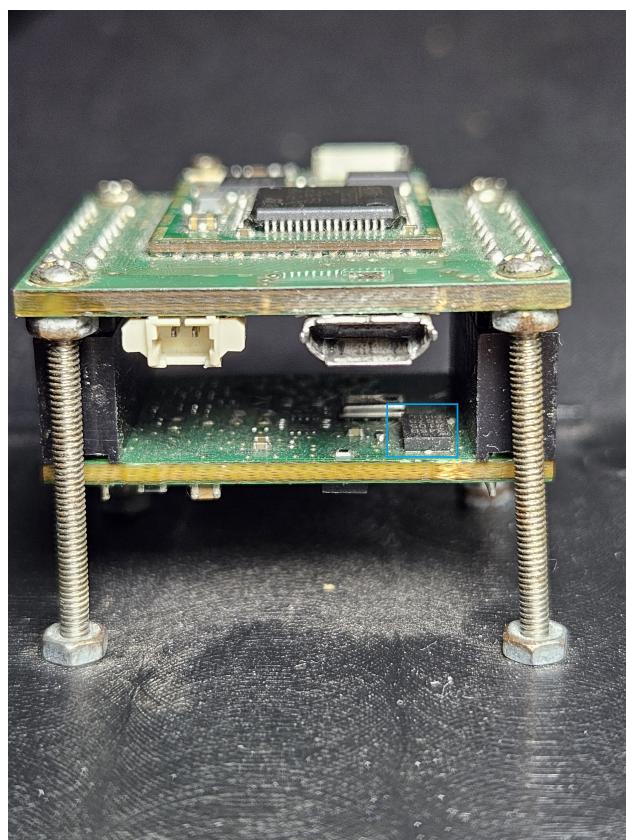
Gambar 3.10: Preon32 VariSen ²

Berdasarkan paragraf sebelumnya ditetapkan untuk X+ mengarah timur atau kanan. Dari gambar 3.9 dapat dilihat X+ menghadap ke atas. Pada gambar 3.10 sensor akselerometer yang di beri kotak biru berada pada salah satu tepian panjang dan x+ menghadap ke arah luar tepian tersebut. Kemudian pada gambar 3.11 dapat dilihat akselerometer yang penulis beri kotak biru berada

¹<https://www.analog.com/media/en/technical-documentation/data-sheets/adx1345.pdf> Diakses Juni 2032

²dokumentasi pribadi

³dokumentasi pribadi



Gambar 3.11: Preon32 ³

ditepiant antarmuka *micro-usb*. Oleh sebagai penanda, antarmuka *micro usb* harus menghadap kekanan atau ketimur.

3.4.2 Preon32 Tidak Terdapat Baterai khusus untuk Mengaktifkan Modul RTC

Pada perangkat Preon32 terdapat modul RTC, namun pada perangkat tidak terdapat baterai yang khusus hanya untuk memberikan energi untuk modul RTC. Sehingga modul RTC selalu kembali ke *epoch time* yaitu 1 Januari 1970. Hal ini menjadi masalah karena setiap data *sensing* seharusnya memiliki sebuah data waktu kapan data tersebut diambil. Untuk mengatasi hal ini dapat dilakukan sinkronisasi waktu. Sinkronisasi waktu pada WSN dibagi menjadi dua ada yang dilakukan oleh program WSN *base station* pada Komputer ada juga yang dilakukan oleh program WSN *node* sensor pada Preon32. Untuk yang program *base station* pada komputer tidak terlalu bermasalah karena *base station* selalu dijalankan oleh sebuah komputer, sehingga cukup mengambil informasi waktu dari komputer. Sedangkan yang dilakukan oleh *node* dibutuhkan sinkronisasi waktu menggunakan radio, karena tidak selamanya *node* sensor harus dijalankan dengan komputer, tetapi terkadang bisa dapat dayanya berasal dari baterai. Untuk sinkronisasi waktu pada *node* sensor akan dijelaskan urutan pada bagian 4.3.5.

BAB 4

PERANCANGAN

Pada bab ini menjelaskan mengenai perancangan perangkat lunak yang terdiri dari perancangan antarmuka, perancangan API, dan perancangan bagian WSN. Pada bagian perancangan antar muka akan ditampilkan *mock-up*. Kemudian pada perancangan API akan membahas mengenai dokumentasi API yang berisikan masukan dan keluaran. Terakhir bagian wsn akan membahas mengenai diagram kelas.

4.1 Perancangan Antarmuka

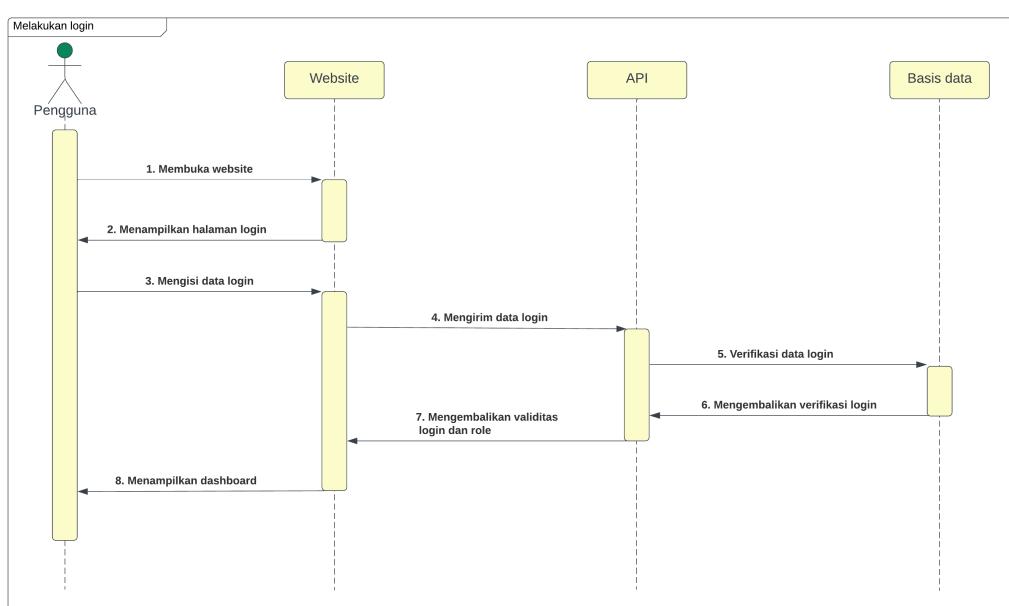
Perangkat lunak memiliki beberapa tujuan yang dijabarkan pada 3.1.2. Untuk memenuhi kebutuhan tersebut maka dibuatlah *sequence diagram* yang menunjukkan urutan dan aksi yang dilakukan. Kemudian pada bagian ini juga akan dirancang sebuah antarmuka tampilan website yang dapat memenuhi kebutuhan atau tujuan.

4.1.1 Diagram Sequence

Berikut merupakan diagram *sequence* yang mengambarkan urutan atau alur interaksi yang dilakukan oleh pengguna.

- Diagram *Sequence* Melakukan Login

Pada diagram *sequence* menunjukkan alur baik untuk pengguna maupun admin menggunakan diagram ini karena aktivitas yang dilakukan sama.

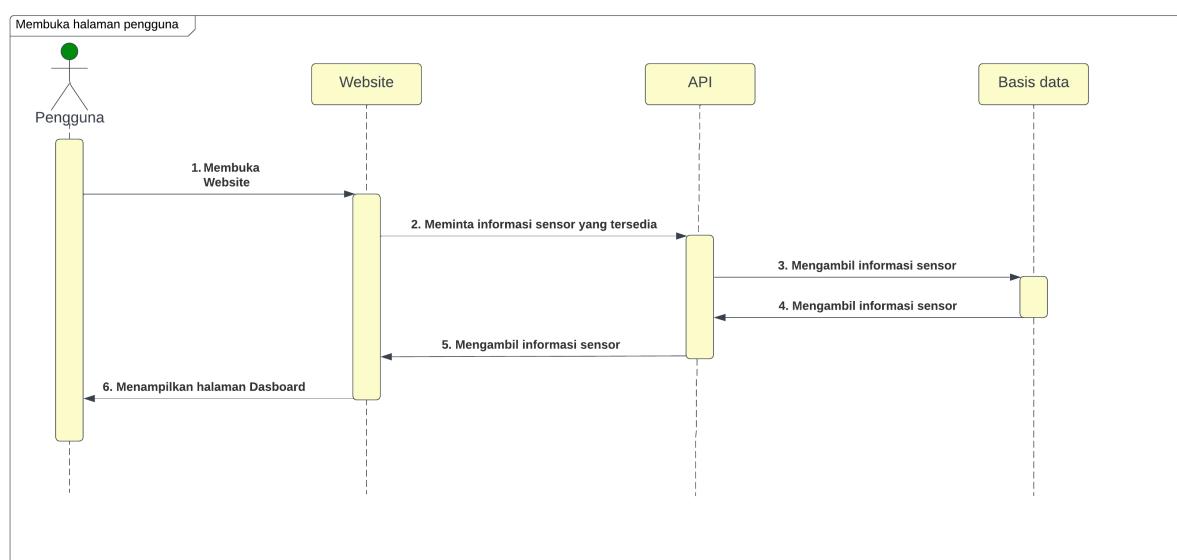


Gambar 4.1: Diagram *Sequence* Melakukan Login

Pada saat pertama kali pengguna akan membuka website dengan memasukan alamat website. Website akan memberikan halaman login. Kemudian pengguna dapat mengisikan informasi akun pengguna. Setelah mendapatkan data login website akan mengirimkan ke API agar oleh API dicek informasi akun dan diberikan informasi *role* jika informasi yang dimasukan pengguna benar. Kemudian website akan menampilkan halaman utama sesuai dengan dengan *role* pengguna.

- Diagram *Sequence* Membuka *Dashboard*

Pada diagram *sequence* menunjukkan alur komunikasi pengguna membuka halaman *dashboard*. Halaman *dashboard* disini merujuk pada halaman visualisasi data *realtime*, *history*, tabel, maupun statistik, karena ketiganya memiliki kesamaan dalam informasi yang dibutuhkan pada tampilan website. Halaman ini juga hanya dapat dibuka jika sudah melakukan *login* pada suatu sesi tertentu.



Gambar 4.2: Diagram *Sequence* Membuka Dashboard

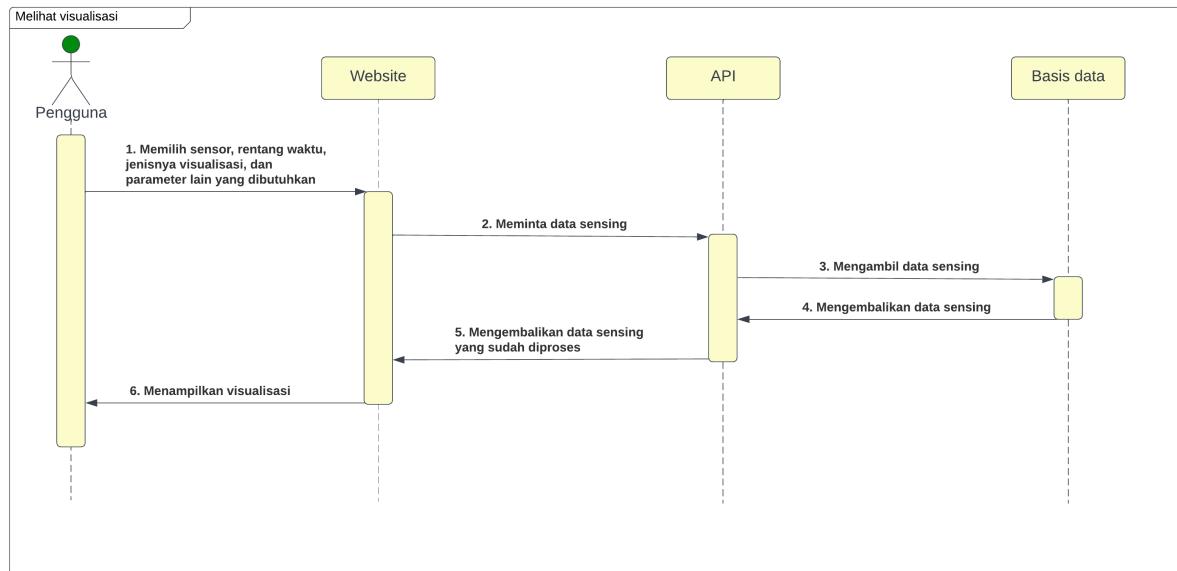
Pada saat pertama kali pengguna akan membuka halaman dasboard. Website akan mengambil informasi sensor ke API. API akan melakukan suatu pengambilan data ke basis data. Setelah API berhasil mendapatkan informasi sensor, API akan mengembalikan informasi sensor menuju website. Selanjutnya oleh website akan ditampilkan halaman *dashboard*.

- Diagram *Sequence* Melihat Visualisasi Data

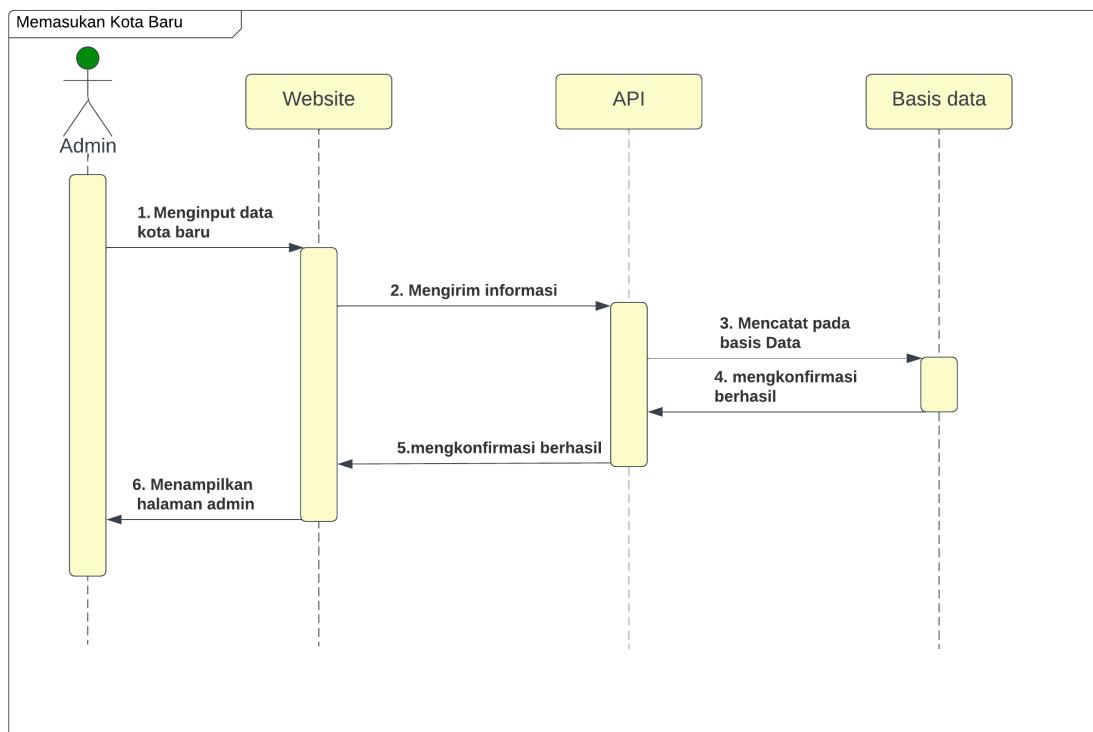
Pada diagram *sequence* menunjukkan alur cara pengguna untuk melihat visualisasi data. Diagram ini berlaku untuk keseluruhan tujuan visualisasi data dikarenakan perbedaan yang terjadi hanya dijumlah parameter yang harus diisi. Setelah pada berhasil membuka halaman *dashboard* pengguna ditampilkan halaman untuk visualisasikan data. Pengguna harus mengisi semua pilihan yang tersedia dan melakukan permintaan visualisasi data. Website akan membuat sebuah *request JSON* kepada API. API kemudian menentukan akan membutuhkan tabel pada basis data yang diperlukan. Setelah ditentukan tabel mana saja yang diperlukan API mengambil data pada basis data kemudian dikembalikan ke website agar oleh website ditampilkan visualisasi yang sesuai kebutuhan pengguna.

- Diagram *Sequence* Menambahkan *base station* atau *node sensor* atau kota baru

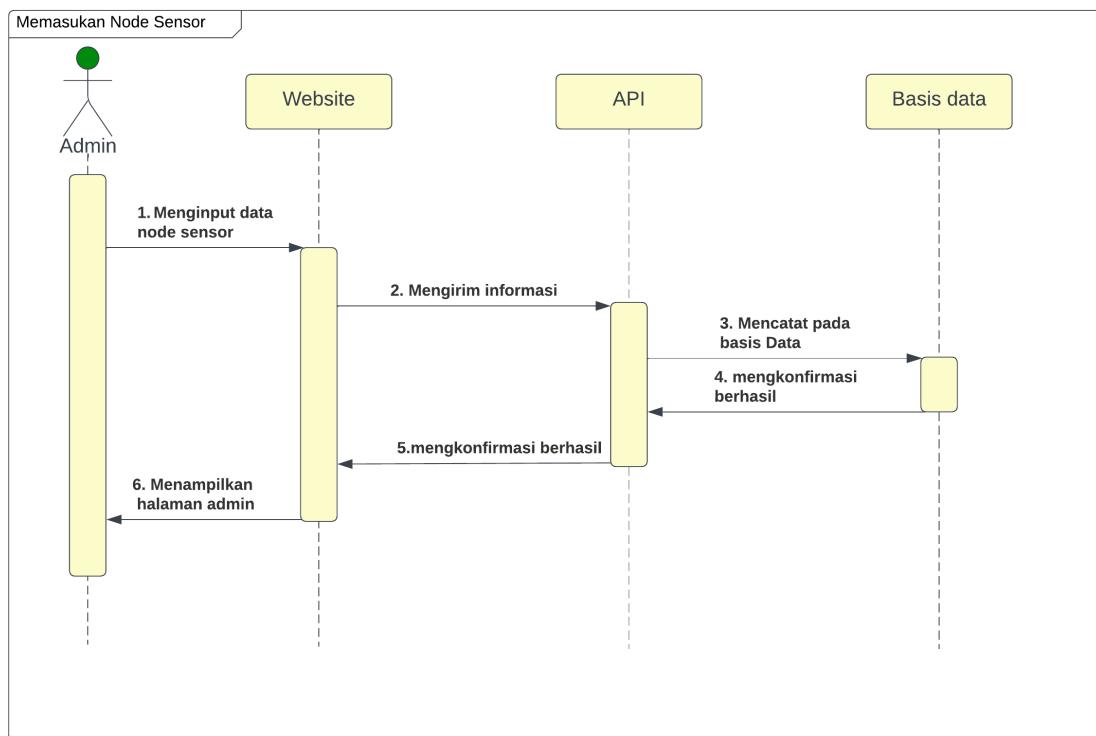
Pada diagram *sequence* menunjukkan alur pengguna untuk menambahkan *base station* atau *node sensor* atau kota baru. ketiganya memiliki kesamaan langkah-langkah setelah admin mengisi data input. Perbedaannya terletak pada data-data yang diminta oleh website akan berbeda antara *base station* atau *node sensor* atau kota baru. Berikut adalah diagram *sequence*:



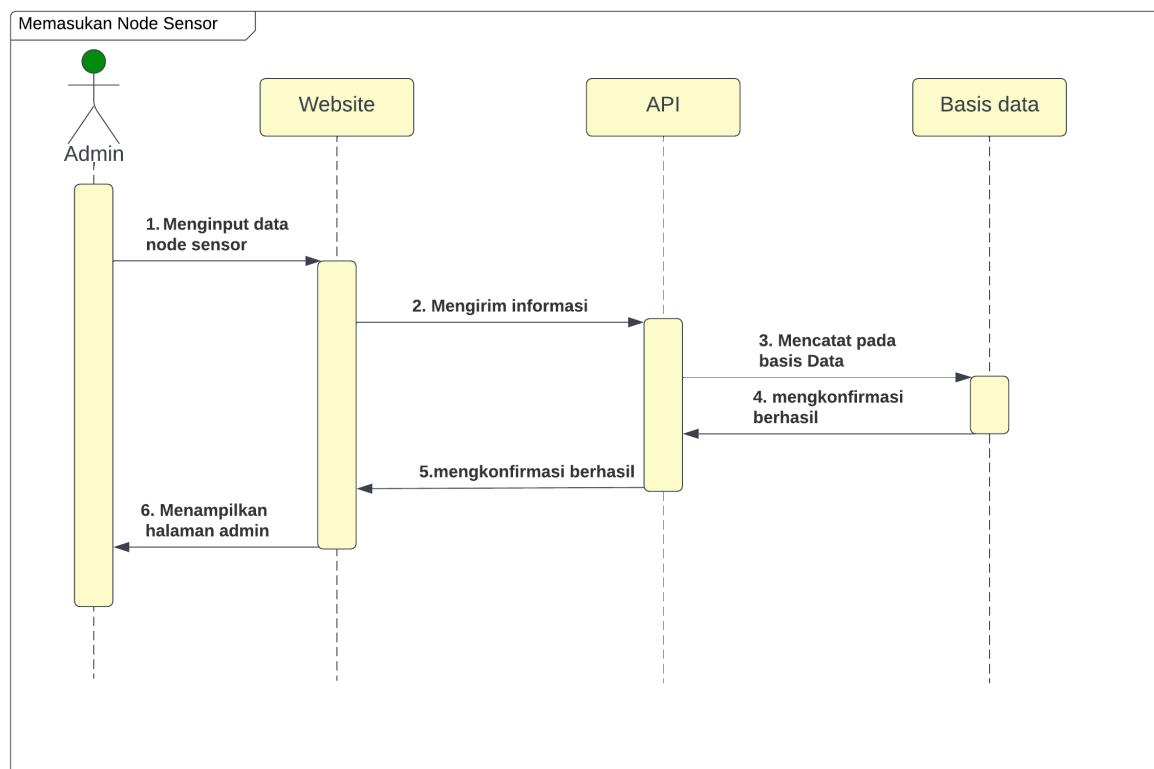
Gambar 4.3: Diagram Sequence Melihat Visualisasi Data



Gambar 4.4: Diagram Sequence Menambah Kota Baru



Gambar 4.5: Diagram Sequence Menambah base station



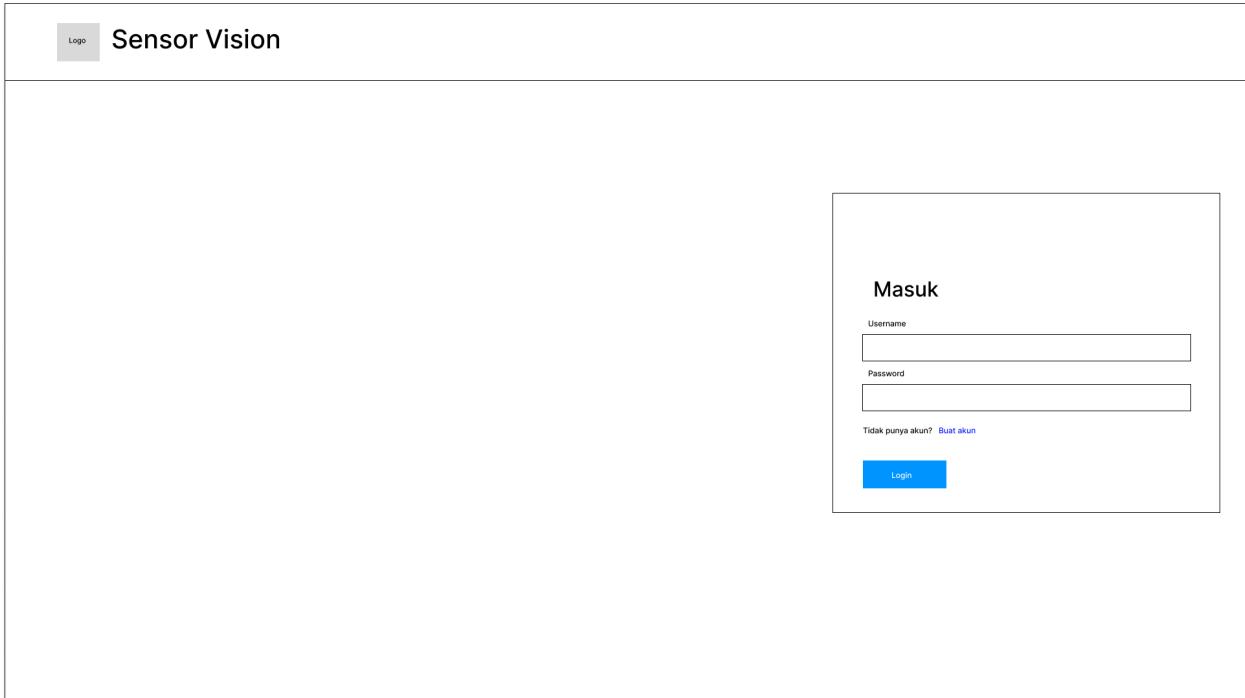
Gambar 4.6: Diagram Sequence Menambah base station

4.1.2 Mock-up

Berikut merupakan *mock-up* atau rancangan yang mengambarkan sekiranya tata-letak perangkat lunak yang dibangun pada skripsi ini.

- *Mock-up* Halaman Login

Halaman Login memiliki fungsi sebagai halaman pertama pada saat seseorang membuka website. Halaman ini juga digunakan sebagai pemisah antara pengguna dengan admin. *Mock-up* dapat dilihat pada gambar 4.7.



Gambar 4.7: *Mock-up* Halaman Login

- *Mock-up* Halaman Daftar

Halaman daftar memiliki fungsi sebagai halaman dimana seseorang mendaftarkan dirinya sebagai pengguna. *Mock-up* dapat dilihat pada gambar 4.8.

- *Mock-up* Halaman Visualisasi Data *History*

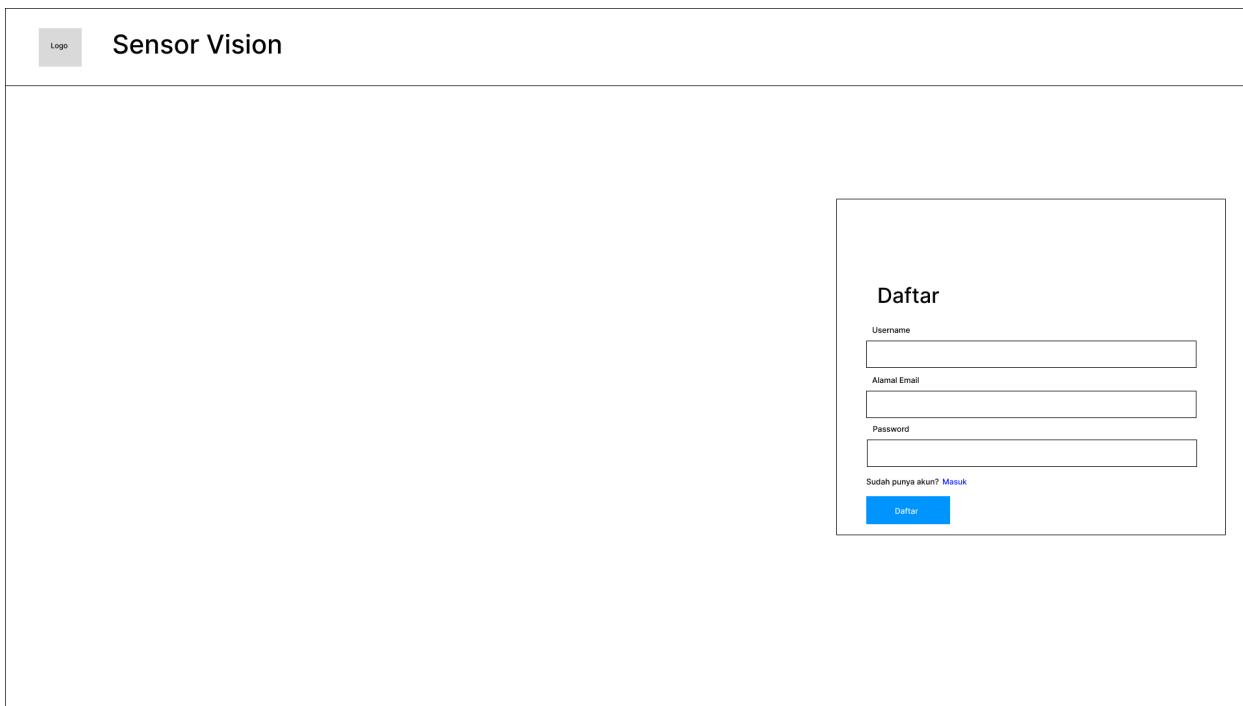
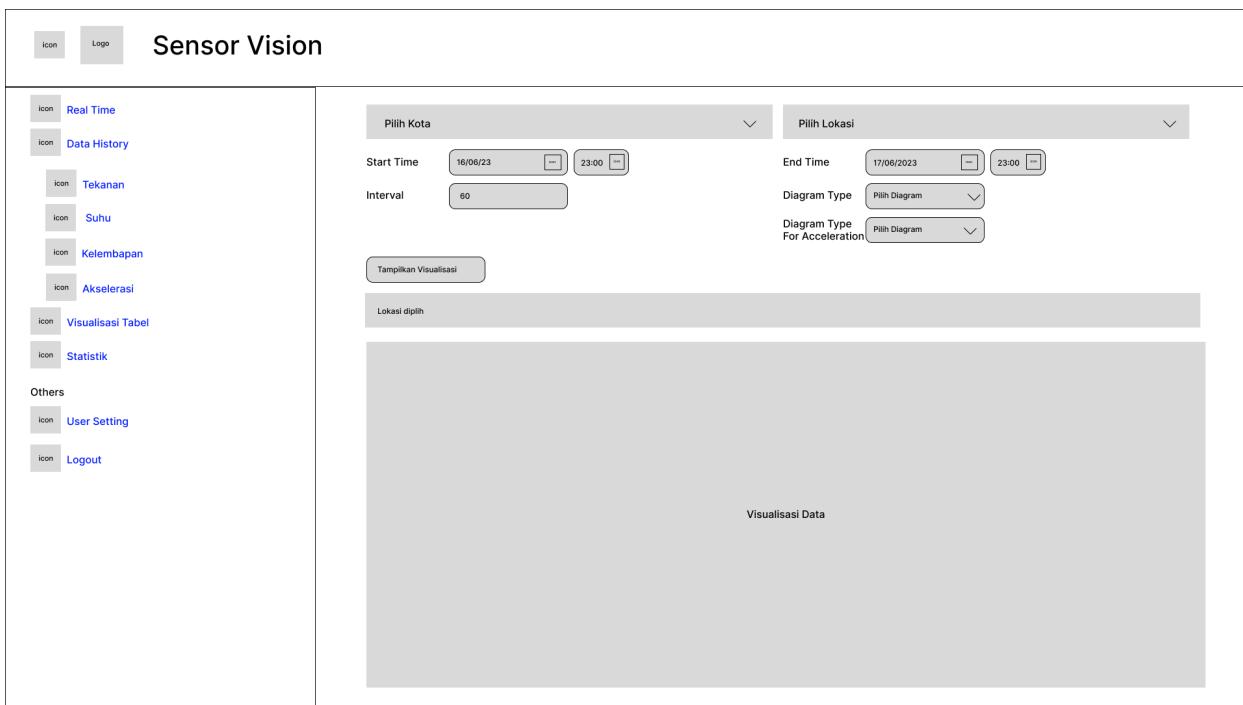
Halaman *history* memiliki fungsi yaitu untuk menampilkan visualisasi data berdasarkan data dimasa lampau. Dalam halaman ini terdapat beberapa yaitu memilih tipe yang sesuai dengan memilih opsi kemudian ada juga opsi untuk memilih interval data yang akan diambil. *Mock-up* dapat dilihat pada gambar 4.9.

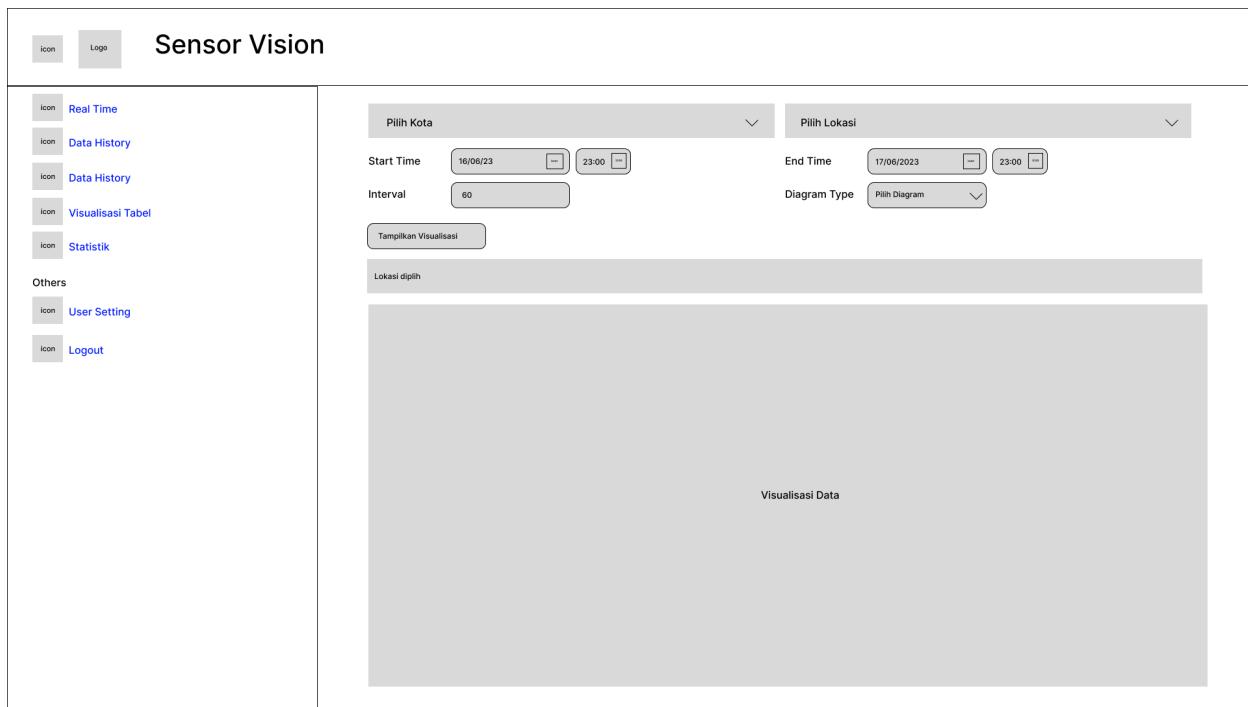
- *Mock-up* Halaman Visualisasi Data Statistik

Halaman *history* memiliki fungsi yaitu untuk menampilkan visualisasi data berdasarkan data dimasa lampau. Dalam halaman ini terdapat beberapa yaitu memilih tipe yang sesuai dengan memilih opsi kemudian ada juga opsi untuk memilih interval data yang akan diambil. *Mock-up* dapat dilihat pada gambar 4.10.

- *Mock-up* Halaman Visualisasi Data *RealTime*

Halaman *realtime* memiliki fungsi yaitu untuk menampilkan visualisasi data berdasarkan data terjadi pada saat pengguna mengakses. Data akan diambil sesuai dengan interval *node* sensor yang dipilih oleh pengguna. Dihalaman ini pengguna dapat memilih grafik berbeda antar jenis data, hal ini dikarenakan biasanya pada data *realtime* pengguna bukan untuk membandingkan antar jenis data tetapi pengguna umumnya lebih cenderung ke membandingkan antar lokasi. *Mock-up* dapat dilihat pada gambar 4.11.

Gambar 4.8: *Mock-up* Halaman DaftarGambar 4.9: *Mock-up* Halaman Visualisasi Data *History*

Gambar 4.10: *Mock-up* Halaman Visualisasi Data StatistikGambar 4.11: *Mock-up* Halaman Visualisasi Data *RealTime*

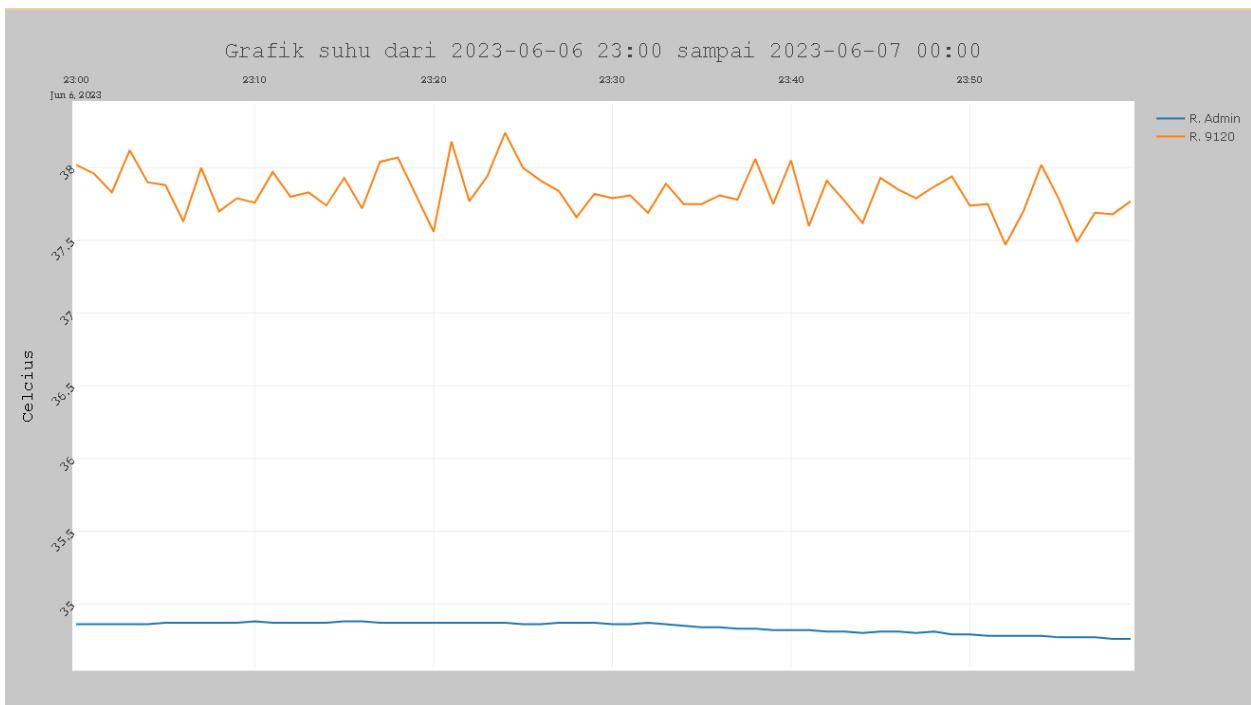
4.1.3 Tata Letak Visualisasi Data Antara Jenis Data

Tata letak visualisasi data antar jenis datanya akan dibuat bertumpuk atau bersebelahan seperti pada gambar 4.12. Hal ini dikarenakan setiap jenis data memiliki metrik yang berbeda-beda. Perbedaan metrik dapat menyebabkan dapat menyebabkan grafik menjadi sukar dibaca karena bisa saja memiliki selisih yang jauh sehingga perubahan sedikit nilai keduanya menjadi tidak terlihat

untuk beberapa tipe grafik seperti diagram luas. Kemudian untuk kebutuhan visualisasi data dengan banyak lokasi atau *node* sensor maka akan ditaruh pada visualisasi yang sama terkecuali untuk kebutuhan realtime, visualisasi data yang melibatkan banyak lokasi dapat dilihat pada gambar 4.13



Gambar 4.12: Contoh Halaman Visualisasi Data untuk Banyak Jenis Data



Gambar 4.13: Contoh Halaman Visualisasi Data untuk Banyak Node Sensor

4.2 Perancangan Application Programming Interface (API)

Perangkat lunak membutuhkan API untuk menghubungkan bagian WSN dengan bagian website. Pada bagian ini API bertugas untuk menyimpan data. API sendiri memiliki *endpointnya* berikut merupakan perancangannya:

Tabel 4.1: Endpoint API untuk Pendaftaran

Endpoint	/signup
Method	POST
Tujuan	Untuk mendaftarkan pengguna baru
Masukan	username, password dan email pengguna
Keluaran	Keterangan berhasil atau gagal

Tabel 4.2: Endpoint API untuk Masuk

Endpoint	/login
Method	POST
Tujuan	Untuk memverifikasi username dan password pengguna
Masukan	username, password
Keluaran	Keterangan berhasil beserta role atau gagal

Tabel 4.3: Endpoint API untuk Memasukan Kota Baru

Endpoint	/kota
Method	POST
Tujuan	Memasukan kota baru
Masukan	nama kota tersebut
Keluaran	Keterangan berhasil atau gagal

Tabel 4.4: Endpoint API untuk Mengambil Kota

Endpoint	/kota
Method	GET
Tujuan	Mengambil seluruh kota
Masukan	
Keluaran	Seluruh id kota beserta nama kota

Tabel 4.5: Endpoint API untuk Memasukan *Base Station* Baru

Endpoint	/bs
Method	POST
Tujuan	Memasukan <i>base station</i> baru
Masukan	nama <i>base station</i> , posisinya (latitude dan longitude)
Keluaran	Keterangan berhasil atau gagal

Tabel 4.6: Endpoint API untuk Mengambil *Base Station*

Endpoint	/bs
Method	GET
Tujuan	Mengambil <i>base station</i> baru
Masukan	
Keluaran	Informasi Seluruh <i>base station</i> yang terdaftar

Tabel 4.7: Endpoint API untuk Memasukan *Node Sensor* Baru

Endpoint	/node
Method	POST
Tujuan	Memasukan <i>node sensor</i> baru
Masukan	Informasi sensor seperti identifier, idBS yang terhubung, nama lokasi atau <i>node</i> , berada di luar ruangan atau di dalam ruangan, selang waktu <i>sensing</i> , sensor apa saja yang terdapat pada <i>node</i>
Keluaran	Informasi Seluruh <i>base station</i> yang terdaftar

Tabel 4.8: Endpoint API untuk Mengambil *Node Sensor* Baru

Endpoint	/node
Method	GET
Tujuan	Mengambil <i>node sensor</i>
Masukan	idBS yang terhubung dengan <i>node</i>
Keluaran	Informasi <i>node sensor</i> yang terhubung dengan <i>base station</i> yang dicari

Tabel 4.9: Endpoint API untuk Menyimpan Data *Sensing*

Endpoint	/sensing
Method	POST
Tujuan	Menyimpan data <i>sensing</i>
Masukan	informasi sensor seperti identifier, waktu <i>sensing</i> , dan data <i>sensing</i> itu sendiri
Keluaran	Informasi berhasil, gagal, atau terdapat perintah untuk <i>node</i> sensor

Tabel 4.10: Endpoint API untuk Melakukan Pembaharuan Interval *Sensing*

Endpoint	/update
Method	POST
Tujuan	Melakukan pembaharuan interval <i>sensing</i>
Masukan	Perintah berupa kata yang digabung dengan interval dalam milidetik
Keluaran	Informasi berhasil

Tabel 4.11: Endpoint API untuk Mengambil Selang Waktu *Node* Sensor

Endpoint	/interval
Method	GET
Tujuan	Mengambil selang waktu <i>node</i> sensor
Masukan	Informasi <i>node</i> sensor yang meminta
Keluaran	Perintah untuk mengganti interval

Tabel 4.12: Endpoint API untuk Mengambil Data untuk Divisualisasikan

Endpoint	/data
Method	POST
Tujuan	Mengambil data untuk divisualisasikan
Masukan	Informasi <i>node</i> sensor yang akan diambil, waktu mulai, waktu selesai, dan statistik
Keluaran	Data Hasil Sensingnya

Tabel 4.13: Endpoint API untuk Mengambil Data *Real Time* untuk Visualisasi

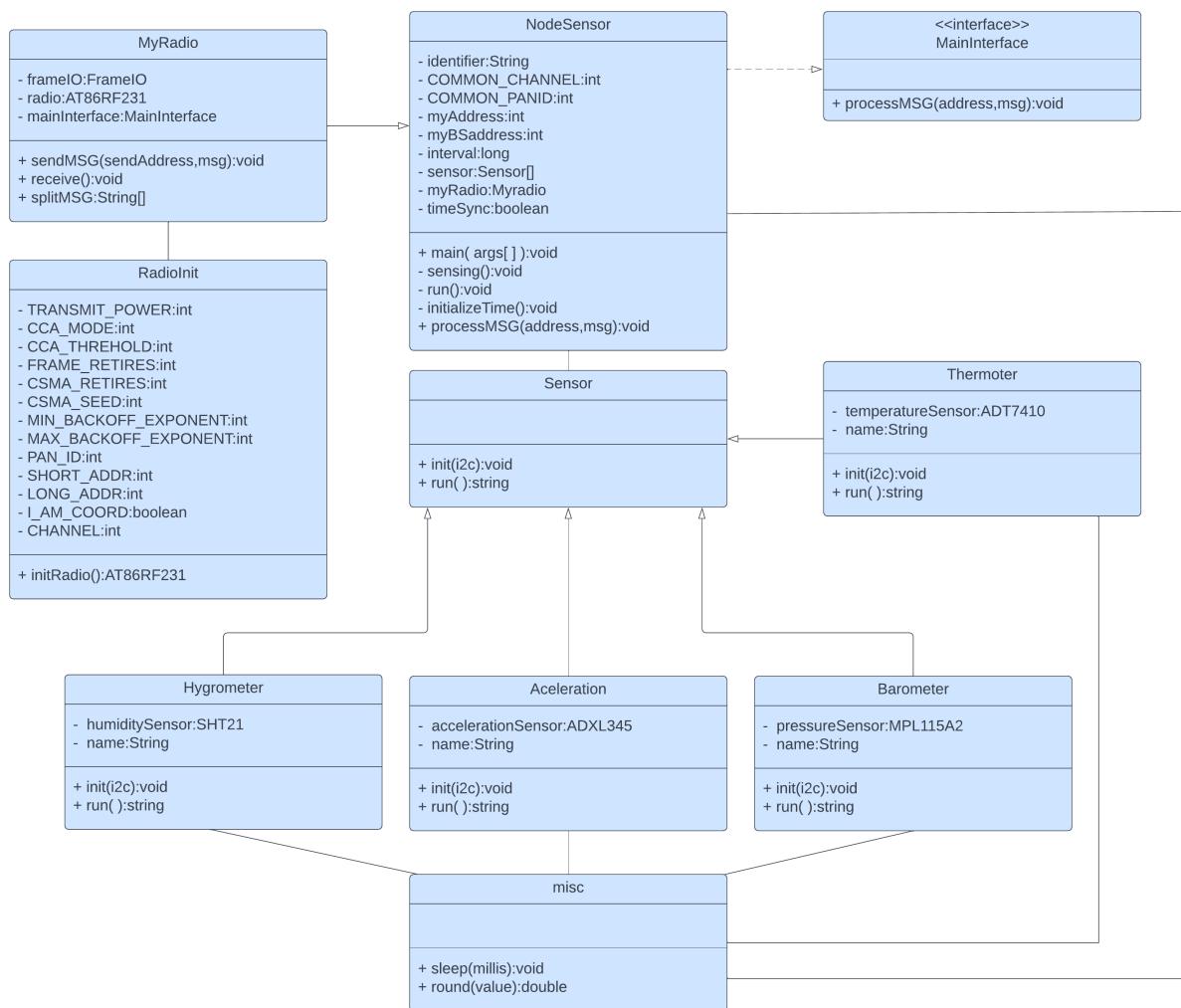
Endpoint	/realTime
Method	POST
Tujuan	Mengambil data yang terbaru untuk visualisasi
Masukan	Informasi <i>node</i> sensor yang akan diambil
Keluaran	Data hasil <i>sensing</i> yang terbaru

4.3 Perancangan Perangkat Lunak pada WSN

Pada Bagian WSN terdapat empat buah program. Berikut merupakan rancangan kelas diagram dan penjelasannya untuk setiap program.

4.3.1 Node Sensor pada Preon32

Perangkat lunak ini diunggah pada WSN yang dijadikan sebagai *node*. Perangkat lunak ini didesain khusus untuk perangkat Preon32. Perangkat lunak ini secara sederhana berfungsi untuk melakukan *sensing* dalam suatu rentang kemudian data *sensing* dikirimkan ke *base station*. Perangkat lunak ini dirancang dalam sebuah diagram kelas untuk menunjukkan interaksi antar kelas. Diagram kelas dari perangkat lunak ini dapat dilihat pada gambar 4.14. Berikut adalah penjelasan kelas diagram mengenai kelas, method, dan atribut:



Gambar 4.14: Kelas Diagram *node* sensor pada Preon32

- Kelas NodeSensor merupakan kelas utama dari perangkat lunak. Kelas ini berfungsi untuk menangani inisialisasi awal aktif, memfasilitasi interaksi antar kelas dan memerintahkan untuk melakukan *sensing*.

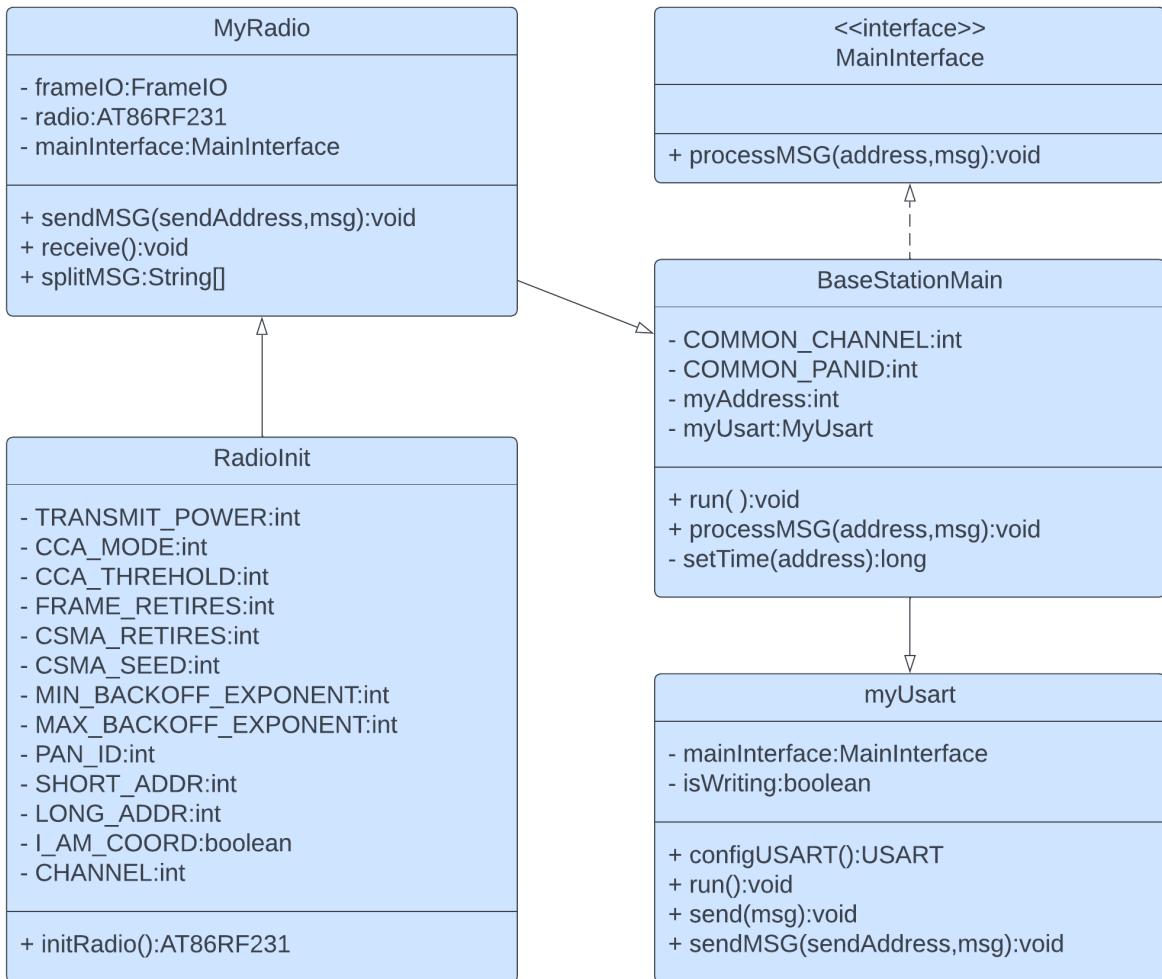
- Variabel Identifier
Sebuah variabel yang merupakan penanda *global node* tersebut. Variabel ini akan digunakan sebagai penanda antara *node* baik yang terhubung dengan satu *base* maupun yang tidak terhubung satu *base*.
 - Variabel COMMON_CHANNEL
Variabel yang diisikan nilai *channel* yang akan dipakai oleh radio.
 - Variabel COMMON_PANID
Variabel yang diisikan nilai dari sebuah node terhubung dengan jaringan yang mana atau berapa.
 - Variabel myAddress
Variabel yang merupakan nilai alamat *node* tersebut. Variabel ini juga sebagai penanda tetapi hanya bersifat lokal atau yang terhubung dengan satu *channel* dan satu PAN_ID.
 - Variabel Interval
Sebuah variabel yang menampung nilai selang waktu *node* akan melakukan *sensing* dan mengirimkan data.
 - Variabel Sensor
Sebuah variabel objek kumpulan sensor-sensor yang terdapat pada *node*.
 - Variabel myRadio
Sebuah variabel objek yang menampung objek radio yang digunakan untuk berkomunikasi.
 - Variabel timeSync
Sebuah variabel yang digunakan pada saat *node* baru aktif untuk mengecek apakah waktu dan interval sudah di sinkronisasi
 - Fungsi main
Sebuah fungsi yang akan dijalankan pertama saat *node* aktif, fungsi ini digunakan untuk melakukan inisialisasi atribut atau variabel kelas NodeSensor.
 - Fungsi sensing
Sebuah fungsi yang bertugas untuk melakukan *sensing* dan membuat *string* hasil *sensing*.
 - Fungsi run
Fungsi yang akan dijalankan setelah inisialisasi selesai oleh fungsi main. Fungsi ini akan menjalankan prosedur nyala seperti memanggil fungsi initializeTime untuk mendapatkan waktu dan interval, setelah waktu dan interval didapatkan baru *sensing* setiap rentang waktu dijalankan.
 - Fungsi processMSG
Fungsi yang akan menangani pesan-pesan yang diterima oleh radio.
- Kelas Sensor merupakan kelas abstrak yang akan dijadikan *parent class* oleh kelas lain yaitu Thermometer, Hygrometer, Accelerometer, dan Barometer. Kelas ini memiliki dua fungsi abstrak yaitu init dan run. Fungsi init bertujuan untuk melakukan inisialisasi sensor masing-masing sedangkan run untuk mengambil data *sensing* sesuai dengan jenis sensor.
 - Kelas MyRadio merupakan kelas yang memiliki fungsi untuk menerima dan mengirimkan pesan ke atau dari radio. Kelas ini memiliki atribut dan fungsi sebagai berikut:
 - Variabel frameIO
Sebuah variabel objek yang bertujuan untuk membuat sebuah *frame* Zigbee.
 - Variabel radio
Sebuah variabel yang menampung objek *driver* radio.
 - Variabel mainInterface
Sebuah variabel objek interface agar radio dapat berinteraksi dengan kelas NodeSensor.
 - Fungsi send
Sebuah fungsi ini bertugas untuk mengirimkan pesan ke tujuan.
 - Fungsi receive
Sebuah fungsi yang akan menerima pesan. Saat pesan diterima fungsi ini akan memecah pesan menggunakan fungsi splitMSG kemudian diteruskan ke MainInterface

- splitMSG
Sebuah fungsi yang akan melakukan *decode* sebuah pesan agar dapat dilanjutkan sesuai tujuannya atau kebutuhannya.
- kelas Misc singkatan dari *miscellaneous* merupakan kelas yang berisikan fungsi-fungsi yang umum seperti sleep yang berfungsi untuk menghentikan proses suatu *thread* selama sekian milidetik dan fungsi round yang bertugas untuk membulatkan sebuah desimal menjadi desimal dengan presisi dua.
- Interface MainInterface merupakan sebuah antarmuka yang berfungsi untuk menghubungkan *thread receiver* radio agar dapat diproses dikelas NodeSensor.
- Kelas RadioInit merupakan kelas yang disediakan oleh Virtenio, kelas ini bertugas untuk melakukan inialisasi konfigurasi radio. Kelas ini memiliki banyak variabel-variabel konfigurasi seperti CSMA_RETRIES yaitu variabel yang menyimpan jika terjadi tabrakan akan dilakukan pengulangan hingga sekian kali, variabel TRANSMIT_POWER merupakan variabel yang menyimpan nilai kekuatan transmisi yang akan digunakan oleh radio, dan lain sebagainya.

4.3.2 Base Station pada Preon32

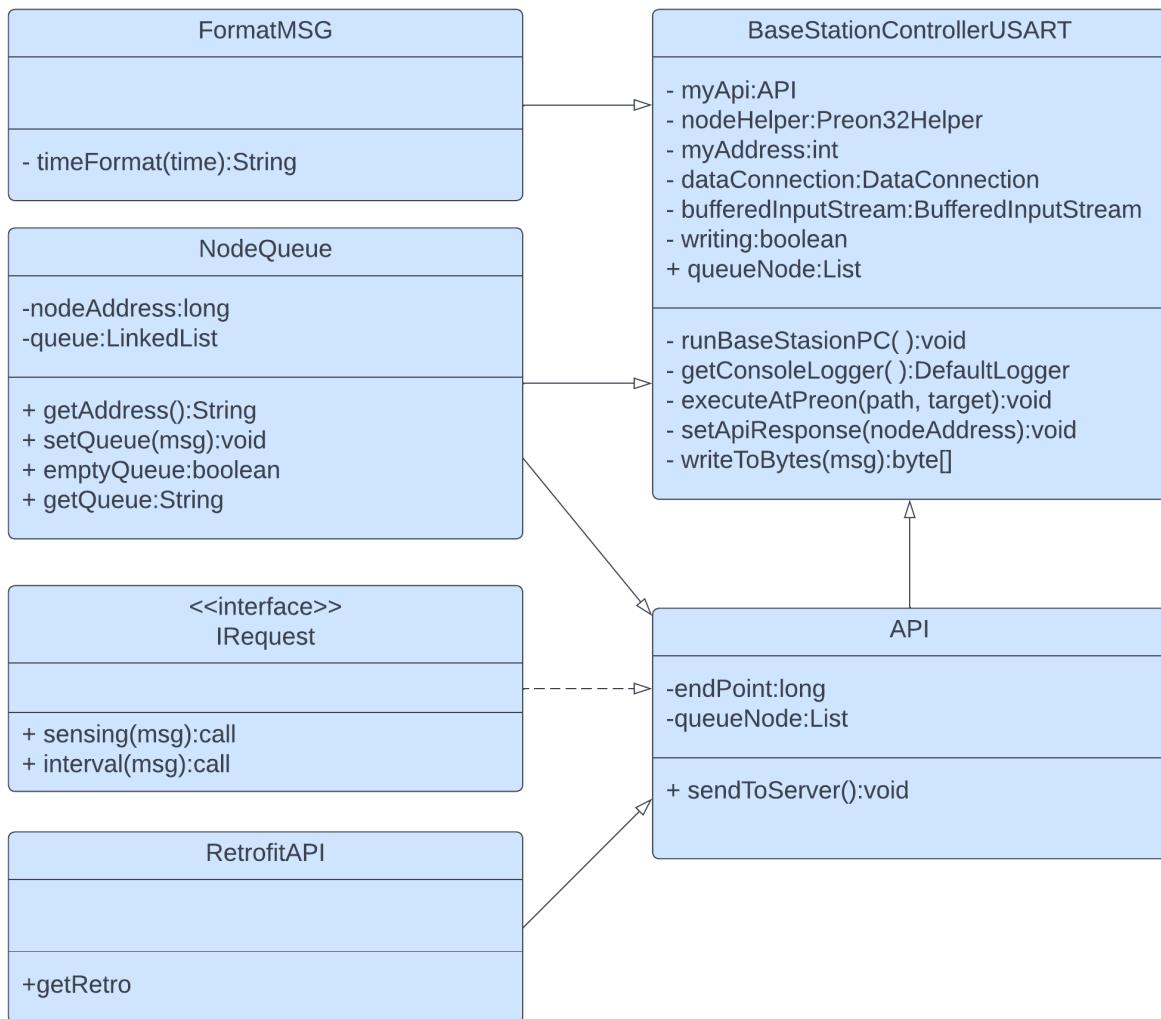
Perangkat lunak ini diunggah pada WSN yang dijadikan sebagai *base station*. Perangkat lunak ini didesain khusus untuk perangkat Preon32. Perangkat lunak ini secara sederhana berfungsi untuk menerima data yang dikirimkan oleh *node* sensor kemudian diteruskan ke komputer yang terhubung. Perangkat lunak ini dirancang dalam sebuah diagram kelas untuk menunjukkan interaksi antar kelas. Diagram kelas dari perangkat lunak ini dapat dilihat pada gambar 4.15. Berikut adalah penjelasan kelas diagram mengenai kelas, method, dan atribut:

- Kelas BaseStationMain merupakan kelas yang akan dijalankan pertama tugas nya untuk menginisiasi, dan memberikan fasilitas interaksi antar kelas. Kelas BaseStationMain memiliki atribut dan fungsi sebagai berikut:
 - Variabel COMMON_CHANNEL
Sebuah variabel yang diisikan nilai *channel* yang akan dipakai oleh radio.
 - Variabel COMMON_PANID
Sebuah variabel yang diisikan nilai dari sebuah *base* terhubung dengan jaringan yang mana atau berapa.
 - Variabel myAddress
Sebuah variale berisikan nilai alamat *base* tersebut. Variabel ini juga sebagai penanda tetapi hanya bersifat lokal atau yang terhubung dengan satu *channel* dan satu PAN_ID.
 - Fungsi run
Fungsi yang bertugas untuk mengaktifkan radio dan USART.
 - Fungsi processMSG
Fungsi yang bertugas untuk menentukan apakah sebuah pesan akan diteruskan atau cukup *base station* yang memberikan balasan.
 - setTime fungsi
Fungsi yang bertugas untuk memberikan balasan waktu ke *node* dan meminta interval *node* ke *server*
- Kelas myUsart merupakan kelas yang berfungsi untuk berkomunikasi melalui antar muka fisik *micro usb*.
- Interface MainInterface merupakan sebuah antarmuka yang berfungsi untuk menghubungkan *thread receiver* radio dan *thread receiver* usart agar dapat diproses di kelas BaseStationMain. *thread receiver* radio ataupun usart agar dapat diproses dikelas BaseStationMain.
- Kelas MyRadio kelas ini memiliki atribut dan fungsi yang sama seperti pada bagian 4.3.1.
- Kelas RadioInitkelas ini memiliki atribut dan fungsi yang sama seperti pada bagian 4.3.1.

Gambar 4.15: Kelas Diagram *Base Station* pada Preon32

4.3.3 *Base Station* pada Komputer

Perangkat lunak ini berjalan pada komputer. Perangkat lunak ini secara sederhana berfungsi untuk menerima data yang dikirimkan oleh *base* sensor melalui USART kemudian diteruskan ke server yang terhubung. Perangkat lunak menggunakan *library* Retrofit untuk bertukar pesan dengan *server*. Perangkat lunak menggunakan *library* Retrofit karena *library* ini umumnya digunakan diperangkat bergerak *mobile* yang performanya terbatas, hal ini bertujuan agar perangkat seperti Raspberry pi dapat menjalankan perangkat lunak ini. Perangkat lunak ini dirancang dalam sebuah diagram kelas untuk menunjukkan interaksi antar kelas. Diagram kelas dari perangkat lunak ini dapat dilihat pada gambar 4.16. Berikut adalah penjelasan kelas diagram mengenai kelas, method, dan atribut:



Gambar 4.16: Kelas Diagram *Base Station* pada Komputer

- **BaseStationControllerUSART** merupakan kelas utama yang berfungsi untuk menerima pesan dari USART kemudian diteruskan keserver dan berfungsi juga untuk mengaktifkan Preon32 yang berperan sebagai *base station*. Perangkat lunak ini dirancang dalam sebuah diagram kelas untuk menunjukkan interaksi antar kelas. Kelas ini memiliki variabel dan method sebagai berikut:
 - Variabel `myAPI`
Sebuah variabel yang menyimpan objek API.
 - Variabel `nodeHelper`
Sebuah variabel yang menyimpan objek Preon32Helper. Preon32Helper ini berfungsi untuk menjembatani antara komputer dengan perangkat Preon32 menggunakan antarmuka berjenis serial.
 - Variabel `dataConnection`
Sebuah variabel yang memerintahkan perangkat lunak *base station* pada Preon32 untuk beroperasi dan juga mengambil segala keluaran yang dihasilkan oleh Preon32.
 - Variabel `bufferInputStream`
Sebuah variabel yang digunakan untuk menampung sementara pesan dari antar muka serial agar dapat dibaca secara utuh.

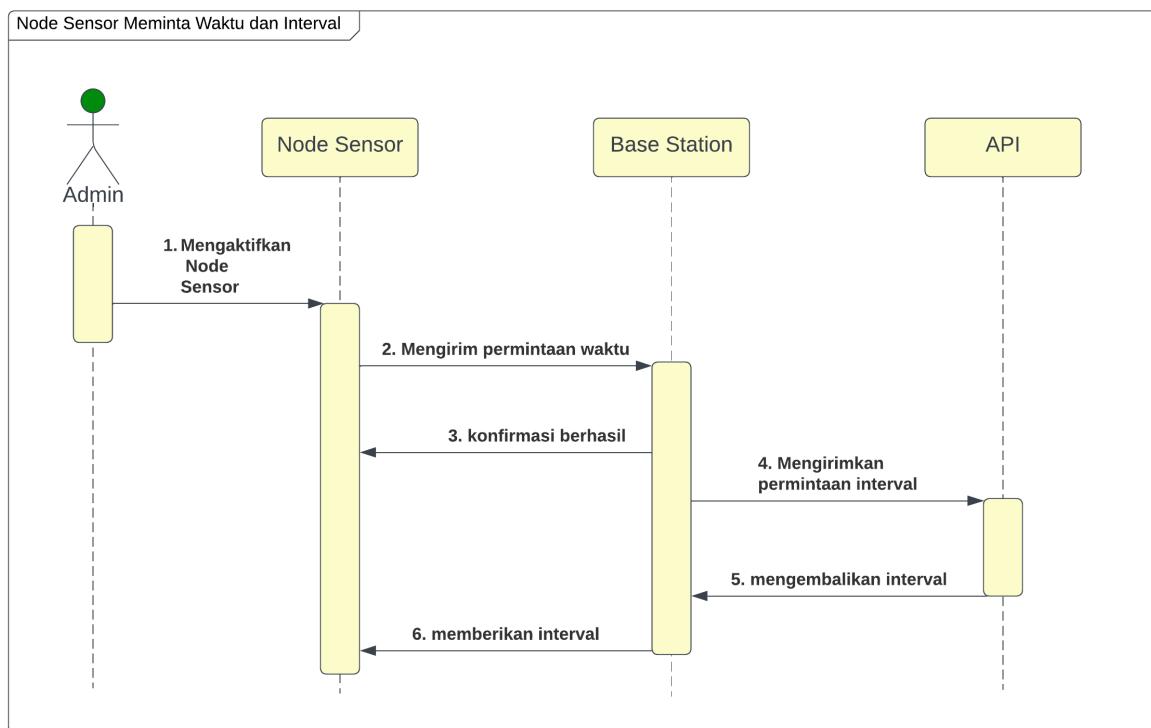
- Variabel writing
Sebuah variabel berfungsi agar tidak ada yang menggunakan antar muka serial secara bersamaan, hal ini dikarenakan adanya kemungkinan terjadi *race condition* jika ada yang secara bersamaan menggunakan antarmuka serial.
 - Variabel queueNode
Sebuah variabel merupakan queue yang digunakan untuk menampung sementara balasan dari *server* diluar balasan "benar" atau "salah". Balasan butuh ditampung sementara dikarenakan jika menunggu *server* untuk membalas maka *blocking* yang terjadi terlalu panjang. *Blocking* yang panjang menyebabkan pesan lain baik dari *node* yang sama maupun berbeda tidak terkirim.
 - Fungsi runBaseStationPC
Sebuah fungsi yang digunakan untuk menerima dan memproses pesan dari antarmuka serial dengan USART.
 - Fungsi getConsoleLogger
Sebuah fungsi yang dijalankan satu kali pada saat awal perangkat lunak dijalankan berfungsi untuk melakukan inisialisasi pada *logger*.
 - Fungsi executeAtPreon
Sebuah fungsi yang dijalankan satu kali pada saat awal perangkat lunak dijalankan berfungsi untuk mengambil file eksekusi ant yang disediakan dalam dokumentasi Preon32. Setelah file eksekusi ant diambil kemudian dieksekusi sesuai targetnya.
 - Fungsi setApiResponse
Sebuah fungsi yang bertugas untuk mengembalikan balasan ke perangkat lunak *base station* pada Preon32.
 - Fungsi writeToBytes
Sebuah fungsi yang bertugas untuk memastikan pesan memiliki panjang selalu sama yaitu 128 karakter.
- NodeQueue sebuah kelas objek yang berfungsi untuk menampung dahulu balasan untuk sebuah *node* diluar balasan "benar" atau "salah". Kelas ini memiliki variabel dan method sebagai berikut:
 - Variabel nodeAddress
Sebuah variabel yang dipakai untuk memisahkan *queue* antar *node*.
 - Variabel queue
Sebuah variabel yang dipakai untuk menampung balasan dari *server*.
 - Fungsi getAddress
Sebuah fungsi yang digunakan untuk mengembalikan alamat *node*.
 - Fungsi setQueue
Fungsi yang digunakan untuk mengisi *queue*
 - Fungsi emptyQueue
Sebuah fungsi yang digunakan untuk mengembalikan apakah fungsi kosong atau tidak.
 - Fungsi getQueue
Sebuah fungsi yang digunakan untuk mengembalikan isi dari *queue*.
 - Format MSG sebuah kelas yang berfungsi untuk melakukan format pesan dari milidetik ke format waktu pada umumnya.
 - API sebuah kelas yang bertugas untuk mengirimkan pesan menuju ke *server*. Kelas ini memiliki variabel dan method sebagai berikut:
 - Variabel endpoint sebuah variabel berisikan alamat *server*.
 - queueNode sebuah variabel objek yang akan diisi oleh balasan *server*.
 - Fungsi sendToServer sebuah fungsi untuk melakukan pengiriman pesan ke *server*.
 - Interface IRequest merupakan antarmuka yang Fungsi-Fungsinya dipanggil sesuai dengan *endpoint* tujuannya.
 - RetrofitAPI sebuah kelas yang digunakan untuk melakukan inisialisasi *library* retrofit.

4.3.4 Node Sensor pada Komputer

Perangkat lunak ini berjalan pada komputer. Perangkat lunak ini secara sederhana berfungsi untuk mengaktifkan program yang sebelumnya sudah diunggah ke Preon32. Perangkat keras ini dapat dijalankan di komputer maupun mini komputer seperti Raspberry pi.

4.3.5 Perancangan Sinkronisasi Waktu dan Interval

Pada perangkat Preon32 terdapat modul RTC, namun pada perangkat tidak terdapat baterai yang khusus hanya untuk memberikan energi untuk modul RTC. Sehingga modul RTC selalu kembali ke *epoch time* pada saat diaktifkan. Berikut merupakan urutan sinkronisasi waktu dan interval yang digambarkan pada sebuah diagram *sequence*.



Gambar 4.17: Diagram *Sequence* Sinkronisasi Waktu dan Interval pada *Node*

Pada saat pertama dinyalakan karena Preon32 tidak terdapat RTC maka *node sensor* harus meminta informasi waktu dan informasi interval kepada *base station*. *Base station* karena sebelumnya sudah melakukan sinkronisasi waktu dengan komputer yang mengaktifkannya maka *base station* secara langsung yang memberikan informasi waktu. Kemudian karena *base station* tidak memiliki infomasi interval *node sensor* maka *base station* meminta ke *server* untuk diberikan interval. Setelah didapatkan interval maka *base station* tinggal melanjutkan informasi interval ke *node* yang meminta.

BAB 5

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini membahas mengenai implementasi perangkat lunak yang terdiri dari Lingkungan perangkat keras, Program, *library*, *framework* yang digunakan. Kemudian akan dilanjutkan dengan pengujian terhadap aplikasi.

5.1 Implementasi

Pada bagian ini akan dicantumkan data-data berupa spesifikasi perangkat keras yang digunakan penulis untuk membuat , detail versi program, bahasa pemrograman dan *library* dan*framework*. Kemudian akan d oleh implementasi dari tampilan yang dibuat.

5.1.1 Lingkungan Perangkat Keras

Perangkat Keras yang digunakan untuk membangun dan menguji perangkat lunak ini adalah MSI Modern 14 A10rb dengan spesifikasi:

Tabel 5.1: Tabel Lingkungan Perangkat Keras

Prosesor	Intel(R) Core(TM) i5-10210U
RAM	32768MB
VGA	Intel(R) UHD 620 & NVIDIA GeForce MX250

5.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan untuk membangun dan menguji perangkat lunak ini terbagi menjadi dua yaitu program-program pada tabel 5.2 dan *library* dan*framework* pada tabel 5.3.

Tabel 5.2: Tabel Lingkungan Perangkat Lunak

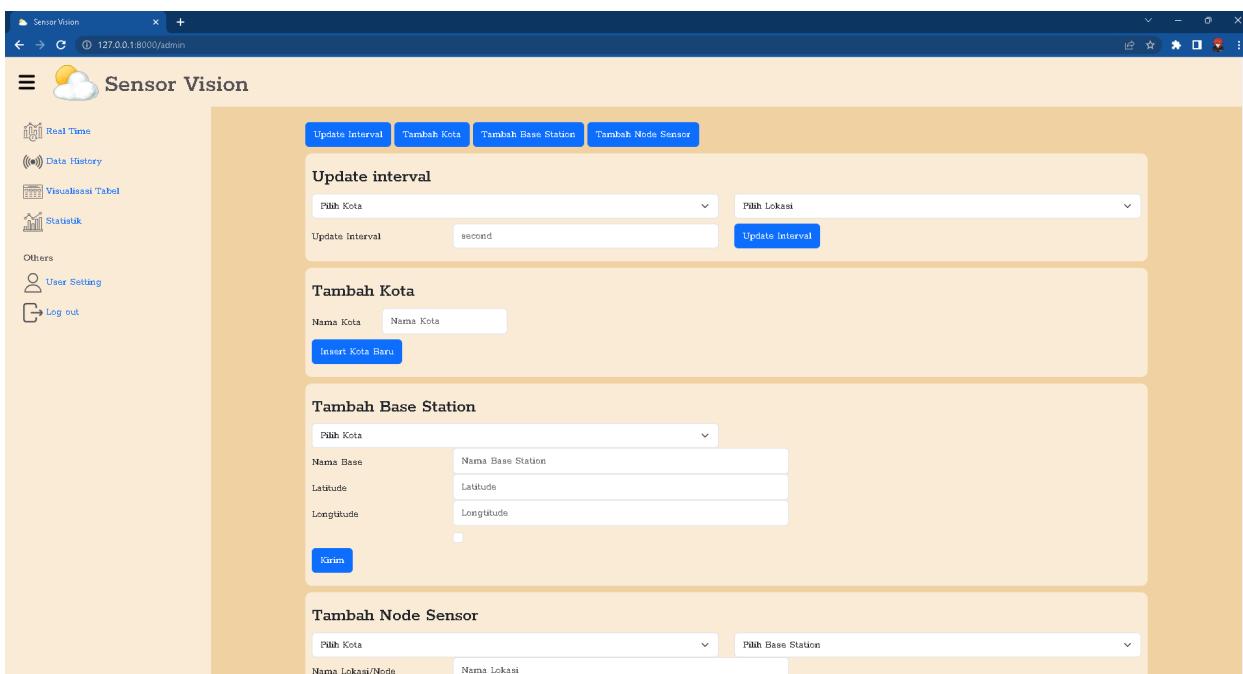
Sistem Operasi	Windows 11 Pro 64-bit (10.0, Build 22621)
Python	3.9.7
Java	17.0.3 2022-04-19 LTS
MySQL	8.0.32
PhpMyAdmin	5.1.3
Google Chrome	114.0.5735.111
IntelliJ IDEA	2022.1.4 (Ultimate Edition)
Visual Studio Code	1.78.2

Tabel 5.3: Tabel Lingkungan Library dan Framework

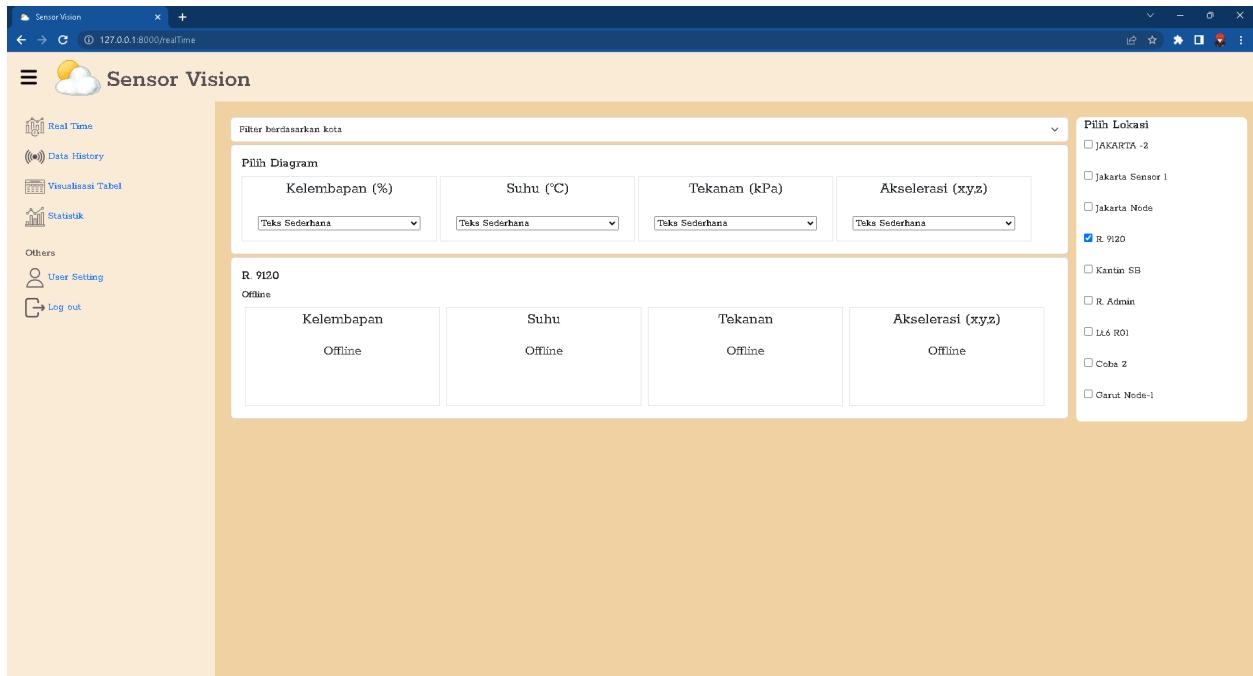
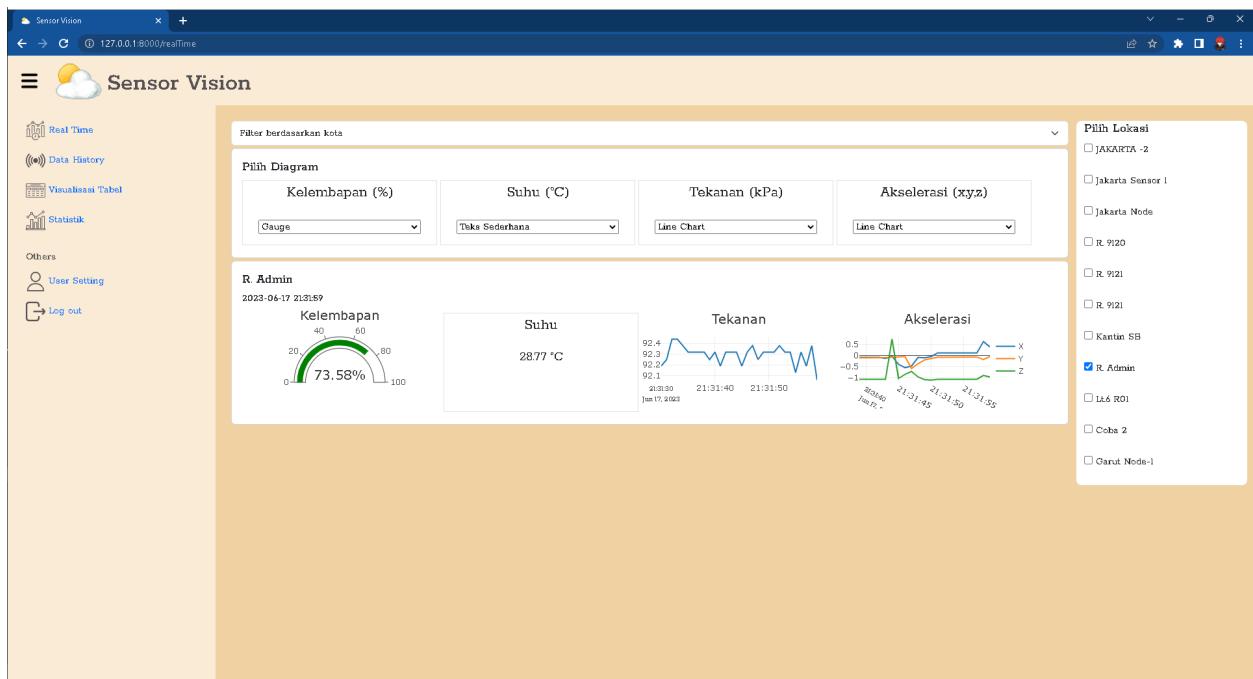
Laravel	8.8.2
Plotly	2.23.2
mysql-connector-python	8.0.29
uvicorn	0.20.0
fastapi	0.87.0
bcrypt	3.2.0
numpy	1.21.4
ant	1.10.13
retrofit2	2.90

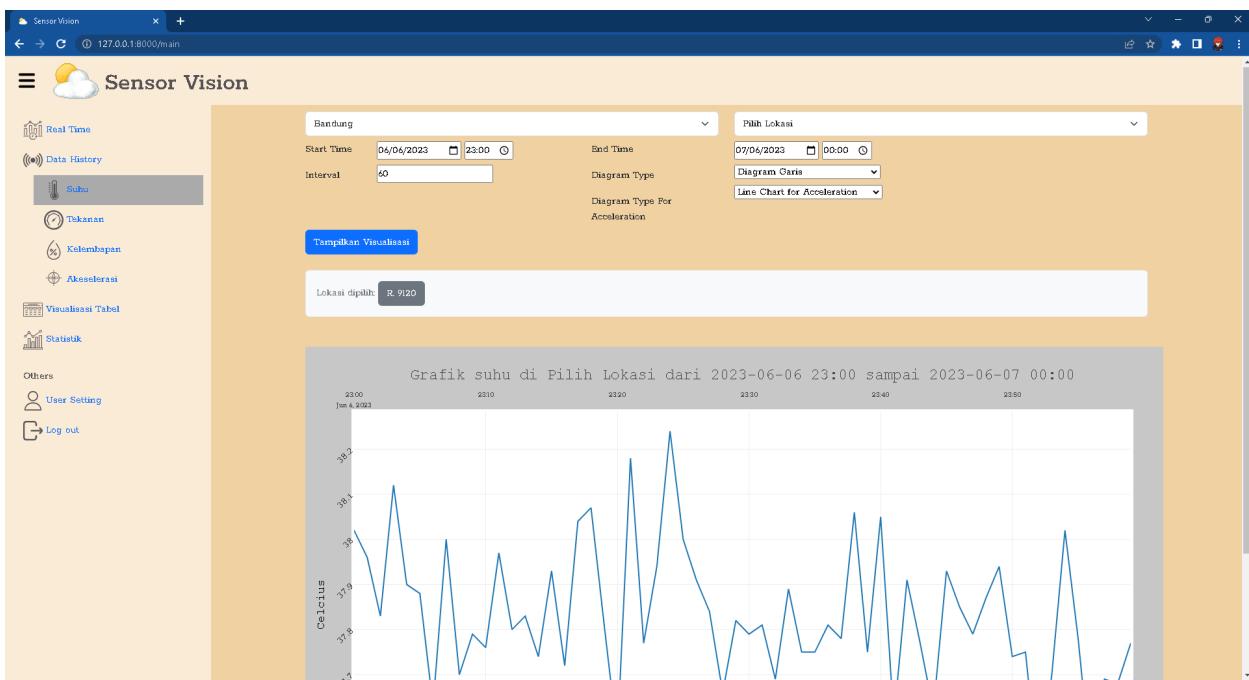
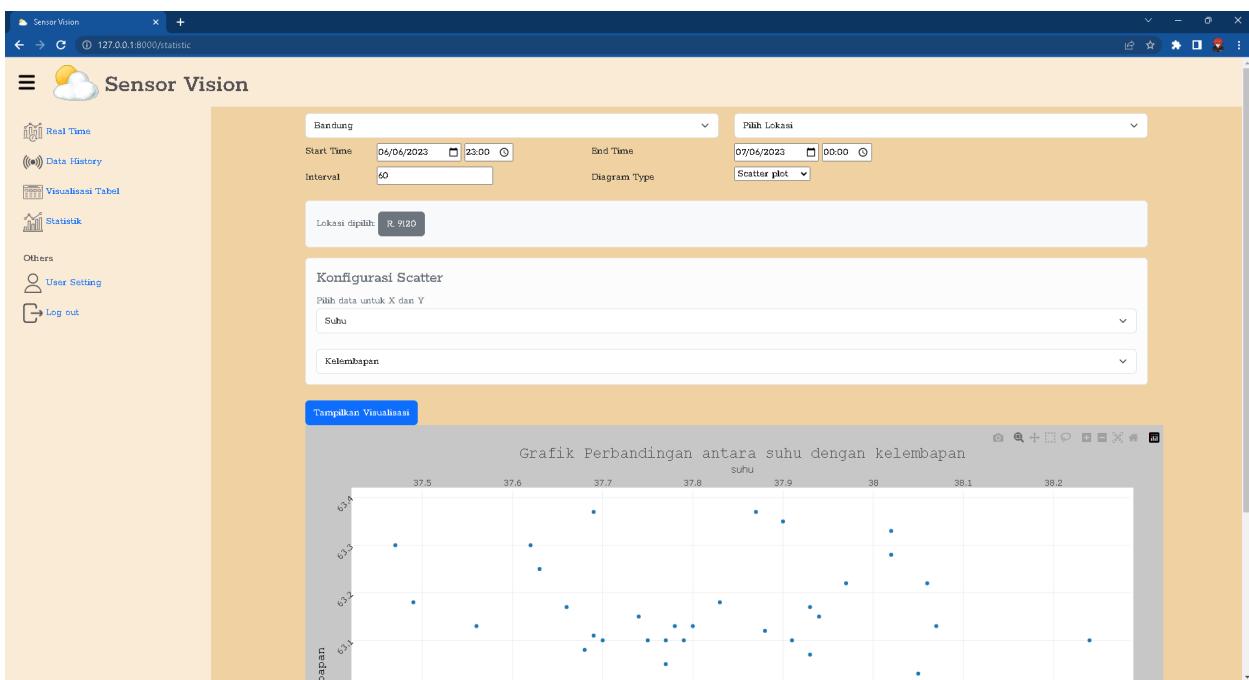
5.1.3 Implementasi tampilan

Berikut akan ditampilkan hasil dari implementasi berupa tangkapan layar website:



Gambar 5.1: Tangkapan Layar Halaman Admin

Gambar 5.2: Tangkapan Layar Halaman *Realtime*Gambar 5.3: Tangkapan Layar Halaman *Realtime Aktif*

Gambar 5.4: Tangkapan Layar Halaman *History*Gambar 5.5: Tangkapan Layar Halaman *Statistik*

The screenshot shows a web-based application titled "Sensor Vision". On the left, there is a sidebar with icons for Real Time, Data History, Visualisasi Tabel, Statistik, User Setting, and Log out. The main area has a title "Bandung" and dropdown menus for "Start Time" (06/06/2023, 23:00), "End Time" (07/06/2023, 00:00), and "Interval" (60). A blue button labeled "Tampilkan Tabel" is visible. Below this, a table lists data for 24 rows, each with a timestamp, temperature, humidity, pressure, and acceleration values.

Tanggal	Suhu	Kehilangan	Tekanan	Akselerasi (X,Y,Z)*
2023-05-08T00:00:00	29.75°C	72.63%	92kPa	(-2.21, -113.34, 207.38)
2023-05-08T00:16:41	29.62°C	73%	92kPa	(-2.31, -113.43, 207.47)
2023-05-08T00:33:51	29.93°C	71.97%	92kPa	(0.09, -110.41, 209.23)
2023-05-08T00:50:31	30.09°C	71.42%	92kPa	(-3.97, -46.89, 222.78)
2023-05-08T01:07:11	29.67°C	72.33%	92kPa	(-7.22, 319, 233.69)
2023-05-08T01:23:52	29.68°C	72.01%	92kPa	(-7.48, 334, 233.36)
2023-05-08T01:40:32	29.84°C	71.41%	92kPa	(-7.82, 317, 233.58)
2023-05-08T01:57:12	29.88°C	71.81%	92kPa	(-7.82, 342, 233.61)
2023-05-08T02:42:00	29.12°C	72.95%	92kPa	(-7.95, 326, 234.04)
2023-05-08T02:58:40	29.33°C	71.88%	92kPa	(-7.79, 334, 233.71)
2023-05-08T03:15:21	29.22°C	71.67%	92kPa	(-8.05, 336, 233.72)
2023-05-08T03:32:01	29.06°C	71.91%	92kPa	(-7.69, 319, 233.92)
2023-05-08T03:48:42	28.94°C	72%	92kPa	(-7.64, 325, 233.75)
2023-05-08T04:05:22	28.95°C	72%	92kPa	(-7.8, 311, 233.82)
2023-05-08T04:22:03	28.94°C	72%	92kPa	(-7.97, 315, 233.67)
2023-05-08T04:38:43	28.97°C	71.59%	92kPa	(-7.82, 324, 233.89)

Gambar 5.6: Tangkapan Layar Halaman Tabel

The screenshot shows a password change form within the "Sensor Vision" application. The sidebar on the left includes icons for Real Time, Data History, Visualisasi Tabel, Statistik, User Setting, and Log out. The main form has fields for "Username" (dimas), "Password Lama" (Old Password), "Password Baru" (New Password), and a "Ganti Password" (Change Password) button. There is also a small sun icon above the input fields.

Gambar 5.7: Tangkapan Layar Halaman Ganti Kata Sandi

5.1.4 Implementasi API

Berikut ditampilkan pemanggilan API dikuti contoh masukan dan keluarannya :

Tabel 5.4: Implementasi API untuk Login

Endpoint	/login
Method	POST
Tujuan	Melakukan verifikasi user atau login
Contoh masukan	<pre>{ "username": "DimasKurniawan", "password": "Password", }</pre>
Contoh keluaran	<pre>{ "result": 0 } atau { "result": 1 } atau { "result": -8 }</pre>

Tabel 5.5: Implementasi API untuk Daftar

Endpoint	/signUp
Method	POST
Tujuan	Melakukan Pendaftaran
Contoh masukan	<pre>{ "username": "DimasKurniawan", "password": "Password", "email": "6181901019@student.unpar.ac.id" }</pre>

Contoh keluaran	<pre>{ "result": true }</pre> <p>atau</p> <pre>{ "result": false }</pre>
-----------------	--

Tabel 5.6: Implementasi API untuk Memasukan Kota Baru

Endpoint	/kota
Method	POST
Tujuan	Menyimpan kota baru
Contoh masukan	<pre>{ "nama": "Bandung" }</pre>
Contoh keluaran	<pre>{ "result": true }</pre>

Tabel 5.7: Implementasi API untuk mengambil Seluruh Kota

Endpoint	/kota
Method	GET
Tujuan	mengambil seluruh kota
Contoh masukan	
Contoh keluaran	<pre>{ "result": [{ "id": 3, "nama": "Jakarta" }, { "id": 4, "nama": "Bandung" }] }</pre>

Tabel 5.8: Implementasi API untuk Memasukan Data *Sensing*

Endpoint	/sensing
Method	POST
Tujuan	<i>Base station</i> mengirimkan pesan hasil <i>sensing</i>
Contoh masukan	<pre>{ "time": "2023-05-15 01:28:27", \ "key": "CtB2oTRQFV", "id": "aaaa", "result": { "rh": 90.02, "T": 33,33, "Pa": 63, "a": ["0.01", "0.03", "0.99"] } }</pre>
Contoh keluaran	<pre>{ "result": true } atau { "result": false } atau { "setInterval": <milidetik> }</pre>

Tabel 5.9: Implementasi API untuk Memasukan *Base Station* Baru

Endpoint	/bs
Method	POST
Tujuan	Menyimpan <i>base station</i> baru
Contoh masukan	<pre>{ "nama": "UNPAR", "latitude": "6.875213", "longitude": "107.60502", "idKota": 4 }</pre>
Contoh keluaran	<pre>{ "result": true } atau { "result": false }</pre>

Tabel 5.10: Implementasi API untuk Mengambil Seluruh *Base Station*

Endpoint	/bs
Method	GET
Tujuan	Mengambil seluruh <i>base station</i>
Contoh masukan	
Contoh keluaran	<pre>{ "result": [{ "id": 2, "nama": "UNPAR GEDUNG 9", "latitude": "-6.87500773326699", "longitude": "107.60666052567069", "idKota": 4 }, { "id": 3, "nama": "UNPAR GEDUNG 10", "latitude": "-6.87500773326699", "longitude": "107.60666052567069", "idKota": 4 }] }</pre>

Tabel 5.11: Implementasi API untuk Memasukan *Node* Sensor Baru

Endpoint	/node
Method	POST
Tujuan	Menyimpan <i>node</i> sensor baru
Contoh masukan	<pre>{ "identifier": "AAAt" , "idBS": 39 , "nama": "Lt. 6 R01" , "indoor": 0 , "interval": 3000 , "tipeSensor": ["Suhu", "Kelembapan", "Tekanan", "Akselerasi"] }</pre>
Contoh keluaran	<pre>{ "result": { "success": true, "key": "CtB2oTRQFV" } } atau { "result": false }</pre>

Tabel 5.12: Implementasi API untuk Mengambil Seluruh *Node* Sensor

Endpoint	/node
Method	GET
Tujuan	Mengambil seluruh informasi <i>node</i> sensor
Contoh masukan	?idBS=-1
Contoh keluaran	<pre>{ "result": [{ "identifier": "aaaa", "interval": 1000, "idKota": 4, "namaLBase": "UNPAR GEDUNG 9", "latitude": "-6.87500773326699", "longitude": "107.60666052567069", "indoors": 1, "namaKota": "Bandung", "namaLokasi": "R. 9120" }, { "identifier": "aaab", "interval": 5000, "idKota": 4, "namaLBase": "UNPAR GEDUNG 10", "latitude": "-6.87500773326699", "longitude": "107.60666052567069", "indoors": 0, "namaKota": "Bandung", "namaLokasi": "Kantin SB" }] }</pre>

Tabel 5.13: Implementasi API untuk Mengganti Selang Waktu *Sensing Node* Sensor

Endpoint	/update
Method	POST
Tujuan	Mengganti selang waktu <i>sensing node</i> sensor
Contoh masukan	<pre>{ "idBS": "AAAc", "command": "setInterval:6000" }</pre>
Contoh keluaran	<pre>{ "result": true }</pre>

Tabel 5.14: Implementasi API untuk Mengambil Interval Waktu *Node Sensor*

Endpoint	/interval
Method	GET
Tujuan	Mengambil interval waktu <i>node sensor</i>
Contoh masukan	?node=aaaa
Contoh keluaran	<pre>{ "setInterval:6000" }</pre>

Tabel 5.15: Implementasi API untuk Mengirim Data *Sensing*

Endpoint	/sensing
Method	POST
Tujuan	Untuk mengirim data <i>sensing</i>
Contoh masukan	<pre>{ "time": "2023-10-01 12:28:27", "id": "AAAt", "key": "0e02khvvgj", "result": { "rh": 80, "T": -32, "Pa": -23, "a": [9.80, 9.80, 9.80] } }</pre>
Contoh keluaran	<pre>{ "result": true }</pre> <p>atau</p> <pre>{ "result": false }</pre> <p>atau</p> <pre>{ "setInterval": 2000 }</pre>

Tabel 5.16: Implementasi API untuk Mengambil Data *Sensing*

Endpoint	/data
Method	POST
Tujuan	Untuk mengambil data <i>sensing</i>
Contoh masukan	<pre>{ { "node": "aaac", "start": "2023-06-06 00:00:00", "end": "2023-06-07 00:00:00", "interval": 1000 "stat": "stat" } }</pre>
Contoh keluaran	<pre>{ "result": [{ "timeStamp": "2023-06-06T04:24:01", "suhu": 32.32, "kelembapan": 65.39, "tekanan": 92.0, "akselerasi": { "x": -0.0, "y": -0.0, "z": 0.13 } }, { "timeStamp": "2023-06-06T04:40:41", "suhu": 33.18, "kelembapan": 63.77, "tekanan": 92.0, "akselerasi": { "x": 0.06, "y": -0.03, "z": 0.9 } }] }</pre>

Tabel 5.17: Implementasi API untuk Mengambil Data *Sensing Realtime*

Endpoint	/realTime
Method	GET
Tujuan	Untuk mengambil data <i>sensing</i> yang terbaru (<i>realtime</i>)
Contoh masukan	?node=aaaa
Contoh keluaran	<pre>{ "time": "1970-01-01 00:00:00", "node": "aaac", "result": { "kelembapan": null, "tekanan": null, "suhu": null, "akselerasi": null } }</pre>

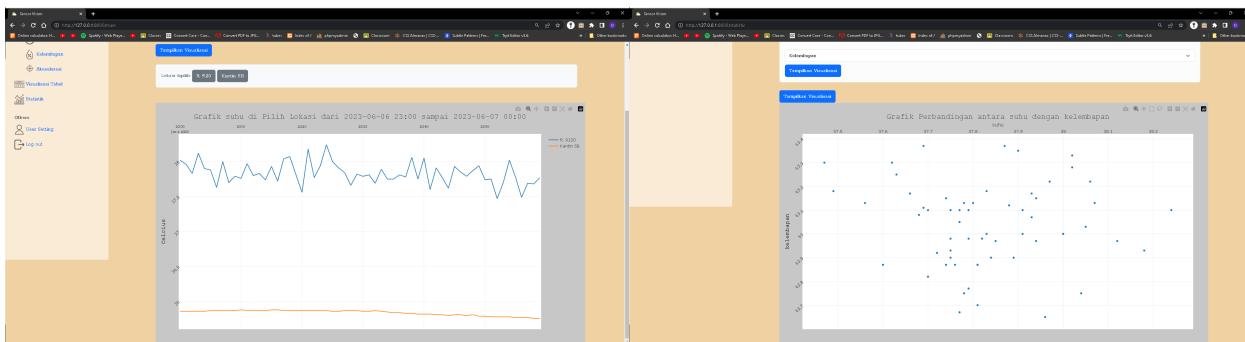
5.2 Pengujian

Pada bagian pengujian akan dilakukan dua buah pengujian yaitu pengujian fungsionalitas yang menguji apakah seluruh fungsi-fungsi dapat berjalan sebagai mana mestinya. Kemudian pengujian kedua yaitu pengujian eksperimental yang menguji bagaimana perangkat lunak menangani *error*.

5.2.1 Pengujian Fungsionalitas

Pada pengujian fungsionalitas akan dilakukan 6 buah pengujian. Berikut merupakan pengujian fungsionalitas yang dilakukan:

- Menampilkan berbagai visualisasi data



Gambar 5.8: Tangkapan Layar Website Menampilkan Diagram Garis
Gambar 5.9: Tangkapan Layar Website Menampilkan Scatter Plot

Hasil: berhasil ditampilkan, hal ini menandakan *framework* visualisasi data, website, API, dan basis data dapat bekerja

- Mencoba menganti interval kepada *node* sensor. Harapan yaitu *node* sensor mengganti intervalnya sesuai dengan diisi oleh admin. Percobaan dilakukan 10 kali dengan percobaan pertama mengganti interval menjadi 2 detik, percobaan kedua menjadi 3 detik, dan seterusnya.

```

Console - 50219
2333 46 Clear Autoscroll Update Log

server: "time":1687124778299, "key": "0e02khwvgj", "id": "AAAc", "T":29.76, "rh":69.36, "Pa":92.13, "a":[-0.15, -0.07, 0.89]
=====
server: "time":1687124780299, "key": "0e02khwvgj", "id": "AAAc", "T":29.76, "rh":69.49, "Pa":92.13, "a":[-0.15, -0.06, 0.9]
=====
server: "time":1687124782299, "key": "0e02khwvgj", "id": "AAAc", "T":29.76, "rh":69.55, "Pa":92.13, "a":[-0.15, -0.07, 0.89]
=====

Radio receive: setInterval:3000
server: "time":1687124784299, "key": "0e02khwvgj", "id": "AAAc", "T":29.76, "rh":69.58, "Pa":92.0, "a":[-0.15, -0.07, 0.9]
=====

server: "time":1687124787299, "key": "0e02khwvgj", "id": "AAAc", "T":29.83, "rh":69.58, "Pa":92.25, "a":[-0.15, -0.07, 0.89]
=====

server: "time":1687124790299, "key": "0e02khwvgj", "id": "AAAc", "T":29.76, "rh":69.61, "Pa":92.06, "a":[-0.15, -0.07, 0.89]
=====

server: "time":1687124793299, "key": "0e02khwvgj", "id": "AAAc", "T":29.76, "rh":69.61, "Pa":92.25, "a":[-0.15, -0.07, 0.9]
=====

LF

```

Gambar 5.10: Tangkapan layar *Log* pada *Node Sensor* diPercobaan Kedua Eksperimen Ganti Interval

Tabel 5.18: Percobaan *Node Sensor* Ganti Interval

Pengujian	Hasil
1	berhasil
2	berhasil
3	berhasil
4	berhasil
5	berhasil
6	berhasil
7	berhasil
8	berhasil
9	berhasil
10	berhasil

Hasil: Berdasarkan tangkapan layar yang dapat dilihat pada gambar 5.10 dapat dilihat *node sensor* berhasil mengubah interval menjadi 3 detik, ditandai dengan yang sebelumnya interval 2000 menjadi interval 3000. Kemudian dari hasil percobaan pada tabel 5.18 yang menyatakan berhasil secara keseluruhan, oleh karena itu percobaan kali ini dinyatakan berhasil

- Mencoba menganti interval negatif kepada *node sensor*. Harapannya yaitu *node sensor* menolak dengan tidak menganti intervalnya.

```

Console - 50219
13878 288 Clear  Autoscroll  Update Log

server: "time": 1687124499292, "key": "0e02khwvgj", "id": "AAAc", "T": 29.33, "rh": 70.97, "Pa": 92.19, "a": [-0.16, -0.07, 0.89]
=====
server: "time": 1687124500292, "key": "0e02khwvgj", "id": "AAAc", "T": 29.33, "rh": 71.04, "Pa": 92.25, "a": [-0.16, -0.07, 0.89]
=====

Radio receive: setInterval:-6000
Interval not updated because lower than 1ms
server: "time": 1687124501292, "key": "0e02khwvgj", "id": "AAAc", "T": 29.33, "rh": 71.04, "Pa": 92.06, "a": [-0.16, -0.07, 0.9]
=====

server: "time": 1687124502292, "key": "0e02khwvgj", "id": "AAAc", "T": 29.33, "rh": 71.04, "Pa": 92.19, "a": [-0.16, -0.07, 0.89]
=====

server: "time": 1687124503292, "key": "0e02khwvgj", "id": "AAAc", "T": 29.33, "rh": 71.01, "Pa": 92.25, "a": [-0.16, -0.07, 0.89]
=====

server: "time": 1687124503292, "key": "0e02khwvgj", "id": "AAAc", "T": 29.33, "rh": 71.01, "Pa": 92.25, "a": [-0.16, -0.07, 0.89]
=====

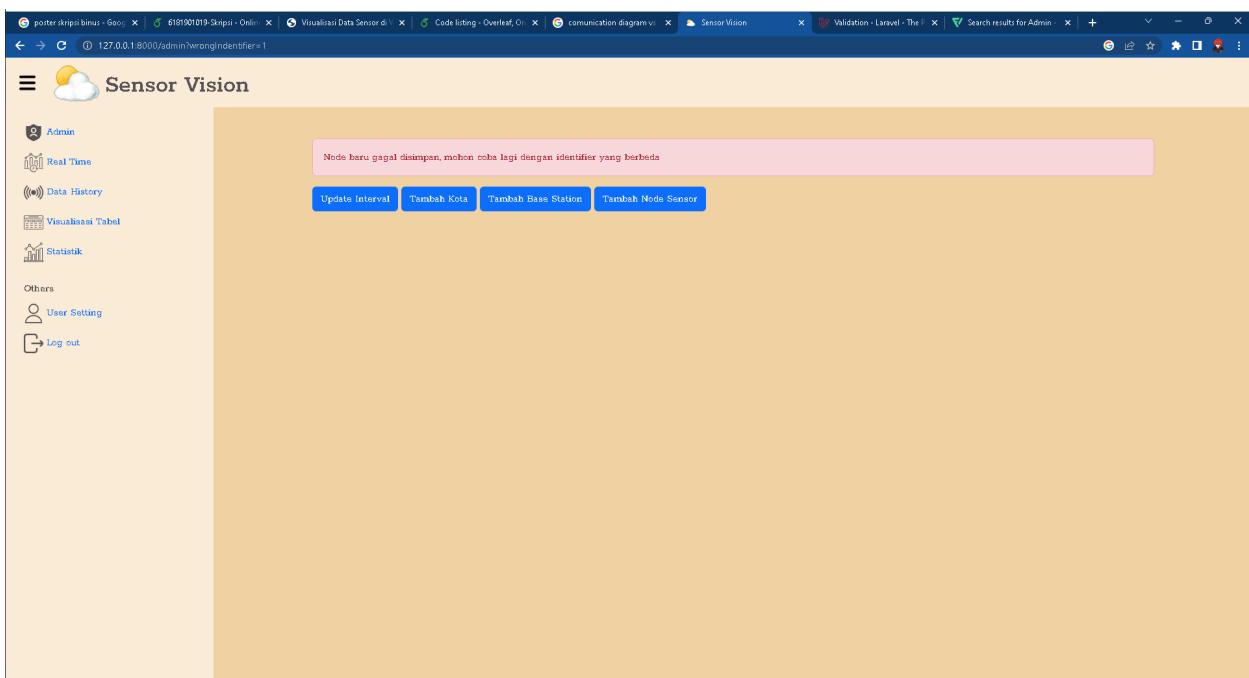
LF

```

Gambar 5.11: Tangkapan Layar *Log* pada *Node Sensor* Eksperimen Memberikan Interval Negatif

Hasil: Berdasarkan tangkapan layar yang dapat dilihat pada gambar 5.11 dapat dilihat *node sensor* berhasil menolak interval negatif yaitu dapat dilihat kalau data keempat dan kelima masih selisih 1000 milidetik atau 1 detik, oleh karena itu percobaan kali ini dinyatakan berhasil

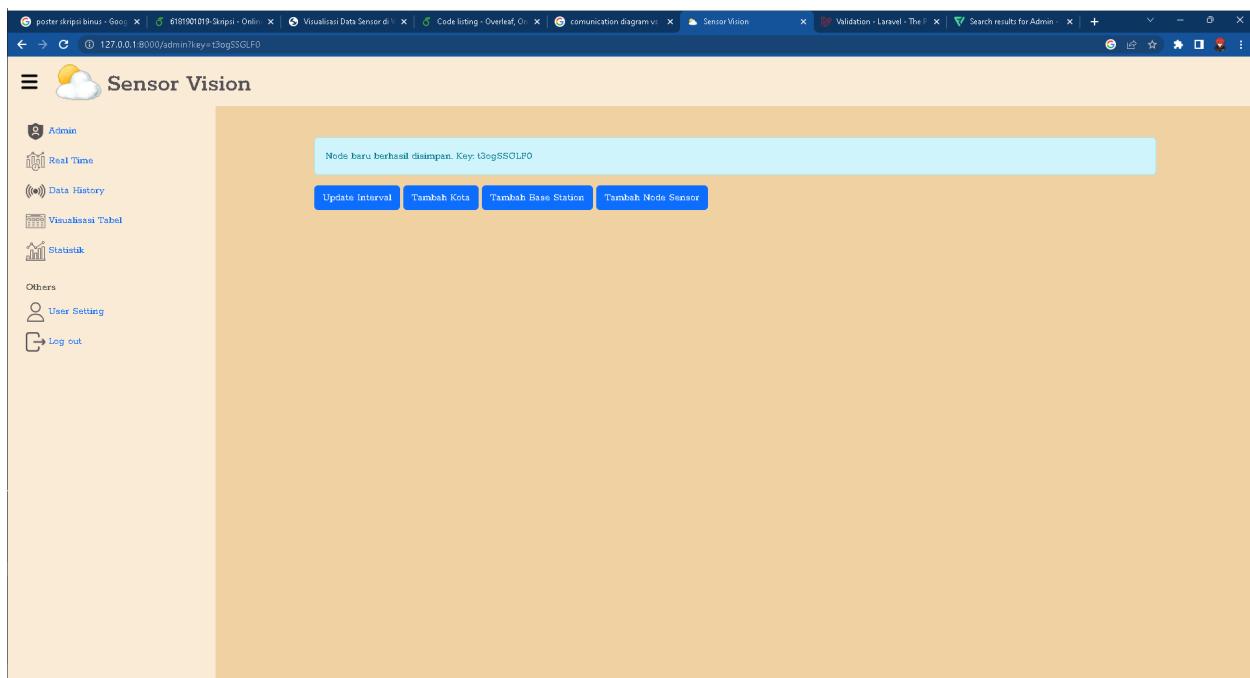
- Mencoba memberikan identifier yang sudah terdaftar pada saat pendaftaran *node baru*. Harapannya yaitu *node* yang baru tidak berhasil tersimpan dan admin diberikan notifikasi.



Gambar 5.12: Tangkapan Layar Website Admin Memberikan Notifikasi Gagal Input Node

Hasil: Berdasarkan tangkapan layar yang dapat dilihat pada gambar 5.12 dapat dilihat website berhasil menampilkan pesan *error*, oleh karena itu percobaan ini dinyatakan berhasil.

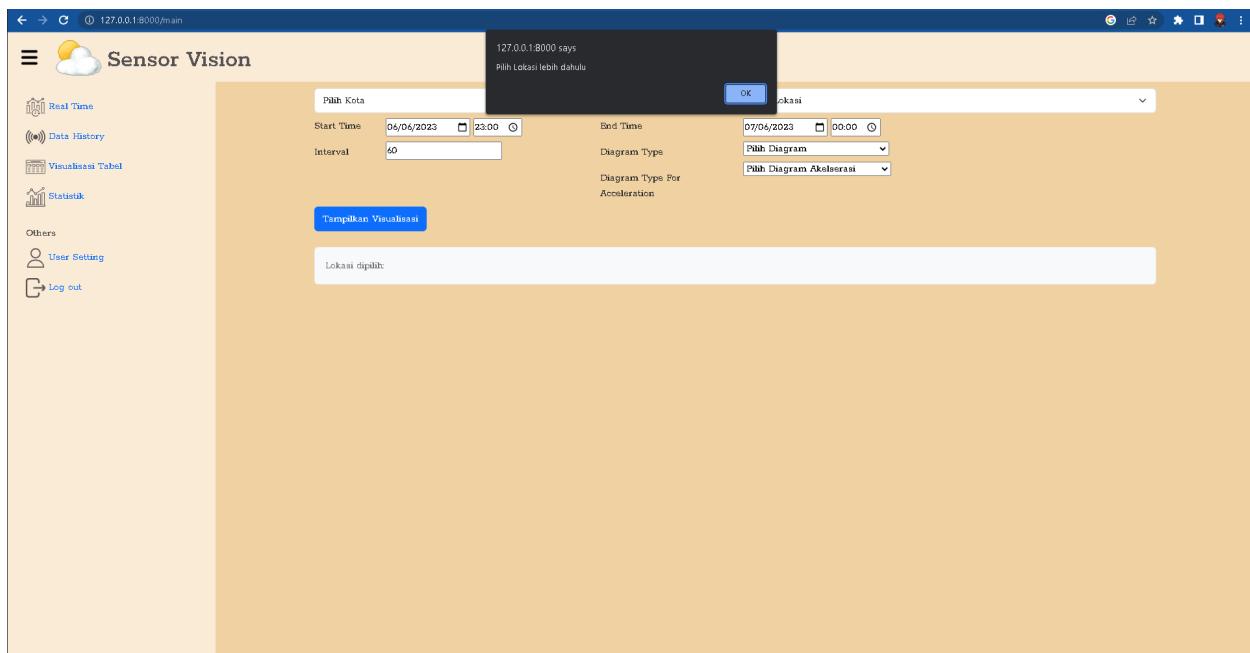
- Mencoba mendaftarkan *node baru*. Harapannya yaitu *node* yang baru dimasukan tersimpan dan admin diberikan notifikasi megenai key nodenya.



Gambar 5.13: Tangkapan Layar Website Admin Memberikan Notifikasi *Key Node* yang Baru disimpan

Hasil: Berdasarkan tangkapan layar yang dapat dilihat pada gambar 5.13 dapat dilihat website berhasil menampilkan pesan berisi *key*, oleh karena itu percobaan ini dinyatakan berhasil.

- Mencoba menampilkan visualisasi data tanpa mengisi seluruh parameter. Harapannya yaitu website memberitahu pengguna terdapat isian yang belum diisi.



Gambar 5.14: Tangkapan Layar Website Menampilkan Pesan belum Memilih Lokasi



Gambar 5.15: Tangkapan Layar Website Menampilkan Pesan Belum Memilih Tanggal Awal



Gambar 5.16: Tangkapan Layar Website Menampilkan pesan belum memilih tanggal Awal



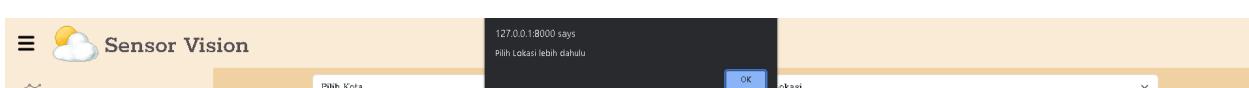
Gambar 5.17: Tangkapan Layar Website Menampilkan Pesan Belum Memilih Waktu Awal



Gambar 5.18: Tangkapan Layar Website Menampilkan Pesan Belum Memilih Tanggal Akhir



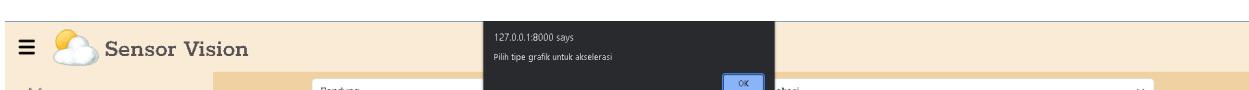
Gambar 5.19: Tangkapan Layar Website Menampilkan Pesan Belum Memilih Waktu Akhir



Gambar 5.20: Tangkapan Layar Website Menampilkan Pesan Belum Mengisi Interval



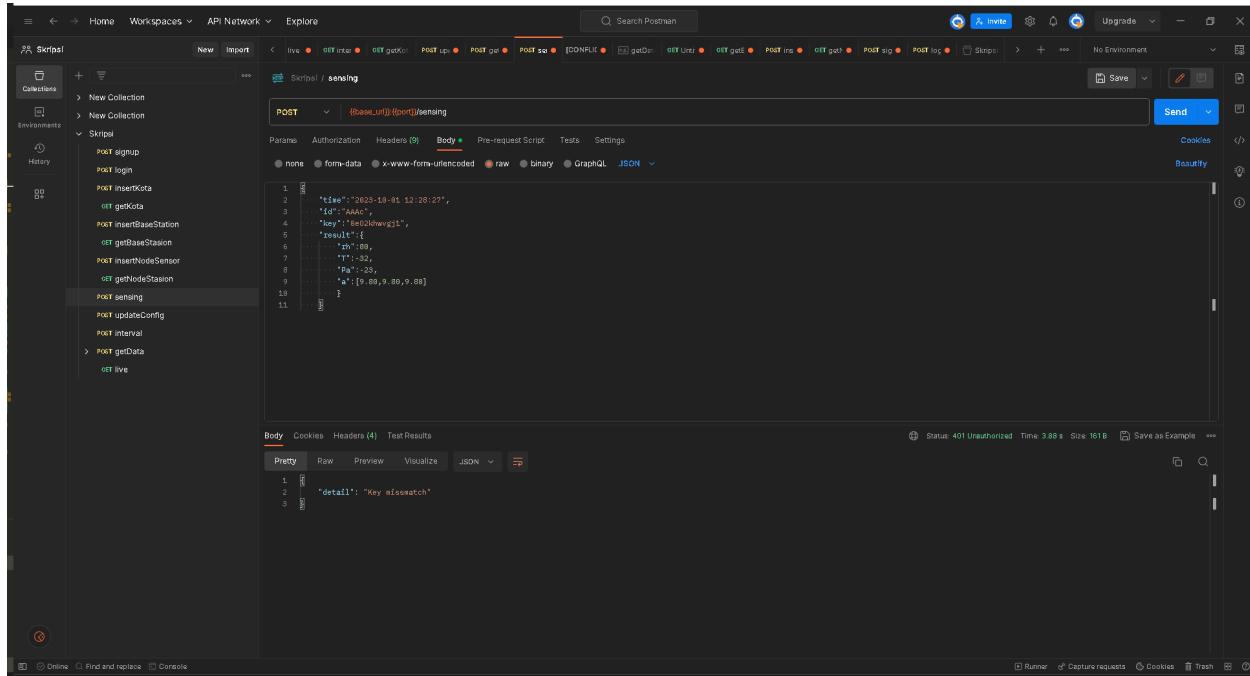
Gambar 5.21: Tangkapan Layar Website Menampilkan Pesan Belum Memilih Tipe Visualisasi



Gambar 5.22: Tangkapan Layar Website Menampilkan Pesan Belum Memilih Tipe Visualisasi untuk Akselerasi

Hasil: Berdasarkan tangkapan layar yang dapat dilihat pada gambar 5.14 sampai dengan gambar 5.22 dapat dilihat website berhasil menampilkan pesan, oleh karena itu percobaan ini dinyatakan berhasil.

- Mencoba mengirimkan data *sensing* tetapi menggunakan *key* yang tidak sesuai dengan *node*nya. Harapannya data akan ditolak dan diberikan balasan berupa autentikasi tidak sesuai.



Gambar 5.23: Tangkapan Layar Mencoba untuk Menginput Data *Sensing* dengan Key Tidak Cocok

Hasil: API menolak data *sensing* dan diberikan balasan berupa informasi tentang key yang salah bersama juga kode HTTP 401, Oleh karena itu percobaan ini dinyatakan berhasil.

5.2.2 Pengujian Eksperimental

Pada pengujian Eksperimental akan dilakukan 3 buah pengujian. Berikut merupakan pengujian eksperimental yang dilakukan:

- Mencoba menonaktifkan *base station* kemudian mengaktifkan kembali. Harapan yaitu *node* sensor dapat tersambung kembali dan melanjutkan mengirimkan data. Pengujian dilakukan 10 kali percobaan. Hasil untuk percobaan dapat dilihat pada tabel 5.19

The screenshot shows a terminal window titled "Console - 53956". At the top, there are buttons for "12002", "395", "Clear", "Autoscroll" (unchecked), "Update" (checked), and "Log". The terminal displays a series of log entries from a server, each starting with "server:" followed by timestamp, ID, and various sensor values (T, rh, Pa, a). The log entries are separated by double asterisks (***) and double equals signs (====).

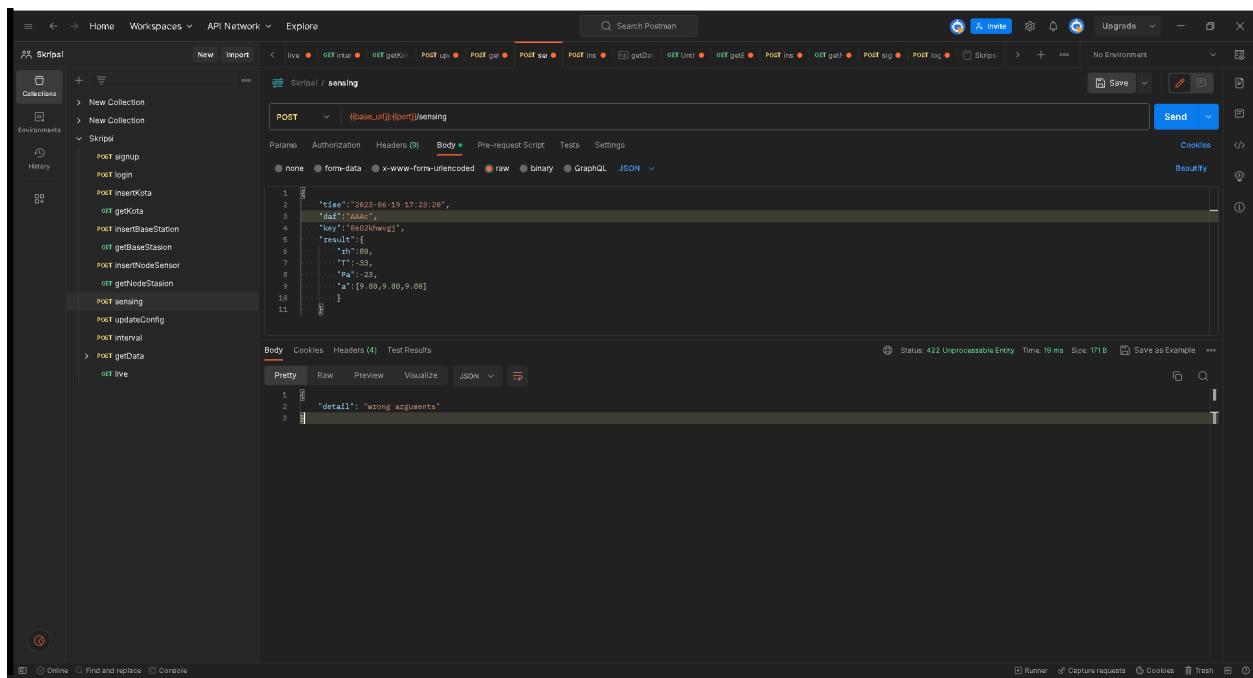
```
server:"time":1686782309329,"idBS": "AAAc", "T":29.58, "rh":67.71, "Pa":92.25, "a": [0.07, 0.02, 0.9]
=====
failed send msg to 0
autostart 43
    autostart 770 75
    autostart 520 6
    autostart 984 13
    autostart 503 72
    autostart 3 151
    autostart 4 41
    autostart 1 12
=====
=====
server:"time":1686782310329,"idBS": "AAAc", "T":29.58, "rh":67.74, "Pa":92.0, "a": [0.08, 0.02, 0.91]
=====
failed send msg to 0
autostart 43
    autostart 770 75
    autostart 520 6
    autostart 984 13
    autostart 503 72
    autostart 3 151
    autostart 4 41
    autostart 1 12
=====
=====
server:"time":1686782311329,"idBS": "AAAc", "T":29.58, "rh":67.87, "Pa":92.19, "a": [0.07, 0.02, 0.91]
=====
=====
server:"time":1686782312329,"idBS": "AAAc", "T":29.52, "rh":68.12, "Pa":92.25, "a": [0.07, 0.02, 0.91]
=====
=====
server:"time":1686782313329,"idBS": "AAAc", "T":29.58, "rh":68.12, "Pa":92.25, "a": [0.07, 0.02, 0.91]
=====
=====
server:"time":1686782314329,"idBS": "AAAc", "T":29.52, "rh":67.96, "Pa":92.19, "a": [0.07, 0.02, 0.91]
=====
=====
server:"time":1686782315330,"idBS": "AAAc", "T":29.58, "rh":68.03, "Pa":92.13, "a": [0.07, -0.02, 0.9]
```

Gambar 5.24: Tangkapan Layar Percobaan ke-3 Node Sensor Melakukan Sensing Kembali

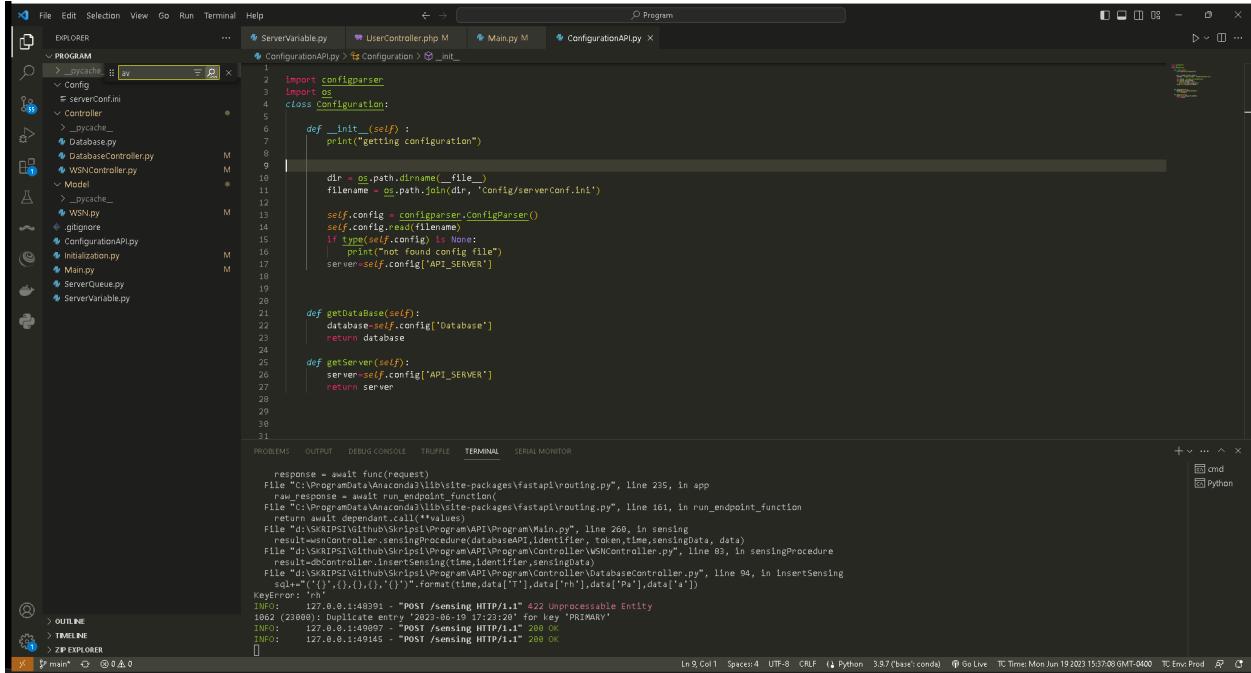
Tabel 5.19: Percobaan *Node* Sensor Tersambung Kembali

Pengujian	Hasil
1	berhasil
2	berhasil
3	berhasil
4	berhasil
5	berhasil
6	berhasil
7	berhasil
8	berhasil
9	berhasil
10	berhasil

- Mencoba mengirimkan pesan yang salah kepada API. Harapan yaitu untuk melihat apakah API dapat menangani masukan yang bermasalah.



Gambar 5.25: Tangkapan Layar Mencoba Memasukan Masukan yang Salah



```

File Edit Selection View Go Run Terminal Help
EXPLORER
PROGRAM
    > __pycache__ av
    > Config
    > Controller
    > __pycache__
        Database.py
        DatabaseController.py
    Model
        WSNController.py
    WSN.py
    gitignore
    ConfigurationAPI.py
    Initialization.py
    Main.py
    ServerQueue.py
    ServerVariable.py

ConfigurationAPI.py Configuration > __init__.py
1 import configparser
2
3 import os
4
5 class Configuration:
6
7     def __init__(self):
8         print("getting configuration")
9
10
11     dir = os.path.dirname(__file__)
12     filename = os.path.join(dir, "Config/serverConf.ini")
13
14     self.config = configparser.ConfigParser()
15     self.config.read(filename)
16
17     if type(self.config) is None:
18         print("not found config file")
19     server=self.config["API_SERVER"]
20
21
22     def getDatabase(self):
23         database=self.config['Database']
24         return database
25
26     def getServer(self):
27         server=self.config['API_SERVER']
28         return server
29
30
31
PROBLEMS OUTPUT DEBUG CONSOLE TRUFFLE TERMINAL SERIAL MONITOR
RESPONSE = await func.future()
File "C:\ProgramData\anaconda3\lib\site-packages\fastapi\routing.py", line 235, in app
    raw_response = await run_endpoint_function
File "C:\ProgramData\anaconda3\lib\site-packages\fastapi\routing.py", line 161, in run_endpoint_function
    return await dependant.call(**values)
File "C:\ProgramData\anaconda3\lib\site-packages\fastapi\constraints.py", line 269, in sensing
    result=WSNController.sensingProcedure(database=WSNIdentifier, token_time=sensingData, data)
File "d:\daskr1ps1\github\Shrishti\ProgramAPI\ProgramAPI\program\controller\WSNController.py", line 83, in sensingProcedure
    result=WSNController.insertSensing(time,Identifier,sensingData)
File "d:\daskr1ps1\github\Shrishti\ProgramAPI\ProgramAPI\program\controller\WSNController.py", line 94, in insertSensing
    sensingData['t']=str(datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S"))
    sensingData['t']=str(datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")) .join(data['T'],data['Pa'],data['A'])
KeyError: 't'
INFO: 127.0.0.1:49891 - "POST /sensing HTTP/1.1" 422 Unprocessable Entity
1082: (2300): Duplicate entry '2023-06-19 17:23:20' for key 'PRIMARY'
INFO: 127.0.0.1:49997 - "POST /sensing HTTP/1.1" 200 OK
INFO: 127.0.0.1:49145 - "POST /sensing HTTP/1.1" 200 OK

```

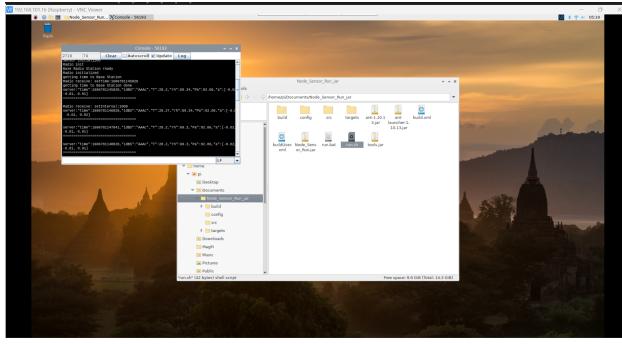
L in 9, Col 0 Spaces: 4 UTF-8 CR/LF (Python 3.8.7(base:conda)) Go Live TC Time: Mon Jun 12 2023 15:37:03 GMT+0400 TC Env Prod

Gambar 5.26: Tangkapan Layar API Crash

Hasil: Berdasarkan tangkapan layar yang dapat dilihat pada gambar 5.25. API dapat menangani masukan yang bermasalah. Namun pada gambar 5.26 dapat dilihat API pernah mengalami *crash*, tetapi API tetap dapat melanjutkan fungsinya pada masukan-masukan berikutnya. Oleh karena itu percobaan ini dinyatakan berhasil.

Hasil: Berdasarkan tangkapan layar yang dapat dilihat pada gambar 5.24 dan tabel 5.19. *Node* sensor berhasil melanjutkan mengirimkan data, oleh karena itu percobaan ini dinyatakan berhasil.

- Mencoba mengaktifkan *node* sensor dari Raspberry Pi. Para percobaan ini akan digunakan Raspberry Pi 3 Model B+ Rev 1.3 yang memiliki RAM 1gb dengan sistem operasi Linux Raspberry 5.10.103-v7+. Percobaan ini bertujuan untuk melihat apakah perangkat lunak dan perangkat keras Preon32 yang dibuat mampu berjalan pada sistem operasi yang berbeda dan arsitektur prosesor yang berbeda.



Gambar 5.27: Raspberry Mengaktifkan Preon32



```

C:\WINDOWS\system32> + v
FROM USART source:2, "time": "2023-06-15 05:11:09", "idBS": "AAAc", "T": 31.26, "rh": 62.79, "Pa": 92.25, "a": [-0.01, -0.07, 0.91]
TO USART 2:0k
FROM USART source:2, "time": "2023-06-15 05:11:09", "idBS": "AAAc", "result": [{"T": 31.26, "rh": 62.79, "Pa": 92.25, "a": [-0.01, -0.07, 0.91]}]
TO API {"time": "2023-06-15 05:11:09", "result": [{"T": 31.26, "rh": 62.79, "Pa": 92.25, "a": [-0.01, -0.07, 0.91]}]}
FROM USART source:2, "time": "2023-06-15 05:11:09", "idBS": "AAAc", "T": 31.26, "rh": 62.78, "Pa": 92.25, "a": [-0.02, -0.07, 0.91]
TO USART 2:0k
FROM USART source:2, "time": "2023-06-15 05:11:09", "idBS": "AAAc", "result": [{"T": 31.26, "rh": 62.78, "Pa": 92.25, "a": [-0.02, -0.07, 0.91]}]
TO API {"time": "2023-06-15 05:11:09", "result": [{"T": 31.26, "rh": 62.78, "Pa": 92.25, "a": [-0.02, -0.07, 0.91]}]}
FROM USART source:2, "time": "2023-06-15 05:11:09", "idBS": "AAAc", "T": 28.2, "rh": 69.34, "Pa": 92.06, "a": [-0.02, -0.01, 0.91]
TO USART 2:0k
FROM USART source:2, "time": "2023-06-15 05:11:09", "idBS": "AAAc", "result": [{"T": 28.2, "rh": 69.34, "Pa": 92.06, "a": [-0.02, -0.01, 0.91]}]
FROM USART source:2, "time": "2023-06-15 05:11:09", "idBS": "AAAc", "T": 28.2, "rh": 69.34, "Pa": 92.06, "a": [-0.02, -0.01, 0.91]
TO USART 2:0k
FROM USART source:2, "time": "2023-06-15 05:11:09", "idBS": "AAAc", "result": [{"T": 28.2, "rh": 69.34, "Pa": 92.06, "a": [-0.02, -0.01, 0.91]}]
FROM USART source:2, "time": "2023-06-15 05:19:07", "idBS": "AAAc", "T": 28.2, "rh": 69.34, "Pa": 92.06, "a": [-0.02, -0.01, 0.91]
FROM USART source:2, "time": "2023-06-15 05:19:07", "idBS": "AAAc", "result": [{"T": 28.2, "rh": 69.34, "Pa": 92.06, "a": [-0.02, -0.01, 0.91]}]
FROM USART source:2, "time": "2023-06-15 05:19:07", "idBS": "AAAc", "T": 28.2, "rh": 69.34, "Pa": 92.06, "a": [-0.02, -0.01, 0.91]
TO USART 2:0k
FROM USART source:2, "time": "2023-06-15 05:19:07", "idBS": "AAAc", "result": [{"T": 28.2, "rh": 69.34, "Pa": 92.06, "a": [-0.02, -0.01, 0.91]}]
FROM USART source:2, "time": "2023-06-15 05:19:07", "idBS": "AAAc", "T": 28.2, "rh": 69.34, "Pa": 92.06, "a": [-0.02, -0.01, 0.91]
TO USART 2:0k
FROM USART source:2, "time": "2023-06-15 05:19:07", "idBS": "AAAc", "result": [{"T": 28.2, "rh": 69.34, "Pa": 92.06, "a": [-0.02, -0.01, 0.91]}]
FROM USART source:2, "time": "2023-06-15 05:19:07", "idBS": "AAAc", "T": 28.2, "rh": 69.27, "Pa": 92.06, "a": [-0.02, -0.01, 0.91]
TO USART 2:0k
FROM USART source:2, "time": "2023-06-15 05:19:07", "idBS": "AAAc", "result": [{"T": 28.2, "rh": 69.27, "Pa": 92.06, "a": [-0.02, -0.01, 0.91]}]

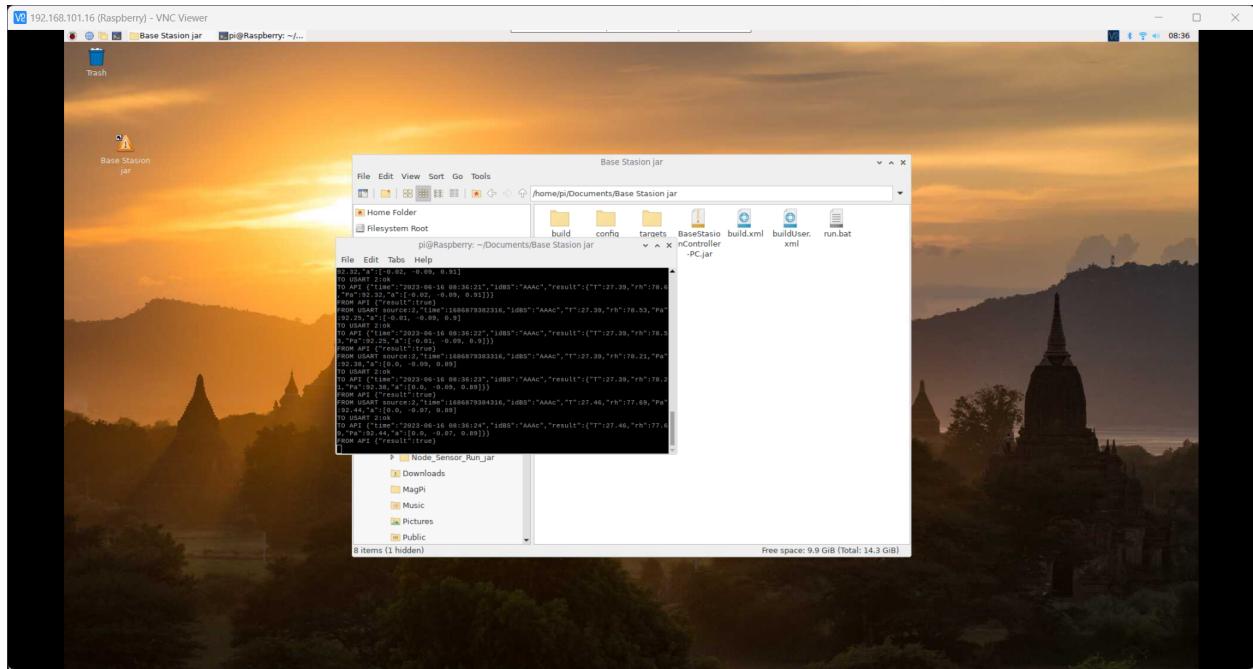
```

Gambar 5.28: Data Diterima DiBase Station

Hasil: Berdasarkan tangkapan layar yang dapat dilihat pada gambar 5.22. Dapat dilihat bahwa *node* sensor berhasil melakukan *sensing* dan diterima oleh *base station*, oleh karena itu percobaan ini dinyatakan berhasil.

- Mencoba menjalankan *base station* dari Raspberry Pi. Para percobaan ini akan digunakan Raspberry Pi 3 Model B+ Rev 1.3 yang memiliki RAM 1gb dengan sistem operasi Linux

Raspberry 5.10.103-v7+.



Gambar 5.29: Raspberry Mengaktifkan Preon32 yang Bertindak Sebagai *Base Station*

Hasil: Berdasarkan tangkapan layar yang dapat dilihat pada gambar 5.29. Dapat dilihat bahwa *base stasion* sensor berhasil menerima *sensing* dari *node*, oleh karena itu percobaan ini dinyatakan berhasil.

5.2.3 Kesimpulan Hasil Eksperimen

Berdasarkan eksperimen fungsional yang telah dilakukan, penulis mengambil kesimpulan bahwa perangkat lunak dapat menampilkan visualisasi data yang tepat sesuai dengan kebutuhan pengguna dan ketersediaan data yang ada. Kemudian berdasarkan eksperimen eksperimental yang dilakukan dapat disimpulkan bahwa perangkat keras dapat melanjutkan kembali setelah *base station* kembali aktif serta perangkat keras dapat berfungsi di sistem operasi lain.

BAB 6

KESIMPULAN DAN SARAN

Pada bab ini membahas mengenai kesimpulan berdasarkan pengujian, serta saran untuk pengembangan berikutnya,

6.1 Kesimpulan

Berdasarkan hasil pengujian yang dilakukan diperoleh kesimpulan yaitu:

1. Aplikasi Visualisasi data berhasil menampilkan visualisasi data sesuai dengan kebutuhan yang dianalisis pada bagian 3.2.
2. Berdasarkan dari pengujian baik pengujian eksperimental maupun pengujian fungsional untuk bagian WSN, bagian *server*, dan bagian tampilan dinilai sudah cukup untuk memenuhi kebutuhan dan dapat berfungsi didunia nyata.

6.2 Saran

Berdasarkan pengamatan, pengujian terdapat beberapa saran yaitu:

1. Menambahkan jumlah sensor yang digunakan agar dapat dibuatkan visualisasi dengan jenis *map* dan juga tetap memenuhi kaidah WSN.
2. Pada API dapat dibuatkan model untuk kota dan *base station* agar dapat disimpan pada sebuah variabel. Sehingga tidak perlu mengambil ke basis data setiap kali ada permintaan.
3. API hanya dapat menampilkan data dari waktu mulai hingga waktu selesai dibulan yang sama sehingga saran untuk berikutnya bisa diperbaiki.

DAFTAR REFERENSI

- [1] Kirk, A. (2012) *Data Visualization: a successful design process*, 2st edition. Packt Publishing, Birmingham.
- [2] Unggana, D. (2019) Pengembangan aplikasi pemantauan wireless sensor network. Skripsi. Universitas Katolik Parahyangan, Indonesia.
- [3] 1.0 (2015) *Preon32 - Wireless Module*. Virtenio GmbH. Berlin, Jerman.
- [4] Kurniawan, F. K. (2022) Analisis visualisasi data covid-19 berdasarkan kaidah visualisasi data. Skripsi. Universitas Katolik Parahyangan, Indonesia.
- [5] Wei, X. dan Zhang, X. (2018) Web browser based data visualization scheme for xbee wireless sensor network. *Transactions On Networks and CommunicationsS*, **6**.
- [6] Predic, B., Yan, Z., Eberle, J., Stojanovic, D., dan Aberer, K. (2013) Exposuresense: Integrating daily activities with air quality using mobile participatory sensing. *2013 IEEE International Conference on Pervasive Computing and Communications Workshops PerCom Workshops 2013*, San Diego, Amerika Serikat, 03, pp. 303–305. IEEE, San Diego.
- [7] Lopez, J. C. B. dan Villaruz, H. (2015) Low-cost weather monitoring system with online logging and data visualization. *2014 IEEE International Conference on Advanced Networks and Telecommunication Systems*, Filipina, 12, pp. 1–6. IEEE, Filipina.
- [8] Sarangi, S., Bisht, A., Rao, V., dan Kar, S. K. (2014) Development of a wireless sensor network for animal management: Experiences with moosense. *2014 IEEE International Conference on Advanced Networks and Telecommunication Systems*, India, 12, pp. 1–6. IEEE, India.
- [9] Astuti, L. D. (2017) Extending wireless sensor network lifetime using clustered shortest geopath routing protocol (c-sgp). Thesis. Institut Teknologi Sepuluh Nopember, Indonesia.
- [10] McGrath, M. J. dan Scanaill, C. N. (2013) *Sensor Technologies: Healthcare, Wellness, and Environmental Applications*, 1st edition. Apress Media, New York.
- [11] Sutrisno, C. (2020) Pengembangan aplikasi agregasi data akselerometer di wsn. Skripsi. Universitas Katolik Parahyangan, Indonesia.
- [12] Farahani, S. (2008) *ZigBee Wireless Networks and Transceivers*, 1st edition. Newnes, Burlington.
- [13] Mihajlov, B. dan Bogdanoski, M. (2011) Overview and analysis of the performances of zigbee based wireless sensor networks. *International Journal of Computer Applications*, **29**.
- [14] Dargie, W. dan Poellabauer, C. (2010) *Fundamentals of Wireless Sensor Networks: Theory and Practice*, 1st edition. A John Wiley and Sons, Ltd., Chichester.
- [15] 1.0 (2015) *Sensor board for the expansion board Preon32 Shuttle*. Virtenio GmbH. Berlin, Jerman.
- [16] Gourley, D. dan Totty, B. (2002) *HTTP The Definitive Guide*, 1st edition. OReilly, California.

- [17] Horstmann, C. S. (2013) *Big Java: Late Objects*, 2nd edition. John Wiley & Sons, Inc, New Jersey.
- [18] Flanagan, D. (2005) *Java in a Nutshell*, 5st edition. OReilly, California.
- [19] Chun, W. J. (2006) *Core Python Programming*, 2nd edition. Pearson Education, Inc., New Jersey.
- [20] Eriksson, M. dan HallBerg, V. (2011) Comparison between json and yaml for data serialization. Skripsi. KTH Royal Institute of Technology, Swedia.

LAMPIRAN A

KODE PROGRAM NODE SENSOR PADA PREON32

Kode A.1: Node Sensor pada Preon32 - NodeSensor.java

```
1 package com.dimas519;
2
3 import com.dimas519.Radio.*;
4 import com.dimas519.Sensor.*;
5 import com.virtenio.driver.i2c.I2C;
6 import com.virtenio.driver.i2c.I2CEception;
7 import com.virtenio.driver.i2c.NativeI2C;
8 import com.virtenio.vm.Time;
9
10
11 public class NodeSensor implements MainInterface {
12     private final String identifier; //ini public identifier, wajib unique
13     private final int COMMON_CHANNEL; //channel node ini
14     private final int COMMON_PANID; //pan id node ini
15     private final int myAddress; // alamat node ini
16     private final int myBSAddress; // alamat bs node ini
17     private long interval; //selang waktu sensing dilakukan
18     private final Sensor[] sensors; //array objek sensor yang terdapat pada sensor init
19     private final MyRadio myRadio; //variabel objek radio, radio ini digunakan untuk berkomunikasi dengan bs
20     private boolean timeSync=false; //variabel yang digunakan agar sensor menunggu waktu dari bs terlebih dahulu
21     private String token; //variabel yang digunakan sebagai authentikasi
22
23     /**
24      *
25      * Sebuah method yang akan dijalankan pada saat pertama kali node sensor aktif
26      *
27      */
28     public static void main(String[] args) {
29         NodeSensor node=new NodeSensor();
30         node.run();
31     }
32
33
34     /**
35      *
36      * Sebuah method yang digunakan untuk inisialisasi variabel-variabel pada kelas ini yang sudah ditentukan
37      *
38      */
39     public NodeSensor() {
40
41         System.out.println("Variable_INITIALIZATION");
42         this.identifier="AAAC"; //ini public identifier, wajib unique
43
44         this.COMMON_CHANNEL =24;
45         this.COMMON_PANID =0xCAFE;
46         this.myAddress =0x0002;
47         this.myBSAddress=0x0000;
48         //this.interval=300000;
49         this.interval=1000;
50         //this.interval=200;
51         this.sensors=new Sensor[4];
52         this.token="0e02khwvgj";
53
54         System.out.println("Variable_INITIALIZED");
55
56
57
58         System.out.println("I2C(Init)");
59         NativeI2C i2c= NativeI2C.getInstance(1);
60         try {
61             i2c.open(I2C.DATA_RATE_400);
62         } catch (I2CEception e) {
63             System.out.println("failed_i2c");
64         }
65
66         System.out.println("Sensors_init");
67         this.sensors[0]=new Thermometer(i2c);
68         this.sensors[1]=new Hygrometer(i2c);
69         this.sensors[2]=new Barometer(i2c);
70         this.sensors[3]=new Accelerometer();
71         System.out.println("Sensor_INITIALIZED");
72
73         System.out.println("Radio_init");
74         this.myRadio=new MyRadio(this ,myAddress,COMMON_PANID,COMMON_CHANNEL);
```

```

76    System.out.println("Radio_initialized");
77    System.out.flush();
78 }
79
80 /**
81 *
82 * Sebuah method yang bertugas untuk melakukan sensing setiap selang waktu tertentu.
83 * setelah melakukan sensing methods ini juga bertugas untuk mengirimkan data sensing ke bs
84 *
85 */
86
87
88 private void sensing(){
89     String sensing;
90     long start, end,processTime;
91     while (true){
92         start=Time.currentTimeMillis();
93         sensing = "server:\"time\":" + Time.currentTimeMillis() + ",\"token\":"+this.token+"\",\"node\":" + this.
94         identifier + "\",";
95         for (Sensor sensor : this.sensors) {
96             sensing += ",";
97             sensing += sensor.run();
98         }
99         System.out.println(sensing);
100        System.out.flush();
101        this.myRadio.sendMSG(this.myBSAddress, sensing);
102
103        System.out.println("=====");
104        System.out.println(">");
105
106        System.out.flush();
107        end=Time.currentTimeMillis();
108        processTime=end-start;
109        Misc.sleep(this.interval-processTime);
110    }
111
112 /**
113 *
114 * Sebuah method yang bertugas untuk melakukan meminta nilai-nilai lainnya yang dibutuhkan seperti waktu, interval, dll dari
115 * bs
116 * Selain hal tersebut ini juga bertugas untuk memanggil sensing jika variabel sudah diinisialisasi
117 */
118 private void run(){
119     this.myRadio.receive();
120     System.out.println("getting_time_to_Base_Station");
121     initializeTime();
122     while(!timeSync){}
123     System.out.println("getting_time_to_Base_Station_done");
124     sensing();
125 }
126
127 /**
128 *
129 * Sebuah method digunakan untuk mengirimkan informasi kepada bs kalau node ini meminta interval dan waktu
130 *
131 */
132
133 private void initializeTime(){
134     this.myRadio.sendMSG(this.myBSAddress,"bs:intervalRequest,\"node\":"+this.identifier+\"");
135 }
136
137 /**
138 *
139 * Sebuah method yang digunakan untuk memproses informasi dari radio yang dikirimkan oleh base stasion
140 *
141 * @param address sebuah parameter siapa address pengirimnya. untuk setTime dan setInterval hanya boleh dilakukan oleh address
142 * bnsya
143 *
144 * @param msg sebuah parameter array dengan index 0 berupa commandnya dan index 1 berupa valuenya
145 * misalkan ["setInterval","5000"] maka commandnya untuk mengganti interval dan nilai interval barunya 5000ms
146 *
147 */
148 @Override
149 public void processMsg(long address, String[] msg) {
150     if(address==this.myBSAddress) { //memastikan pengirimannya berasal dari base stasion
151         if (msg[0].equals("setTime")) {
152             Time.setCurrentTimeMillis(Long.parseLong(msg[1]));
153             this.timeSync = true;
154
155         } else if (msg[0].equals("setInterval")) {
156             int inputInterval=Integer.parseInt(msg[1]);
157             if(inputInterval>=1000){
158                 this.interval = Integer.parseInt(msg[1]);
159             }else{
160                 System.out.println("Interval_not_updated_because_lower_than_1ms");
161             }
162
163         }
164     }
165 }
166 }

```

Kode A.2: Node Sensor pada Preon32 - Sensor.java

```

1 package com.dimas519.Sensor;
2
3 import com.virtenio.driver.i2c.NativeI2C;
4
5 abstract public class Sensor {
6     public abstract String run();
7     abstract void init(NativeI2C i2c) throws Exception;
8 }

```

Kode A.3: Node Sensor pada Preon32 - Accelerometer.java

```

/*
 * Copyright (c) 2011., Virtenio GmbH
 * All rights reserved.
 *
 * Commercial software license.
 * Only for test and evaluation purposes.
 * Use in commercial products prohibited.
 * No distribution without permission by Virtenio.
 * Ask Virtenio for other type of license at info@virtenio.de
 *
 * Kommerzielle Softwarelizenz.
 * Nur zum Test und Evaluierung zu verwenden.
 * Der Einsatz in kommerziellen Produkten ist verboten.
 * Ein Vertrieb oder eine Veröffentlichung in jeglicher Form ist nicht ohne Zustimmung von Virtenio erlaubt.
 * Fr andere Formen der Lizenz nehmen Sie bitte Kontakt mit info@virtenio.de auf.
 */
package com.dimas519.Sensor;
import java.util.Arrays;
import com.dimas519.Radio.Misc;
import com.virtenio.driver.device.ADXL345;
import com.virtenio.driver.gpio.GPIO;
import com.virtenio.driver.gpio.NativeGPIO;
import com.virtenio.driver.i2c.NativeI2C;
import com.virtenio.driver.spi.NativeSPI;
/***
 *
 * Kelas Akselerometer merupakan kelas yang bertugas untuk menginisialisasi akselerometer(alat ukur akselerasi) dan melakukan pembacaan akselerasi.
 */
public class Accelerometer extends Sensor{
    private ADXL345 accelerationSensor;
    private String name="a"; //nama disingkat karena max frame 802.15.4 itu 127 char
    public Accelerometer(){
        try {
            init(null); //cukup diset null karena tidak pakai i2c
        } catch (Exception e) {
            System.out.println(name+" failed");
        }
    }
    @Override
    void init(NativeI2C i2c) throws Exception {
        System.out.println("GPIO(Init)");
        GPIO accelCs = NativeGPIO.getInstance(20);
        System.out.println("SPI(Init)");
        NativeSPI spi = NativeSPI.getInstance(0);
        spi.open(ADXL345.SPI_MODE, ADXL345.SPI_BIT_ORDER, ADXL345.SPI_MAX_SPEED); //disediakan dengan parameter yang dibutuhkan (mode,bit order dan max speed)
        System.out.println("ADXL345(Init)");
        accelerationSensor = new ADXL345(spi, accelCs); //init sensor accelerasi
        accelerationSensor.open();
        accelerationSensor.setDataFormat(ADXL345.DATA_FORMAT_RANGE_2G); //diset max pembacaan sensor 2g saja
        accelerationSensor.setDataRate(ADXL345.DATA_RATE_50HZ); //rate data 50hz, 50 data dalam 1 detik, pembacaan akan dilakukan dengan interval-min 1 detik sehingga dicari rate yang kecil agar battery lebih hemat daya
        accelerationSensor.setPowerControl(ADXL345.POWER_CONTROL_MEASURE);
        System.out.println("Done(Init)");
    }
    @Override
    public String run() {
        try {
            short[] values = new short[3]; //variabel yang akan menyimpan nilai xyz
            accelerationSensor.getValuesRaw(values, 0); //membaca xyz sensor berdasarkan offset (parameter kedua)
            //perlu dibagi dengan typ (256), angka diperoleh dari dokumentasi sensor adxl345.
            //typ merupakan angka mayoritas, jika lokasi suatu sumbu yang mengarah ke inti dibagi dengan 256 maka hasilnya mendekati 1g
            double[] inG=new double[3];
            inG[0]= Misc.round( (values[0]/256.0)); //set agar dia desimal dengan presisi 2
            inG[1]= Misc.round( (values[1]/256.0));
            inG[2]= Misc.round( (values[2]/256.0));
        }

```

```

84
85     String result="\\"+name+":\\"+Arrays.toString(inG); //mengubah array menjadi array
86     return result;
87 } catch (Exception e) {
88     System.err.println(name+" failed");
89     String result=name+":null";
90     return result;
91 }
92 }
93
94 }
95 }
```

Kode A.4: Node Sensor pada Preon32 - Thermometer.java

```

1 /*
2 * Copyright (c) 2011., Virtenio GmbH
3 * All rights reserved.
4 *
5 * Commercial software license.
6 * Only for test and evaluation purposes.
7 * Use in commercial products prohibited.
8 * No distribution without permission by Virtenio.
9 * Ask Virtenio for other type of license at info@virtenio.de
10 *
11 * Kommerzielle Softwarelizenz.
12 * Nur zum Test und Evaluierung zu verwenden.
13 * Der Einsatz in kommerziellen Produkten ist verboten.
14 * Ein Vertrieb oder eine Veröffentlichung in jeglicher Form ist nicht ohne Zustimmung von Virtenio erlaubt.
15 * Für andere Formen der Lizenz nehmen Sie bitte Kontakt mit info@virtenio.de auf.
16 */
17
18 package com.dimas519.Sensor;
19
20 import com.dimas519.Radio.Misc;
21 import com.virtenio.driver.device.ADT7410;
22 import com.virtenio.driver.i2c.NativeI2C;
23
24 /**
25 *
26 * Kelas Thermometer merupakan kelas yang bertugas untuk menginisialisasi thermometer(alat ukur suhu) dan melakukan pembacaan
27 * tekanan udara dengan satuan celcius.
28 */
29 public class Thermometer extends Sensor{
30     private ADT7410 temperatureSensor;
31     private String name="T"; //nama disingkat karena max frame 802.15.4 itu 127 char
32
33     public Thermometer(NativeI2C i2c){
34         try {
35             init(i2c);
36         } catch (Exception e) {
37             System.err.println("temperature failed initialize"+e);
38         }
39     }
40
41
42     @Override
43     void init(NativeI2C i2c) throws Exception {
44         System.out.println("ADT7410(Init)");
45         temperatureSensor = new ADT7410(i2c, ADT7410.ADDR_0, null, null);
46         temperatureSensor.open();
47         temperatureSensor.setMode(ADT7410.CONFIG_MODE_CONTINUOUS);
48
49         System.out.println("Done(Init)");
50     }
51
52
53     @Override
54     public String run() {
55         try {
56             float celsius = temperatureSensor.getTemperatureCelsius();
57
58             String result="\\"+name+":\\"+ Misc.round(celsius);
59             return result;
60         } catch (Exception e) {
61             System.out.println(name+" failed"+e.getMessage());
62             String result=name+":null";
63             return result;
64         }
65     }
66
67 }
68
69 }
70 }
```

Kode A.5: Node Sensor pada Preon32 - Hygrometer.java

```

1 /*
2 * Copyright (c) 2011., Virtenio GmbH
3 * All rights reserved.
4 *
5 * Commercial software license.
6 * Only for test and evaluation purposes.
67 }
```

```

7 * Use in commercial products prohibited.
8 * No distribution without permission by Virtenio.
9 * Ask Virtenio for other type of license at info@virtenio.de
10 *
11 * Kommerzielle Softwarelizenz.
12 * Nur zum Test und Evaluierung zu verwenden.
13 * Der Einsatz in kommerziellen Produkten ist verboten.
14 * Ein Vertrieb oder eine Veröffentlichung in jeglicher Form ist nicht ohne Zustimmung von Virtenio erlaubt.
15 * Fr andere Formen der Lizenz nehmen Sie bitte Kontakt mit info@virtenio.de auf.
16 */
17
18 package com.dimas519.Sensor;
19
20 import com.dimas519.Radio.Misc;
21 import com.virtenio.driver.device.SHT21;
22 import com.virtenio.driver.i2c.NativeI2C;
23
24 /**
25 *
26 * Kelas Barometer merupakan kelas yang bertugas untuk menginisialisasi barometer(alat ukur tekanan) dan melakukan pembacaan tekanan udara dengan satuan kPa.
27 *
28 */
29 public class Hygrometer extends Sensor{
30     private SHT21 humiditySensor;
31     private String name="rh"; //nama disingkat karena max frame 802.15.4 itu 127 char
32
33     public Hygrometer(NativeI2C i2c){
34         try {
35             init(i2c);
36         } catch (Exception e) {
37             System.err.println("Humidity_failed_initialize");
38         }
39     }
40
41     @Override
42     void init(NativeI2C i2c) throws Exception {
43
44         System.out.println("SHT21(Init)");
45         humiditySensor = new SHT21(i2c);
46         humiditySensor.open();
47         humiditySensor.setResolution(SHT21.RESOLUTION_RH12_T14);
48         humiditySensor.reset();
49
50         System.out.println("Done(Init)");
51     }
52
53
54     @Override
55     public String run() {
56         try {
57             humiditySensor.startRelativeHumidityConversion();
58             Thread.sleep(1000);
59             int rawRH = humiditySensor.getRelativeHumidityRaw();
60             float rh = SHT21.convertRawRHToRHw(rawRH);
61
62             String result="\\""+name+":\""+ Misc.round(rh);
63             return result;
64         } catch (Exception e) {
65             System.err.println(name+_failed"+e.getMessage());
66             String result=name+":null";
67             return result;
68         }
69     }
70 }
71
72 }

```

Kode A.6: Node Sensor pada Preon32 - Barometer.java

```

1 /*
2 * Copyright (c) 2011., Virtenio GmbH
3 * All rights reserved.
4 *
5 * Commercial software license.
6 * Only for test and evaluation purposes.
7 * Use in commercial products prohibited.
8 * No distribution without permission by Virtenio.
9 * Ask Virtenio for other type of license at info@virtenio.de
10 *
11 * Kommerzielle Softwarelizenz.
12 * Nur zum Test und Evaluierung zu verwenden.
13 * Der Einsatz in kommerziellen Produkten ist verboten.
14 * Ein Vertrieb oder eine Veröffentlichung in jeglicher Form ist nicht ohne Zustimmung von Virtenio erlaubt.
15 * Fr andere Formen der Lizenz nehmen Sie bitte Kontakt mit info@virtenio.de auf.
16 */
17
18 package com.dimas519.Sensor;
19
20 import com.dimas519.Radio.Misc;
21 import com.virtenio.driver.device.MPL115A2;
22 import com.virtenio.driver.gpio.GPIO;
23 import com.virtenio.driver.gpio.NativeGPIO;
24 import com.virtenio.driver.i2c.NativeI2C;
25
26 /**
27 *

```

```

28 * Kelas Barometer merupakan kelas yang bertugas untuk menginisialisasi barometer(alat ukur kelembapan) dan melakukan pembacaan
29 * tekanan udara dengan satuan kPa.
30 */
31 public class Barometer extends Sensor{
32     private MPL115A2 pressureSensor;
33     private String name="Pa"; //nama disingkat karena max frame 802.15.4 itu 127 char
34
35
36     public Barometer(NativeI2C i2c){
37         try {
38             init(i2c);
39         } catch (Exception e) {
40             System.err.println("Barometer_Failed_initialize");
41         }
42     }
43     @Override
44     void init(NativeI2C i2c) throws Exception {
45
46         System.out.println("GPIO(Init)");
47         GPIO resetPin = NativeGPIO.getInstance(24);
48         GPIO shutDownPin = NativeGPIO.getInstance(12);
49
50         System.out.println("MPL115A2(Init)");
51         pressureSensor = new MPL115A2(i2c, resetPin, shutDownPin);
52         pressureSensor.open();
53         pressureSensor.setReset(false);
54         pressureSensor.setShutdown(false);
55
56         System.out.println("Done(Init)");
57     }
58
59
60     @Override
61     public String run() {
62         try {
63             pressureSensor.startBothConversion();
64             Thread.sleep(MPL115A2.BOTH_CONVERSION_TIME);
65             int pressurePr = pressureSensor.getPressureRaw();
66             int tempRaw = pressureSensor.getTemperatureRaw();
67             float pressure = pressureSensor.compensate(pressurePr, tempRaw);
68
69             String result="\\"+name+" ":"+Misc.round(pressure);
70             return result;
71         } catch (Exception e) {
72             System.err.println(name+"_failed");
73             String result=name+":null";
74             return result;
75         }
76     }
77 }
78 }
```

Kode A.7: Node Sensor pada Preon32 - MyRadio.java

```

1 /*
2  * Copyright (c) 2011., Virtenio GmbH
3  * All rights reserved.
4  *
5  * Commercial software license.
6  * Only for test and evaluation purposes.
7  * Use in commercial products prohibited.
8  * No distribution without permission by Virtenio.
9  * Ask Virtenio for other type of license at info@virtenio.de
10 *
11 * Kommerzielle Softwarelizenzen.
12 * Nur zum Test und Evaluierung zu verwenden.
13 * Der Einsatz in kommerziellen Produkten ist verboten.
14 * Ein Vertrieb oder eine Verffentlichtung in jeglicher Form ist nicht ohne Zustimmung von Virtenio erlaubt.
15 * Fr andere Formen der Lizenz nehmen Sie bitte Kontakt mit info@virtenio.de auf.
16 */
17
18 package com.dimas519.Radio;
19
20
21
22 import com.virtenio.driver.device.at86rf231.AT86RF231;
23 import com.virtenio.driver.device.at86rf231.AT86RF231RadioDriver;
24 import com.virtenio.radio.ieee_802_15_4.Frame;
25 import com.virtenio.radio.ieee_802_15_4.FrameIO;
26 import com.virtenio.radio.ieee_802_15_4.RadioDriver;
27 import com.virtenio.radio.ieee_802_15_4.RadioDriverFrameIO;
28
29
30 /**
31  * Einfaches Beispiel der Funkbertragung mit Senden und Empfangen.
32  *
33  * Sebuah kelas objek radio yang berfungsi untuk menerima dan mengirimkan pesan ke bs
34  */
35 public class MyRadio {
36
37
38     private FrameIO frameIO;
39     private AT86RF231 radio;
40     private final MainInterface mainInterface;
41
42
43 }
```

```

44    /**
45     * method ini bertujuan untuk menginisialisasi radio agar siap digunakan dalam bertukar pesan dengan bs
46     *
47     * @param mainInterface sebuah interface yang digunakan untuk berinteraksi dengan kelas utama node sensor(NodeSensor)
48     *
49     * @param resv alamat node ini
50     *
51     * @param panID personal area networks node ini
52     *
53     * @param channel channel node ini
54     *
55     */
56
57 public MyRadio(MainInterface mainInterface, int resv, int panID, int channel){
58
59     this.mainInterface=mainInterface;
60
61     //inisialisasi radio
62     try {
63         this.radio=RadioInit.initRadio();
64         this.radio.setChannel(channel);
65         this.radio.setPANId(panID);
66         this.radio.setShortAddress(resv);
67         RadioDriver driver = new AT86RF231RadioDriver(this.radio);
68         this.frameIO=new RadioDriverFrameIO(driver);
69         System.out.println("Base_Radio_Station_ready");
70     } catch (Exception e) {
71         System.out.println("failed_radio");
72     }
73
74
75 }
76
77
78 /**
79  *
80  * Sebuah method yang berfungsi untuk mengirimkan pesan ke tujuan
81  *
82  * @param sendAddress alamat tujuan pesan
83  *
84  * @param msg isi pesannya
85  *
86  */
87 public void sendMSG(int sendAddress, String msg) {
88     Exception tempEx;
89     for (int i = 0; i < 9; i++) {
90         try {
91             // /////////////////////////////////
92             Frame frame = new Frame(Frame.TYPE_DATA | Frame.ACK_REQUEST
93             | Frame.DST_ADDR_16 | Frame.INTRA_PAN | Frame.SRC_ADDR_16);
94             frame.setSrcAddr(this.radio.getShortAddress());
95             frame.setSrcPanId(this.radio.getPANId());
96             frame.setDestAddr(sendAddress);
97             frame.setDestPanId(this.radio.getPANId());// karena untuk berkomunikasi dia butuh dalam satu pan(personal
98             // network)
99             frame.setPayload(msg.getBytes());
100            this.frameIO.transmit(frame); //kirimkan framenya
101            break;
102        } catch (Exception e) {
103            tempEx=e;
104        }
105
106        if(i==8){ //kalau sampai 8 kali gagal kemungkinan alamat tujuannya tidak aktif, diluar jangkauan atau salah.
107            System.out.println("*****");
108            System.out.println("_failed_send_msg_to_" + sendAddress );
109            tempEx.printStackTrace();
110            System.out.println("*****");
111        }
112    }
113
114 }
115
116
117 /**
118  * Ein Programme, dass ber das Startmenu aufgerufen werden kann
119  *
120  * sebuah method yang bertugas untuk menemani pesan yang ditujukan untuk node ini di thread miliknya sendiri.
121  * kemudian akan memproses pesan tersebut/
122  *
123  *
124  */
125
126 public void receive() {
127     Thread receive = new Thread() {
128         @Override
129         public void run() {
130             while (true) {
131                 Frame f = null;
132                 try {
133                     f = new Frame();
134                     frameIO.receive(f);
135                 } catch (Exception e) {
136
137                 if (f != null) {
138                     byte[] dg = f.getPayload();
139                     String msg = new String(dg, 0, dg.length);
140                     long sourceAddress = f.getSrcAddr(); //pengirimnya
141

```

```

142         System.out.println("Radio_receive:"+msg);
143         String[] splittedMSG=splitMSG(msg,':'); //memisahkan perintah dan nilainya
144         mainInterface.processMsg(sourceAddress,splittedMSG); //memproses perintah
145     }
146 }
147 }
148 }
149 }
150 receive.start();
151 }
152 }
153 }
154 /**
155 *
156 * sebuah method yang berfungsi untuk memisahkan perintah dan nilainya
157 *
158 * @param msg pesan yang belum dipisahkan
159 *
160 * @param regex karakter pemisah pesan dan nilai
161 *
162 * @return array string berukuran 2 dengan index 0 berupa perintah dan index 1 berupa nilainya
163 */
164 private String[] splitMSG(String msg,char regex){
165     for (int i=0;i<msg.length();i++){
166         if(msg.charAt(i)==regex){
167             return new String[]{
168                 msg.substring(0,i),
169                 msg.substring(i+1)
170             };
171         }
172     }
173     return null;
174 }
175 }
176 }
177 }
178 }

```

Kode A.8: Node Sensor pada Preon32 - Misc.java

```

1 /*
2  * Copyright (c) 2011., Virtenio GmbH
3  * All rights reserved.
4  *
5  * Commercial software license.
6  * Only for test and evaluation purposes.
7  * Use in commercial products prohibited.
8  * No distribution without permission by Virtenio.
9  * Ask Virtenio for other type of license at info@virtenio.de
10 *
11 * Kommerzielle Softwarelizenzen.
12 * Nur zum Test und Evaluierung zu verwenden.
13 * Der Einsatz in kommerziellen Produkten ist verboten.
14 * Ein Vertrieb oder eine Verffentlichtung in jeglicher Form ist nicht ohne Zustimmung von Virtenio erlaubt.
15 * Fr andere Formen der Lizenz nehmen Sie bitte Kontakt mit info@virtenio.de auf.
16 */
17
18 package com.dimas519.Radio;
19
20 import com.virtenio.driver.led.LED;
21
22 /**
23 * Zusätzliche Hilfsfunktionen zu Programmen fr den Sensorknoten und die
24 * Integration ins Menu.
25 */
26
27 public class Misc {
28
29 /**
30 * Aktueller Thread wartet eine bestimmte Zeit.
31 * sebuah method yang bertugas untuk melakukan sleep pada suatu thread.
32 *
33 * @param millis
34 *           Dauer in Millisekunden, die der aktuelle Thread anhalten soll.
35 *           waktu dalam milidetik sebuah tread akan sleep
36 *
37 */
38 public static void sleep(long millis) {
39     if (millis > 0) {
40         try {
41             Thread.sleep(millis);
42         } catch (InterruptedException e) {
43         }
44     }
45 }
46
47 /**
48 * Sebuah method yang berfungsi untuk memastikan desimal dengan presisi 2
49 *
50 * @param value nilai yang akan diubah presisinya
51 * @return nilai yang sudah dengan 2 presisi
52 */
53
54 public static double round(double value) {
55     return Math.round( value * 100.0 ) / 100.0;
56 }
57 }
58

```

59 }

Kode A.9: Node Sensor pada Preon32 - MainInterface.java

```

1 package com.dimas519.Radio;
2
3 /**
4  * sebuah interface yang menghubungkan thread radio receive dengan kelas utama (NodeSensor)
5  */
6 public interface MainInterface {
7
8     /**
9      * penjelasan terdapat pada kelas utama (NodeSensor)
10     */
11    void processMsg(long address, String[] msg);
12 }

```

Kode A.10: Node Sensor pada Preon32 - RadioInit.java

```

1 /*
2  * Copyright (c) 2011., Virtenio GmbH
3  * All rights reserved.
4  *
5  * Commercial software license.
6  * Only for test and evaluation purposes.
7  * Use in commercial products prohibited.
8  * No distribution without permission by Virtenio.
9  * Ask Virtenio for other type of license at info@virtenio.de
10 *
11 * Kommerzielle Softwarelizenz.
12 * Nur zum Test und Evaluierung zu verwenden.
13 * Der Einsatz in kommerziellen Produkten ist verboten.
14 * Ein Vertrieb oder eine Veröffentlichung in jeglicher Form ist nicht ohne Zustimmung von Virtenio erlaubt.
15 * Fr andere Formen der Lizenz nehmen Sie bitte Kontakt mit info@virtenio.de auf.
16 */
17
18 package com.dimas519.Radio;
19
20 import com.virtenio.driver.device.at86rf231.AT86RF231;
21 import com.virtenio.misc.PropertyHelper;
22 import com.virtenio.preon32.node.Node;
23 import com.virtenio.radio.ieee_802.15.4.Frame;
24
25 /** Beispiele einer Initialisierung der Funkverbindung. */
26 public class RadioInit {
27
28     /**
29      * <pre>
30      * {@literal
31      *   Value      TX Output Power [dBm]
32      *   0x0        3.0
33      *   0x1        2.8
34      *   0x2        2.3
35      *   0x3        1.8
36      *   0x4        1.3
37      *   0x5        0.7
38      *   0x6        0.0
39      *   0x7        -1
40      *   0x8        -2
41      *   0x9        -3
42      *   0xA        -4
43      *   0xB        -5
44      *   0xC        -7
45      *   0xD        -9
46      *   0xE        -12
47      *   0xF        -17
48      * }
49      * </pre>
50     */
51     private final int TRANSMIT_POWER = PropertyHelper.getInt("radio.transmit_power", 0, 0, 15);
52
53     /**
54      * A CCA measurement is used to detect a clear channel. Four modes are
55      * specified by IEEE 802.15.4 - 2006:
56      * <p/>
57      * mode 1: Energy above threshold. CCA shall report a busy medium upon
58      * detecting any energy above the ED threshold. mode 2: Carrier sense only.
59      * CCA shall report a busy medium only upon the detection of a signal with
60      * the modulation and spreading characteristics of an IEEE 802.15.4
61      * compliant signal. The signal strength may be above or below the ED
62      * threshold. mode 0,3: Carrier sense with energy above threshold. CCA shall
63      * report a busy medium using a logical combination of
64      * <li>Detection of a signal with the modulation and spreading
65      * characteristics of this standard and</li>
66      * <li>Energy above the ED threshold.</li> Where the logical operator may be
67      * configured as either OR (mode 0) or AND (mode 3).
68      */
69     private final int CCA_MODE = PropertyHelper.getInt("radio.cca_mode", 3, 0, 3);
70     private final int CCA_THRESHOLD = PropertyHelper.getInt("radio.cca_threshold", 7, 0, 15);
71
72     /**
73      * Number frame retries
74      */
75     private final int FRAME_RETRIES = PropertyHelper.getInt("radio.frame_retries", 1, 0, 15);
76

```

```

77 /**
78 * CSMA related values
79 */
80 private final int CSMA_RETIRES = PropertyHelper.getInt("radio.csma_retries", 1, 0, 7);
81 private final int CSMA_SEED = PropertyHelper.getInt("radio.csma_seed", 511, 0, 1023);
82
83 /**
84 * Backoff exponent
85 */
86 private final int MIN_BACKOFF_EXPONENT = PropertyHelper.getInt("radio.min_backoff_exponent", 3,
87     0, 15);
88 private final int MAX_BACKOFF_EXPONENT = PropertyHelper.getInt("radio.max_backoff_exponent", 5,
89     0, 15);
90
91 /**
92 * Address related parts
93 */
94 private final int PAN_ID = PropertyHelper.getInt("radio.pan_id", 0xCAFE, 0, 0xFFFF);
95 private final int SHORT_ADDR = PropertyHelper.getInt("radio.short_address", 0, 0, 0xFFFF);
96 private final long LONG_ADDR = PropertyHelper.getLong("radio.long_address", 0);
97 private final boolean I_AM_COORD = PropertyHelper.getBool("radio.i_am_coodord", false);
98
99 /**
100 * Radio channel
101 */
102 private final int CHANNEL = PropertyHelper.getInt("radio.channel", AT86RF231.CHANNEL_MIN,
103     AT86RF231.CHANNEL_MIN, AT86RF231.CHANNEL_MAX);
104
105 /**
106 *
107 * @param channel
108 * @param panID
109 * @param addr
110 * @return
111 */
112 public static AT86RF231 initRadio() throws Exception {
113     RadioInit ri = new RadioInit();
114     AT86RF231 radio = Node.getInstance().getTransceiver();
115     radio.open();
116     radio.reset();
117     radio.setState(AT86RF231.STATE_TRX_OFF);
118     radio.setRxSaveMode(true);
119     radio.setMaxFrameRetries(ri.FRAME_RETIRES);
120     radio.setMaxCSMAretries(ri.CSMA_RETIRES);
121     radio.setCSMAseed(ri.CSMA_SEED);
122     radio.setBackoffExponent(ri.MIN_BACKOFF_EXPONENT, ri.MAX_BACKOFF_EXPONENT);
123     radio.setChannel(ri.CHANNEL);
124     radio.setTransmitPower(ri.TRANSMIT_POWER);
125     radio.setCCAMode(ri.CCA_MODE, ri.CCA_THRESHOLD);
126     radio.setAddressFilter(ri.PAN_ID, ri.SHORT_ADDR, ri.LONG_ADDR, ri.I_AM_COORD);
127     radio.setPendingData(false);
128     radio.setState(AT86RF231.STATE_RX_AACK_ON);
129     return radio;
130 }
131 }
```

LAMPIRAN B

KODE PROGRAM BASE STATION PADA PREON32

Kode B.1: Base Station pada Preon32 - BaseStationMain.java

```
1 package com.dimas519;
2
3
4 import com.dimas519.Radio.MyRadio;
5 import com.dimas519.USART.MyUsart;
6 import com.virtenio.vm.Time;
7
8 public class BaseStationMain implements MainInterface {
9
10    private final MyUsart myUsart; //objek usart, untuk berkomunikasi dengan laptop/ pc
11    private final MyRadio myRadio; //objek radio, untuk berkomunikasi dengan ns
12
13
14    //konfigurasi radio
15    private final int COMMON_CHANNEL =24;
16    private final int COMMON_PANID =0xCAFE;
17    private final int myAddress =0X0000;
18
19
20    /**
21     * Method yang pertama akan dijalankan
22     * @param args
23     */
24    public static void main(String[] args) {
25
26        try {
27            new BaseStationMain().run();
28        } catch (Exception e) {
29            System.out.println(e.getMessage());
30        }
31    }
32
33
34    public BaseStationMain(){
35        this.myUsart=new MyUsart(this);
36        this.myRadio=new MyRadio(this, myAddress, COMMON_PANID, COMMON_CHANNEL);
37    }
38
39
40    public void run() throws Exception {
41        this.myRadio.receive();
42        this.myUsart.send("start");
43        this.myUsart.run();
44
45    }
46
47
48    @Override
49    public void processMsg(long address, String[] msg) {
50
51        if(msg[0]!=null && msg[1]!= null) {
52            if (msg[0].equals("bs")) { // kalau prefix nya bs, artinya request sesuatu ke base stasion
53                this.setTime(address); //pada saat ini feature hanya setTime dan request interval saja
54                this.myUsart.send("source:" + address + "," + msg[1]); //untuk minta interval
55            } else if (msg[0].equals("server") ) { // kaalau prefix nya bs artinya request sesuatu ke base stasion
56                this.myUsart.send("source:" + address + "," + msg[1]);
57            } else {
58                this.myRadio.send(address,msg[0]+":"+msg[1]);
59            }
60        }
61    }
62
63
64
65
66    private void setTime(long address) {
67        String msg="setTime:"+ Time.currentTimeMillis();
68        this.myRadio.send(address,msg);//mengirimkan kembali perintah setTime dan valuenya
69    }
70
71
72 }
```

Kode B.2: Base Station pada Preon32 - MainInterface.java

```

1 package com.dimas519;
2
3 /**
4  * Interface yang bertujuan agar dapat memproses pesan dikelas BaseStationMain
5 */
6 public interface MainInterface {
7     /**
8      *
9      * Method yang bergfungsi untuk memisahkan-misahkan pesan dari atau ke usart dan radio
10     *
11     * @param address alamat node yang mengirimkan/ tujuan
12     * @param msg //isi pesan
13     */
14     void processMsg(long address, String[] msg);
15 }
16 }
```

Kode B.3: Base Station pada Preon32 - MyRadio.java

```

1 /*
2  * Copyright (c) 2011., Virtenio GmbH
3  * All rights reserved.
4  *
5  * Commercial software license.
6  * Only for test and evaluation purposes.
7  * Use in commercial products prohibited.
8  * No distribution without permission by Virtenio.
9  * Ask Virtenio for other type of license at info@virtenio.de
10 *
11 * Kommerzielle Softwarelizenzen.
12 * Nur zum Test und Evaluierung zu verwenden.
13 * Der Einsatz in kommerziellen Produkten ist verboten.
14 * Ein Vertrieb oder eine Verffentlichung in jeglicher Form ist nicht ohne Zustimmung von Virtenio erlaubt.
15 * Fr andere Formen der Lizenz nehmen Sie bitte Kontakt mit info@virtenio.de auf.
16 */
17
18 package com.dimas519.Radio;
19
20
21 import com.dimas519.MainInterface;
22 import com.virtenio.driver.device.at86rf231.AT86RF231;
23
24 import com.virtenio.driver.device.at86rf231.AT86RF231RadioDriver;
25 import com.virtenio.radio.ieee_802_15_4.Frame;
26 import com.virtenio.radio.ieee_802_15_4.FrameIO;
27 import com.virtenio.radio.ieee_802_15_4.RadioDriver;
28 import com.virtenio.radio.ieee_802_15_4.RadioDriverFrameIO;
29
30
31
32 /**
33  * Einfaches Beispiel der Funkbertragung mit Senden und Empfangen.
34  *
35  *  * Sebuah kelas objek radio yang berfungsi untuk menerima dan mengirimkan pesan ke node
36  */
37 public class MyRadio {
38
39
40     private FrameIO frameIO;
41     private AT86RF231 radio;
42     private final MainInterface mainInterface;
43
44
45     /**
46      *
47      * method ini bertujuan untuk menginisialisasi radio agar siap digunakan dalam bertukar pesan dengan node-node
48      *
49      * @param mainInterface sebuah interface yang digunakan untuk berinteraksi dengan kelas utama base stasion(BaseStationMain)
50      *
51      * @param resv alamat base ini
52      *
53      * @param panID personal area networks base ini
54      *
55      * @param channel channel base ini
56      *
57      */
58
59     public MyRadio(MainInterface mainInterface, int resv, int panID, int channel){
60
61         this.mainInterface=mainInterface;
62
63         //inisialisasi radio
64         try {
65             this.radio=RadioInit.initRadio();
66             this.radio.setChannel(channel);
67             this.radio.setPANId(panID);
68             this.radio.setShortAddress(resv);
69             RadioDriver driver = new AT86RF231RadioDriver(this.radio);
70             this.frameIO=new RadioDriverFrameIO(driver);
71         } catch (Exception e) {
72             System.out.println("failed_radio");
73         }
74
75
76     }
77 }
```

```

78|
79| /**
80| *
81| * Sebuah method yang berfungsi untuk mengirimkan pesan ke tujuan
82| *
83| * @param sendAddress alamat tujuan pesan
84| *
85| * @param msg isi pesannya
86| *
87| */
88| public void send(long sendAddress, String msg) {
89|     boolean isOK = false;
90|     while (!isOK) {
91|         try {
92|             // /////////////////////////////////
93|             Frame frame = new Frame(Frame.TYPE_DATA | Frame.ACK_REQUEST
94|                                     | Frame.DST_ADDR_16 | Frame.INTRA_PAN | Frame.SRC_ADDR_16);
95|             frame.setSrcAddr(this.radio.getShortAddress());
96|             frame.setSrcPanId(this.radio.getPANId());
97|             frame.setDestAddr(sendAddress);
98|             frame.setDestPanId(this.radio.getPANId()); // karena untuk berkomunikasi dia butuh dalam satu pan(personal network)
99|             frame.setPayload(msg.getBytes());
100|             this.frameIO.transmit(frame);
101|             isOK = true;
102|         } catch (Exception e) {
103|             System.out.println("failed_send_msg_to"+sendAddress);
104|         }
105|     }
106| }
107|
108|
109|
110|
111|
112| /**
113| * Ein Programme, dass bei das Startmenu aufgerufen werden kann
114| *
115| * sebuah method yang bertugas untuk menenerima pesan yang ditujukan untuk base ini di thread miliknya sendiri.
116| * kemudian akan memproses pesan tersebut/
117| *
118| *
119| */
120|
121| public void receive() throws Exception {
122|     Thread receive = new Thread() {
123|         @Override
124|         public void run() {
125|             while (true) {
126|                 Frame f = null;
127|                 try {
128|                     f = new Frame();
129|                     frameIO.receive(f);
130|                 } catch (Exception e) {
131|                 }
132|                 if (f != null) {
133|                     byte[] dg = f.getPayload();
134|                     String msg = new String(dg, 0, dg.length);
135|                     long sourceAddress = f.getSrcAddr(); //pengirimnya
136|                     new Thread() {
137|                         @Override
138|                         public void run() {
139|                             String[] splittedMSG=splitMSG(msg,':'); //memisahkan perintah atau permintaan dan nilainya
140|                             mainInterface.processMsg(sourceAddress,splittedMSG); //memproses perintah atau permintaan
141|                         }
142|                     }.start();
143|                 }
144|             }
145|         };
146|         receive.start();
147|     }
148| }
149| };
150| receive.start();
151|
152|
153|
154|
155| /**
156| *
157| * sebuah method yang berfungsi untuk memisahkan perintah dan nilainya
158| *
159| * @param msg pesan yang belum dipisahkan
160| *
161| * @param regex karakter pemisah pesan dan nilai
162| *
163| * @return array string berukuran 2 dengan index 0 berupa perintah dan index 1 berupa nilainya
164| */
165| private String[] splitMSG(String msg,char regex){
166|     for (int i=0;i<msg.length();i++){
167|         if(msg.charAt(i)==regex){
168|             return new String[]{
169|                 msg.substring(0,i),
170|                 msg.substring(i+1)
171|             };
172|         }
173|     }
174|     return null;
175| }

```

177 }
178 }

Kode B.4: Base Station pada Preon32 - RadioInit.java

```

1  /*
2  * Copyright (c) 2011., Virtenio GmbH
3  * All rights reserved.
4  *
5  * Commercial software license.
6  * Only for test and evaluation purposes.
7  * Use in commercial products prohibited.
8  * No distribution without permission by Virtenio.
9  * Ask Virtenio for other type of license at info@virtenio.de
10 *
11 * Kommerzielle Softwarelizenz.
12 * Nur zum Test und Evaluierung zu verwenden.
13 * Der Einsatz in kommerziellen Produkten ist verboten.
14 * Ein Vertrieb oder eine Veröffentlichung in jeglicher Form ist nicht ohne Zustimmung von Virtenio erlaubt.
15 * Fr andere Formen der Lizenz nehmen Sie bitte Kontakt mit info@virtenio.de auf.
16 */
17
18 package com.dimas519.Radio;
19
20 import com.virtenio.driver.device.at86rf231.AT86RF231;
21 import com.virtenio.misc.PropertyHelper;
22 import com.virtenio.preon32.node.Node;
23 import com.virtenio.radio.ieee_802_15_4.Frame;
24
25 /** Beispiele einer Initialisierung der Funkverbindung. */
26 public class RadioInit {
27
28 /**
29 * <pre>
30 * {@literal
31 *      Value          TX Output Power [dBm]
32 *      0x0            3.0
33 *      0x1            2.8
34 *      0x2            2.3
35 *      0x3            1.8
36 *      0x4            1.3
37 *      0x5            0.7
38 *      0x6            0.0
39 *      0x7            -1
40 *      0x8            -2
41 *      0x9            -3
42 *      0xA            -4
43 *      0xB            -5
44 *      0xC            -7
45 *      0xD            -9
46 *      0xE            -12
47 *      0xF            -17
48 * }
49 * </pre>
50 */
51 private final int TRANSMIT_POWER = PropertyHelper.getInt("radio.transmit_power", 0, 0, 15);
52
53 /**
54 * A CCA measurement is used to detect a clear channel. Four modes are
55 * specified by IEEE 802.15.4 - 2006:
56 * <p>
57 * mode 1: Energy above threshold. CCA shall report a busy medium upon
58 * detecting any energy above the ED threshold. mode 2: Carrier sense only.
59 * CCA shall report a busy medium only upon the detection of a signal with
60 * the modulation and spreading characteristics of an IEEE 802.15.4
61 * compliant signal. The signal strength may be above or below the ED
62 * threshold. mode 0,3: Carrier sense with energy above threshold. CCA shall
63 * report a busy medium using a logical combination of
64 * <li>Detection of a signal with the modulation and spreading
65 * characteristics of this standard and</li>
66 * <li>Energy above the ED threshold.</li> Where the logical operator may be
67 * configured as either OR (mode 0) or AND (mode 3).
68 */
69 private final int CCA_MODE = PropertyHelper.getInt("radio.cca_mode", 3, 0, 3);
70 private final int CCA_THRESHOLD = PropertyHelper.getInt("radio.cca_threshold", 7, 0, 15);
71
72 /**
73 * Number frame retries
74 */
75 private final int FRAME_RETRIES = PropertyHelper.getInt("radio.frame_retries", 1, 0, 15);
76
77 /**
78 * CSMA related values
79 */
80 private final int CSMA_RETRIES = PropertyHelper.getInt("radio.csma_retries", 1, 0, 7);
81 private final int CSMA_SEED = PropertyHelper.getInt("radio.csma_seed", 511, 0, 1023);
82
83 /**
84 * Backoff exponent
85 */
86 private final int MIN_BACKOFF_EXPONENT = PropertyHelper.getInt("radio.min_backoff_exponent", 3,
87 0, 15);
88 private final int MAX_BACKOFF_EXPONENT = PropertyHelper.getInt("radio.max_backoff_exponent", 5,
89 0, 15);
90
91 /**
92 * Address related parts
93 */

```

```

94  private final int PAN_ID = PropertyHelper.getInt("radio.pan_id", 0xCAFE, 0, 0xFFFF);
95  private final int SHORT_ADDR = PropertyHelper.getInt("radio.short_address", 0, 0, 0xFFFF);
96  private final long LONG_ADDR = PropertyHelper.getLong("radio.long_address", 0);
97  private final boolean I_AM_COORD = PropertyHelper.getBool("radio.i_am_coord", false);
98
99 /**
100 * Radio channel
101 */
102 private final int CHANNEL = PropertyHelper.getInt("radio.channel", AT86RF231.CHANNEL_MIN,
103 AT86RF231.CHANNEL_MIN, AT86RF231.CHANNEL_MAX);
104
105 /**
106 *
107 * @param channel
108 * @param panID
109 * @param addr
110 * @return
111 */
112 public static AT86RF231 initRadio() throws Exception {
113     RadioInit ri = new RadioInit();
114     AT86RF231 radio = Node.getInstance().getTransceiver();
115     radio.open();
116     radio.reset();
117     radio.setState(AT86RF231.STATE_TRX_0FF);
118     radio.setRxSaveMode(true);
119     radio.setMaxFrameRetries(ri.FRAME_RETIRES);
120     radio.setMaxCSMARetries(ri.CSMA_RETIRES);
121     radio.setCSMASeed(ri.CSMA_SEED);
122     radio.setBackoffExponent(ri.MIN_BACKOFF_EXPONENT, ri.MAX_BACKOFF_EXPONENT);
123     radio.setChannel(ri.CHANNEL);
124     radio.setTransmitPower(ri.TRANSMIT_POWER);
125     radio.setCCAMode(ri.CCA_MODE, ri.CCA_THRESHOLD);
126     radio.setAddressFilter(ri.PAN_ID, ri.SHORT_ADDR, ri.LONG_ADDR, ri.I_AM_COORD);
127     radio.setPendingData(false);
128     radio.setState(AT86RF231.STATE_RX_AACK_ON);
129     return radio;
130 }
131 /**
132 * @param addrSRC
133 *          Addresse des Senders
134 * @param panID
135 *          Gruppenadresse von Sender und Empfnger
136 * @param addrDST
137 *          Addresse des Empfnger
138 *
139 * @return Gibt ein vorkonfiguriertes Frame zurck
140 */
141
142 public static Frame prepareFrame(int addrSRC, int panID, int addrDST) {
143     Frame frame = new Frame(Frame.TYPE_DATA | Frame.ACK_REQUEST | Frame.DST_ADDR_16
144     | Frame.INTRA_PAN | Frame.SRC_ADDR_16);
145     frame.setSrcAddr(addrSRC);
146     frame.setSrcPanId(panID);
147     frame.setDestAddr(addrDST);
148     frame.setDestPanId(panID);
149     return frame;
150 }
151
152 }
153 }
```

Kode B.5: Base Station pada Preon32 - MyUsart.java

```

1 package com.dimas519.USART;
2
3
4 import com.dimas519.MainInterface;
5 import com.virtenio.driver.usart.NativeUSART;
6 import com.virtenio.driver.usart.USART;
7 import com.virtenio.driver.usart.USARTEception;
8 import com.virtenio.driver.usart.USARTParams;
9
10 /**
11 * Einfaches Beispiel der Funkbertragung mit Senden und Empfangen.
12 *
13 * * Sebuah kelas objek usart yang berfungsi untuk berkomunikasi dengan pc, laptop, rasp pi melalui interface usb
14 */
15 public class MyUsart {
16     private USART usart;
17
18     private final MainInterface mainInterface;
19     private volatile boolean isWriting = false; //sebuah variabel yang digunakan untuk mengecek apakah ada thread lain yang sedang
20     menulis
21
22     public MyUsart(MainInterface mainInterface) {
23         this.mainInterface = mainInterface;
24         try {
25             this.usart = this.configUSART();
26         } catch (Exception e) {
27             throw new RuntimeException(e);
28         }
29     }
30
31     private USART configUSART() throws Exception {
32         int instanceID = 0;
```

```

34|     USARTParams params = new USARTParams(115200, USART.DATA_BITS_8, USART.STOP_BITS_1, USART.PARTY_NONE); // menggunakan
35|     // usart speed 115200bps
36| 
37|     NativeUSART usart = NativeUSART.getInstance(instanceID);
38| 
39|     try {
40|         usart.close();
41|         usart.open(params);
42|         return usart;
43|     } catch (Exception e) {
44|         return null;
45|     }
46| 
47| /**
48| * 
49| * Sebuah method yang digunakan untuk menerima masukan dari pc menggunakan usart
50| * 
51| */
52| 
53| public void run() {
54|     while (true) {
55|         try {
56|             while (usart.available() > 0) {
57|                 byte[] input = new byte[128]; //dibuat 128 karena max dari radio hanya 127.
58|                 try {
59|                     usart.readFully(input,0,128);
60|                 } catch (USARTException e) {
61|                     System.out.println("uart_read_error" + e.getMessage());
62|                 }
63| 
64|                 String inputStr = new String(input); //jadikan sebuah string
65|                 inputStr=inputStr.replace("\0","");
66|                 inputStr=inputStr.replace("\n","");
67| 
68|                 //expected <address node:isi pesan>
69|                 String[] processStr = splitMSG(inputStr, ':'); //memisahkan perintah dan value dengan pemisahnya berupa
69|                 //kata-kata :'
70| 
71|                 if( processStr[1].equals("ok")){
72|                     //do nothing
73|                 }else if (processStr[0]==null) {
74|                     /* ada kondisi dimana usart melakukan kontrol terhadap perangkat lain(dalam hal ini pc)
75|                     dengan berulang enq---enquire---(0x05) dengan ack(0x06) untuk menentukan apakah lawan aktif
76| 
77|                     pengecekan ini dibutuhkan karena format pada umumnya yang digunakan program <command:isi_command>
78|                     sehingga jika pesan enq/ack di split akan menjadi [null,5] untuk enq atau [null,6] untuk ack
79|                     */
80| 
81|                 }else{
82|                     //expected <command ke bs:isi> misalkan setInterval:1000
83|                     String[] isiPesan = splitMSG(processStr[1], ':');
84|                     try {
85|                         this.mainInterface.processMsg(Long.parseLong(processStr[0]), isiPesan); //proses masukan dari pc
86|                     }catch (Exception e){}
87| 
88|                     isWriting = false; // setelah di proses set kembali variabel writing ke false sebagai penanda kalau sudah
89|                     //selesai
90| 
91|                 } catch (USARTException e) {
92|                     System.out.println(e.getMessage());
93|                 }
94|             }
95|         }
96| 
97|     }
98| 
99| /**
100| * 
101| * Sebuah method yang digunakan untuk mengirimkan pesan ke pc menggunakan usart
102| * 
103| */
104| 
105| 
106| public void send(String msg) {
107|     try {
108|         while (isWriting) {
109|             //tidak langsung writing karena satu siklus writing ditandai dengan response ok dari pc. dan pc tidak pernah
109|             //mengkontak duluan
110|             //perlu menunggu agar tidak terjadi penulisan berbarengan dengan pc
111| 
112|             isWriting = true;//set sedang menulis
113| 
114|             this.usart.write(msg.getBytes(), 0, msg.length());
115|             this.usart.flush();
116| 
117|         } catch (USARTException e) {
118|             System.out.println(e.getMessage());
119|         }
120|     }
121| 
122|     private String[] splitMSG(String msg,char regex){
123|         for (int i=0;i<msg.length();i++){
124|             if(msg.charAt(i)==regex){
125|                 return new String[]{
126|                     msg.substring(0,i),
127|                     msg.substring(i+1)
128|                 };
128|             }
128|         }
128|     }

```

```
129|         }
130|     }
131|     return null;
132| }
133|
134|
135}
```


LAMPIRAN C

KODE PROGRAM BASE STATION PADA PC

Kode C.1: Base Station pada PC - BaseStationController.java

```

75         int x=bufferedInputStream.read(data);
76
77 //          if(x==5 || x==6) { //5 adalah enq(enquiry) karakter yaitu katakter kontrol transmisi untuk mengecek apakah
78 //            node yg berlawanan aktif
79 //            break;
80 //
81
82         String usartMsg = new String(data);
83
84 //          usartMsg=usartMsg.substring(0,i);
85 //          usartMsg=usartMsg.replace("\0","");
86 //          usartMsg=usartMsg.replace("\n","");
87 //          System.out.println(usartMsg);
88 //          if(usartMsg.equals("null")){
89
90 //            break;
91 //        }
92         System.out.println("FROM_USART "+usartMsg);
93         String[] msg=usartMsg.split(",");
94         String[] source=msg[0].split(":");
95
96
97         if(source[0].equals("source")){
98             String msgToAPI=msg[1];
99             boolean isSensing=false;
100            String[] splitMSG = msgToAPI.split(",", 3);
101
102            if(!splitMSG[0].equals("intervalRequest")){
103                this.setApiResponse(source[1]);
104                msgToAPI= FormatMSG.formatToAPIFormat(splitMSG); //kalau dia data sensing di format terlebih dahulu
105                isSensing=true;
106                myApi.sendToServer(source[1],msgToAPI,isSensing);
107            }else {
108                msgToAPI = "{" + splitMSG[1] + "}";
109                myApi.sendToServer(source[1],msgToAPI,isSensing);
110
111                //kode ini tidak dipakai karena terlalu banyak blocking, sehingga ada kemungkinan node lain tidak
112                //dapat masuk
113                //solusi nya: biarkan saja, kekurangannya yaitu node yang baru aktif akan menjalankan sensing 1-2 kali
114                //keunggulanya tidak berdampak pada node lain (tidak blocking main thread)
115                //boolean existNode = false;
116                //while (!existNode){
117 //                  for (NodeQueue nodeQueue : this.queueNode) {
118 //                      if (nodeQueue.getAddress() == Long.parseLong(source[1])) {
119 //                          existNode = true;
120 //                          while (nodeQueue.emptyQueue()) {
121 //                              }
122 //                          }
123 //                      }
124 //                  this.setApiResponse(source[1]);
125
126 //              }
127
128
129         }
130
131
132         else{
133             this.setApiResponse("0");
134         }
135     }
136
137 }
138
139 }
140
141 catch(PureJavaIllegalStateException e){
142     BaseStationControllerUSART bs=new BaseStationControllerUSART(this.args);
143     bs.queueNode=this.queueNode;
144 }
145 catch (IOException e) {
146     System.out.println(e.getMessage());
147 }
148 }
149
150
151
152 /**
153 * untuk debug ant, agar debug ant muncul
154 */
155
156 private static DefaultLogger getConsoleLogger() {
157     DefaultLogger consoleLogger = new DefaultLogger();
158     consoleLogger.setErrorPrintStream(System.err);
159     consoleLogger.setOutputPrintStream(System.out);
160     consoleLogger.setMessageOutputLevel(Project.MSG_INFO);
161     return consoleLogger;
162 }
163
164
165 /**
166 * modifikasi dari method time_synchronize.
167 * Modifikasi pada method ini adalah dengan diganti agar menerima target
168 */
169 private void executeAtPreon(String path,String target) {
170     DefaultLogger consoleLogger = getConsoleLogger();

```

```

171|     File buildFile = new File(path);
172|     Project antProject = new Project();
173|     antProject.setUserProperty("ant.file", buildFile.getAbsolutePath());
174|     antProject.addBuildListener(consoleLogger);
175|
176|     antProject.fireBuildStarted();
177|     antProject.init();
178|     ProjectHelper helper = ProjectHelper.getProjectHelper();
179|     antProject.addReference("ant.ProjectHelper", helper);
180|     helper.parse(antProject, buildFile);
181|     antProject.executeTarget(target);
182|     antProject.fireBuildFinished(null);
183}
184}
185
186
187
188 public static void main(String[] args) {
189     new BaseStationControllerUSART(args);
190 }
191
192
193 public void setApiResponse(String nodeAddress) {
194     try {
195         while (writing){} // karena proses push data sensing ke server dilakukan secara async (multi thread) maka ada kemungkinan 2 thread menulis secara bersamaan
196         writing=true;
197         String responseBS="";
198         Long nodeAddress=Long.parseLong(nodeAddress);
199         boolean existNode=false;
200         for (NodeQueue nodeQueue: this.queueNode){
201             if(nodeQueue.getAddress()==nodeAddress){
202                 existNode=true;
203                 if(nodeQueue.emptyQueue()){
204                     responseBS =(nodeAddress+":ok");
205                 }else {
206                     String queueMsg=nodeQueue.getQueue();
207                     queueMsg=queueMsg.replace("{","\"");
208                     queueMsg=queueMsg.replace("}","\"");
209                     queueMsg=queueMsg.replace("\\"","\"");
210                     queueMsg=nodeAddress+":"+queueMsg;
211                     responseBS=queueMsg;
212                 }
213                 break;
214             }
215         }
216         if(!existNode){
217             responseBS =(nodeAddress+":ok");
218         }
219         System.out.println("TO_USART "+responseBS);
220
221         this.dataConnection.write(writeToBytes(responseBS),0,128);
222         writing=false;
223     } catch (IOException e) {
224         System.out.println("failed_api_response");
225         throw new RuntimeException(e);
226     }
227 }
228
229 }
230
231 private byte[] writeToBytes(String msg){
232     byte[] result=new byte[128];
233     for(int i=0;i<msg.length();i++){
234         result[i]= (byte) msg.charAt(i);
235     }
236     return result;
237 }
238
239
240
241
242 }

```

Kode C.2: Base Station pada PC - FormatMSG.java

```

1 package com.dimas519;
2
3 import java.time.LocalDateTime;
4 import java.time.ZoneOffset;
5 import java.time.format.DateTimeFormatter;
6
7
8 public class FormatMSG {
9     private static String timeFormat(String time){
10         long epoch=Long.parseLong(time.split(":")[1]);
11         epoch/=1000; //konversi ke detik
12
13         LocalDateTime date = LocalDateTime.ofEpochSecond(epoch,0, ZoneOffset.ofHours(7));
14
15         //format apinya
16         DateTimeFormatter format = DateTimeFormatter.ofPattern("yyyy-MM-dd_HH:mm:ss");
17         String java_date = date.format(format);
18         return java_date;
19     }
20
21     public static String formatToAPIFormat(String[] splitMSG) {
22         String dateConverted = FormatMSG.timeFormat(splitMSG[0]);

```

```

23     String result = "{\"time\": \"" + dateConverted + "\", \"result\": {" + splitMSG[1] + ", " + splitMSG[2] + "}}";
24     return result;
25 }
26
27 }
28 }
```

Kode C.3: Base Station pada PC - NodeQueue.java

```

1 package com.dimas519;
2
3 import java.util.LinkedList;
4 import java.util.Queue;
5
6 public class NodeQueue {
7     private Long nodeAddress;
8     private Queue<String> queue;
9
10    public NodeQueue(Long nodeAddress){
11        this.nodeAddress=nodeAddress;
12        this.queue= new LinkedList<>();
13    }
14
15    public Long getAddress(){
16        return this.nodeAddress;
17    }
18
19    public void setQueue(String msg){
20        this.queue.add(msg);
21    }
22    public boolean emptyQueue(){
23        return queue.isEmpty();
24    }
25
26    public String getQueue(){
27        return queue.poll();
28    }
29 }
30 }
```

Kode C.4: Base Station pada PC - API.java

```

1 package com.dimas519.retrofit;
2
3
4 import com.dimas519.NodeQueue;
5
6 import retrofit2.Call;
7 import retrofit2.Callback;
8 import retrofit2.Response;
9 import java.util.List;
10
11
12 public class API {
13     private final String endPoint;
14     private List<NodeQueue> queueNode;
15
16
17     public API(String endPoint, List<NodeQueue> queueNode){
18
19         this.endPoint=endPoint;
20         this.queueNode=queueNode;
21     }
22
23     public void sendToServer(String source, String msg, boolean sensing) {
24         System.out.println("TO_API_" + msg);
25         IRequest ireq = RetrofitAPI.getRetro(this.endPoint).create(IRequest.class);
26         Call<String> cl;
27         if(sensing) {
28             cl = ireq.sensing(msg);
29         }else{
30             cl=ireq.interval(msg);
31         }
32
33         cl.enqueue(new Callback<String>() {
34             @Override
35             public void onResponse(Call<String> call, Response<String> response) {
36
37                 if (response.isSuccessful()) {
38                     if (response.body() != null) {
39                         String responseString=response.body();
40                         Long addressNS=Long.parseLong(source);
41                         System.out.println("FROM_API_" + responseString);
42
43                         if(responseString.equals("{\"result\":false}"))|| responseString.equals("{\"result\":true}")){
44                             //do nothing karena dia tidak termasuk kedalam re-config node sensor
45
46                         }else{
47                             boolean addSensor=false;
48                             for (NodeQueue currQueue: queueNode){
49                                 if(currQueue.getAddress()==addressNS){
50                                     currQueue.setQueue(responseString);
51                                     addSensor=true;
52                                     break;
53                             }
54                         }
55                     }
56                 }
57             }
58         });
59     }
60 }
```

```

54        }
55        if(!addSensor){
56            NodeQueue newSensor=new NodeQueue(addressNS);
57            newSensor.setQueue(responseString);
58            queueNode.add(newSensor);
59        }
60    }
61
62
63    } else {
64        System.out.println("fail");
65    }
66}
67
68}
69
70
71    @Override
72    public void onFailure(Call<String> call, Throwable throwable) {
73        System.out.println("fail2");
74    }
75}
76
77 });
78}
79
80
81
82
83
84
85
86
87
88
89}

```

Kode C.5: Base Station pada PC - iRequest.java

```

1 package com.dimas519.retrofit;
2
3
4
5
6 import retrofit2.Call;
7 import retrofit2.http.Body;
8 import retrofit2.http.POST;
9
10 public interface IRequest {
11     @POST("/sensing")
12     Call<String> sensing(@Body String msg);
13
14     @POST("/interval")
15     Call<String> interval(@Body String msg);
16
17
18 }
19

```

Kode C.6: Base Station pada PC - RetrofitAPI.java

```

1 package com.dimas519.retrofit;
2
3
4
5
6 import retrofit2.Retrofit;
7
8 import retrofit2.converter.scalars.ScalarsConverterFactory;
9
10 public class RetrofitAPI{
11     public static Retrofit getRetro(String API_url ) {
12         Gson gson= new GsonBuilder().setLenient().create();
13         Retrofit ret=new Retrofit.Builder()
14             .baseUrl(API_url)
15             .addConverterFactory(ScalarsConverterFactory.create())
16             .addConverterFactory(GsonConverterFactory.create(gson))
17             .build();
18         return ret;
19     }
20 }
21

```

Kode C.7: Base Station pada PC - RetrofitAPI.java

```

1 package com.dimas519.retrofit;
2
3
4
5
6 import retrofit2.Retrofit;
7
8 import retrofit2.converter.scalars.ScalarsConverterFactory;

```

```

9
10 public class RetrofitAPI{
11     public static Retrofit getRetro(String API_url ) {
12         Gson gson= new GsonBuilder().setLenient().create();
13         Retrofit ret=new Retrofit.Builder()
14             .baseUrl(API_url)
15             .addConverterFactory(ScalarsConverterFactory.create())
16         //        .addConverterFactory(GsonConverterFactory.create(gson))
17             .build();
18         return ret;
19     }
20 }
21 }
```

Kode C.8: Base Station pada PC - RetrofitAPI.java

```

1 package com.dimas519.retrofit;
2
3
4
5 import retrofit2.Retrofit;
6
7 import retrofit2.converter.scalars.ScalarsConverterFactory;
8
9
10 public class RetrofitAPI{
11     public static Retrofit getRetro(String API_url ) {
12         Gson gson= new GsonBuilder().setLenient().create();
13         Retrofit ret=new Retrofit.Builder()
14             .baseUrl(API_url)
15             .addConverterFactory(ScalarsConverterFactory.create())
16         //        .addConverterFactory(GsonConverterFactory.create(gson))
17             .build();
18         return ret;
19     }
20 }
```

Kode C.9: Base Station pada PC - pom.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4           xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5     <modelVersion>4.0.0</modelVersion>
6
7     <groupId>org.example</groupId>
8     <artifactId>Skripsi-BSController_UART</artifactId>
9     <version>1.0-SNAPSHOT</version>
10
11    <properties>
12        <maven.compiler.source>8</maven.compiler.source>
13        <maven.compiler.target>8</maven.compiler.target>
14    </properties>
15    <dependencies>
16        <dependency>
17            <groupId>org.apache.ant</groupId>
18            <artifactId>ant</artifactId>
19            <version>1.10.13</version>
20        </dependency>
21
22        <dependency>
23            <groupId>com.squareup.retrofit2</groupId>
24            <artifactId>retrofit</artifactId>
25            <version> 2.9.0 </version>
26        </dependency>
27        <dependency>
28            <groupId>com.squareup.retrofit2</groupId>
29            <artifactId>converter-scalars</artifactId>
30            <version>2.9.0</version>
31        </dependency>
32    </dependencies>
33    <!--
34    <dependency>-->
35    <!--
36    <groupId>com.squareup.retrofit2</groupId>-->
37    <!--
38    <artifactId>converter-gson</artifactId>-->
39    <!--
40    <version>2.9.0</version>-->
41    </dependency>-->
42    </dependencies>
43
44 
```

LAMPIRAN D

KODE PROGRAM NODE SENSOR PADA PC

Kode D.1: Node Sensor pada PC - NodeSensor.java

```
1 import java.io.File;
2
3
4 import org.apache.tools.ant.DefaultLogger;
5 import org.apache.tools.ant.Project;
6 import org.apache.tools.ant.ProjectHelper;
7
8 /**
9  * args pada kelas ini
10 * 0 : path ke buildUser.xml
11 * 1 : path ke build.xml
12 * 2 : port number to preon //sesuaikan dengan yang di context 1
13 * 3 : API endpoint
14 */
15
16
17 public class NodeSensor {
18     public static void main(String[] args) {
19         new NodeSensor();
20     }
21
22     public NodeSensor(){
23         this.executeAtPreon("buildUser.xml","context.set.1");//ubah ke context 1, atau context yg berisikan ant build untuk preon
24         this.executeAtPreon("build.xml","cmd.module.run"); //jalankan modulnya
25
26     }
27
28     private static DefaultLogger getConsoleLogger() {
29         DefaultLogger consoleLogger = new DefaultLogger();
30         consoleLogger.setErrorPrintStream(System.err);
31         consoleLogger.setOutputPrintStream(System.out);
32         consoleLogger.setMessageOutputLevel(Project.MSG_INFO);
33         return consoleLogger;
34     }
35
36
37     private void executeAtPreon(String path,String target) {
38         DefaultLogger consoleLogger = getConsoleLogger();
39         File buildFile = new File(path);
40         Project antProject = new Project();
41         System.out.println(buildFile.getAbsolutePath());
42         antProject.setUserProperty("ant.file", buildFile.getAbsolutePath());
43         antProject.addBuildListener(consoleLogger);
44
45         antProject.fireBuildStarted();
46         antProject.init();
47         ProjectHelper helper = ProjectHelper.getProjectHelper();
48         antProject.addReference("ant.ProjectHelper", helper);
49         helper.parse(antProject, buildFile);
50         antProject.executeTarget(target);
51         antProject.fireBuildFinished(null);
52
53     }
54
55
56
57
58 }
```


LAMPIRAN E

KODE PROGRAM API

Kode E.1: Skripsi.sql

```
1 -- phpMyAdmin SQL Dump
2 -- version 5.1.3
3 -- https://www.phpmyadmin.net/
4 --
5 -- Host: 127.0.0.1
6 -- Waktu pembuatan: 18 Jun 2023 pada 22.08
7 -- Versi server: 10.4.24-MariaDB
8 -- Versi PHP: 7.4.29
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 START TRANSACTION;
12 SET time_zone = "+00:00";
13
14
15 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
16 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
17 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
18 /*!40101 SET NAMES utf8mb4 */;
19
20 --
21 -- Database: 'skripsi'
22 --
23 CREATE DATABASE IF NOT EXISTS 'skripsi' DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
24 USE 'skripsi';
25
26 DELIMITER $$*
27 --
28 -- Prosedur
29 --
30 CREATE DEFINER='root'@'localhost' PROCEDURE 'deleteQueue' ('id' INT) BEGIN
31     DELETE FROM 'queue_update' WHERE 'id'=id;
32 END$$
33
34 CREATE DEFINER='root'@'localhost' PROCEDURE 'getAllTables' () BEGIN
35     SHOW TABLES WHERE 'Tables_in_skripsi' != 'basestation' AND 'Tables_in_skripsi' != 'kota' AND 'Tables_in_skripsi' != 'tipesensor'
36         AND 'Tables_in_skripsi' != 'nodesensor' AND 'Tables_in_skripsi' != 'user' AND 'Tables_in_skripsi' != 'queue_update';
37 END$$
38 CREATE DEFINER='root'@'localhost' PROCEDURE 'getBaseStasion' () BEGIN
39     SELECT * FROM 'basestation';
40 END$$
41
42 CREATE DEFINER='root'@'localhost' PROCEDURE 'getKota' () BEGIN
43     SELECT * FROM 'kota';
44 END$$
45
46 CREATE DEFINER='root'@'localhost' PROCEDURE 'getNodeSensor' (IN 'idNode' INT) begin
47 IF (idNode >= 0) THEN
48     SELECT 'identifier', 'addedTimeStamp', 'lastEditTimeStamp', 'interval' FROM 'nodesensor' WHERE 'idBS'=idNode;
49 ELSE
50     SELECT 'identifier', 'interval', 'kota'.id AS 'idKota', 'basestation'.'nama' AS 'namaBase', 'latitude', 'longitude', 'indoors',
51         'kota'.'nama' AS 'namaKota', 'nodesensor'.'nama' AS 'namaLokasi', 'token' FROM 'nodesensor' INNER JOIN 'basestation' ON '
52             nodesensor'.'idBS'='basestation'.'id' INNER JOIN 'kota' ON 'basestation'.'idKota'='kota'.id';
53 END IF;
54
55 end$$
56
57 CREATE DEFINER='root'@'localhost' PROCEDURE 'getQueue' () BEGIN
58     SELECT 'id', 'command', 'idBS' FROM 'queue_update';
59 END$$
60
61 CREATE DEFINER='root'@'localhost' PROCEDURE 'getSensing' (IN 'namaTabel' VARCHAR(10), IN 'start' DATETIME, IN 'end' DATETIME)
62 BEGIN
63     SELECT * FROM namaTabel WHERE 'timeStamp'>=start AND 'timeStamp'<=end ORDER BY 'timeStamp';
64 END$$
65
66 CREATE DEFINER='root'@'localhost' PROCEDURE 'getTipSensor' (IN 'id' INT) BEGIN
67 IF id !=-1 THEN
68     SELECT 'tipSensor' FROM 'tipSensor' WHERE 'identifier'=id;
69 ELSE
70
```

```

72  SELECT * FROM `tipeSensor`;
73 END IF;
74
75 END$$
76
77 CREATE DEFINER='root'@'localhost' PROCEDURE 'insertBaseStation' (IN `nama` VARCHAR(50), IN `latitude` VARCHAR(20), IN `longitude` VARCHAR(20), IN `idKota` INT) BEGIN
78 START TRANSACTION;
79     INSERT INTO `basestation`(`nama`, `latitude`, `longitude`, `idKota`) VALUES(nama,latitude,longitude,idKota);
80     COMMIT;
81     select last_insert_id();
82
83 END$$
84
85 CREATE DEFINER='root'@'localhost' PROCEDURE 'insertKota' (IN `nama` VARCHAR(100)) BEGIN
86 START TRANSACTION;
87     INSERT INTO `kota`(`nama`) VALUES(nama);
88     COMMIT;
89     select last_insert_id();
90 END$$
91
92 CREATE DEFINER='root'@'localhost' PROCEDURE 'insertNodeSensor' (IN `identifier` VARCHAR(4), IN `nama` VARCHAR(50), IN `token` VARCHAR(10), IN `indoor` TINYINT(1), IN `intervalIN` INT, IN `idBS` INT) BEGIN
93 START TRANSACTION;
94     INSERT INTO `nodesensor`(`identifier`, `nama`, `token`, `indoor`, `interval`, `idBS`)
95     values(identifier, nama, token, indoor, intervalIN, idBS);
96     COMMIT;
97     SELECT `basestation`.`nama` as `namaBase`, `latitude`, `longitude`, `kota`.`nama` as `namaKota` FROM `basestation` INNER JOIN `kota` ON `basestation`.`idKota`=`kota`.`id` WHERE `basestation`.`id` = idBS;
98
99 END$$
100
101 CREATE DEFINER='root'@'localhost' PROCEDURE 'insertQueue' (IN `command` VARCHAR(50), IN `idBS` VARCHAR(4)) BEGIN
102     INSERT INTO `queue_update`(`command`, `idBS`) VALUES(command,idBS);
103 END$$
104
105 CREATE DEFINER='root'@'localhost' PROCEDURE 'login' (IN `name` VARCHAR(10)) BEGIN
106     SELECT `password`, `role` FROM `user` WHERE `username` = name;
107 END$$
108
109 CREATE DEFINER='root'@'localhost' PROCEDURE 'loginSuccess' (IN `tokenInput` VARCHAR(100), IN `usernameInput` VARCHAR(100)) BEGIN
110     UPDATE `user` SET `token`=tokenInput, `lastLogin`= NOW() WHERE `username` = usernameInput;
111 END$$
112
113 CREATE DEFINER='root'@'localhost' PROCEDURE 'signUp' (IN `username` VARCHAR(10), IN `password` VARCHAR(100), IN `email` VARCHAR(30), IN `role` BIT) BEGIN
114     INSERT INTO `user`(`username`, `password`, `email`, `role`, `create`) VALUES(username,password,email,role,NOW());
115 END$$
116
117 CREATE DEFINER='root'@'localhost' PROCEDURE 'updateInterval' (IN `indentifierInput` VARCHAR(4), IN `intervalSensing` INT) begin
118     UPDATE `nodesensor` SET `interval` = intervalSensing, `lastEditTimeStamp` = NOW() WHERE `identifier` = indentifierInput;
119 end$$
120
121 DELIMITER ;
122
123 --
124
125 -- Struktur dari tabel 'basestation'
126 --
127
128 CREATE TABLE `basestation` (
129     `id` int(11) NOT NULL,
130     `nama` varchar(50) COLLATE utf8mb4_bin NOT NULL,
131     `latitude` varchar(20) COLLATE utf8mb4_bin NOT NULL,
132     `longitude` varchar(20) COLLATE utf8mb4_bin NOT NULL,
133     `idKota` int(11) NOT NULL
134 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;
135
136
137 --
138
139 -- Struktur dari tabel 'kota'
140 --
141
142 CREATE TABLE `kota` (
143     `id` int(11) NOT NULL,
144     `nama` varchar(50) COLLATE utf8mb4_bin NOT NULL
145 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;
146
147
148 --
149
150 -- Struktur dari tabel 'nodesensor'
151 --
152
153
154 CREATE TABLE `nodesensor` (
155     `identifier` varchar(4) COLLATE utf8mb4_bin NOT NULL,
156     `nama` varchar(50) COLLATE utf8mb4_bin NOT NULL,
157     `token` varchar(10) COLLATE utf8mb4_bin NOT NULL,
158     `indoor` tinyint(4) NOT NULL COMMENT '0=false,1=true',
159     `addedTimeStamp` datetime NOT NULL DEFAULT current_timestamp(),
160     `lastEditTimeStamp` datetime DEFAULT NULL,
161     `interval` int(11) NOT NULL,
162     `idBS` int(11) NOT NULL
163 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;
164
165
166

```

```

167-- 
168-- Struktur dari tabel 'queue_update'
169--
170CREATE TABLE 'queue_update' (
171    'id' int(11) NOT NULL,
172    'command' varchar(50) COLLATE utf8mb4_bin NOT NULL,
173    'idBS' varchar(4) COLLATE utf8mb4_bin NOT NULL,
174) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;
175
176-- -----
177-- 
178-- Struktur dari tabel 'tipesensor'
179--
180CREATE TABLE 'tipesensor' (
181    'id' int(11) NOT NULL,
182    'tipeSensor' varchar(10) COLLATE utf8mb4_bin NOT NULL,
183    'identifier' varchar(4) COLLATE utf8mb4_bin NOT NULL,
184) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;
185
186-- -----
187-- 
188-- Struktur dari tabel 'user'
189--
190CREATE TABLE 'user' (
191    'id' int(11) NOT NULL,
192    'username' varchar(50) COLLATE utf8mb4_bin NOT NULL,
193    'password' varchar(100) COLLATE utf8mb4_bin NOT NULL,
194    'email' varchar(30) COLLATE utf8mb4_bin NOT NULL,
195    'lastLogin' datetime DEFAULT NULL,
196    'token' varchar(50) COLLATE utf8mb4_bin DEFAULT NULL,
197    'create' datetime NOT NULL,
198    'role' smallint(6) NOT NULL COMMENT '0=_user,_1_untuk_admin'
199) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;
200
201-- 
202-- Dumping data untuk tabel 'user'
203-- 
204INSERT INTO 'user' ('id', 'username', 'password', 'email', 'lastLogin', 'token', 'create', 'role') VALUES
205(14, 'admin', '$2b$12$veHGuILm6HKpnfEuGfzs0ioqvjw6aI9wsLLM7/i1qoj2.kaFFin.', 'admin@sensorVision', '2023-06-01_16:32:41', 'mfKiXSoSrMIS8AZdUrh0LXiG422iiUBYT1mGbEh6', '2023-05-30_03:59:57', 1),
206(15, 'admin1', '$2b$12$wu.mik/Im7aaH7HzmvvlHa.jVpesgJ3xKACWMmiJQ35zto4nf1Xc7i', 'admin1@sensorVision', '2023-06-15_09:17:45', 'POSTMAN-TOKEN', '2023-05-30_04:00:14', 1);
207
208-- 
209-- Indexes for dumped tables
210-- 
211-- Indeks untuk tabel 'basestation'
212ALTER TABLE 'basestation'
213    ADD PRIMARY KEY ('id'),
214    ADD UNIQUE KEY 'nama' ('nama'),
215    ADD KEY 'berada di' ('idKota');
216
217-- 
218-- Indeks untuk tabel 'kota'
219ALTER TABLE 'kota'
220    ADD PRIMARY KEY ('id');
221
222-- 
223-- Indeks untuk tabel 'nodesensor'
224ALTER TABLE 'nodesensor'
225    ADD PRIMARY KEY ('identifier'),
226    ADD KEY 'berada' ('idBS');
227
228-- 
229-- Indeks untuk tabel 'queue_update'
230ALTER TABLE 'queue_update'
231    ADD PRIMARY KEY ('id'),
232    ADD KEY 'fk base' ('idBS');
233
234-- 
235-- Indeks untuk tabel 'tipesensor'
236ALTER TABLE 'tipesensor'
237    ADD PRIMARY KEY ('id'),
238    ADD KEY 'IdBS' ('identifier');
239
240-- 
241-- Indeks untuk tabel 'user'
242ALTER TABLE 'user'
243    ADD PRIMARY KEY ('id'),
244    ADD KEY 'fk base' ('username');
245
246-- 
247-- AUTO_INCREMENT untuk tabel yang dibuang
248ALTER TABLE 'queue_update' AUTO_INCREMENT=15;
249ALTER TABLE 'tipesensor' AUTO_INCREMENT=16;
250ALTER TABLE 'user' AUTO_INCREMENT=16;
251
252-- 
253-- Indeks untuk tabel 'queue_update'
254ALTER TABLE 'queue_update' ADD INDEX 'idx_command' (command);
255
256-- 
257-- Indeks untuk tabel 'tipesensor'
258ALTER TABLE 'tipesensor' ADD INDEX 'idx_tipeSensor' (tipeSensor);
259
260-- 
261-- Indeks untuk tabel 'user'
262ALTER TABLE 'user' ADD INDEX 'idx_username' (username);
263

```

```

264 -- AUTO_INCREMENT untuk tabel 'basestation'
265 --
266 ALTER TABLE 'basestation'
267   MODIFY 'id' int(11) NOT NULL AUTO_INCREMENT;
268
269 --
270 -- AUTO_INCREMENT untuk tabel 'kota'
271 --
272 ALTER TABLE 'kota'
273   MODIFY 'id' int(11) NOT NULL AUTO_INCREMENT;
274
275 --
276 -- AUTO_INCREMENT untuk tabel 'queue_update'
277 --
278 ALTER TABLE 'queue_update'
279   MODIFY 'id' int(11) NOT NULL AUTO_INCREMENT;
280
281 --
282 -- AUTO_INCREMENT untuk tabel 'tipesensor'
283 --
284 ALTER TABLE 'tipesensor'
285   MODIFY 'id' int(11) NOT NULL AUTO_INCREMENT;
286
287 --
288 -- AUTO_INCREMENT untuk tabel 'user'
289 --
290 ALTER TABLE 'user'
291   MODIFY 'id' int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=17;
292
293 --
294 -- Ketidakleluasaan untuk tabel pelimpahan (Dumped Tables)
295 --
296
297 --
298 -- Ketidakleluasaan untuk tabel 'basestation'
299 --
300 ALTER TABLE 'basestation'
301   ADD CONSTRAINT 'berada di' FOREIGN KEY ('idKota') REFERENCES 'kota' ('id');
302
303 --
304 -- Ketidakleluasaan untuk tabel 'nodesensor'
305 --
306 ALTER TABLE 'nodesensor'
307   ADD CONSTRAINT 'berada' FOREIGN KEY ('idBS') REFERENCES 'basestation' ('id');
308
309 --
310 -- Ketidakleluasaan untuk tabel 'queue_update'
311 --
312 ALTER TABLE 'queue_update'
313   ADD CONSTRAINT 'fk base' FOREIGN KEY ('idBS') REFERENCES 'nodesensor' ('identifier');
314
315 --
316 -- Ketidakleluasaan untuk tabel 'tipesensor'
317 --
318 ALTER TABLE 'tipesensor'
319   ADD CONSTRAINT 'tipesensor_ibfk_1' FOREIGN KEY ('identifier') REFERENCES 'nodesensor' ('identifier');
320
321 COMMIT;
322
323 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
324 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
325 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Kode E.2: API - Main.py

```

1 import uvicorn
2 from fastapi import FastAPI ,Request, status, HTTPException
3 from starlette.middleware.cors import CORSMiddleware
4 from fastapi import FastAPI
5
6
7
8
9 #configurasi db dll
10 from ConfigurationAPI import Configuration
11
12
13 import Initialization
14 from Controller.WSNController import WSNController
15
16 # server
17 from ServerQueue import ServerQueue
18 # import ServerVariable
19
20
21 # INITIALIZING
22 print("initializing")
23
24 print("getting_Configuration")
25 Config=Configuration()
26
27 # INITIALIZING DB
28 databaseAPI=Initialization.initDatabase(Config)
29
30 # INITIALIZING WSN
31 WSN=Initialization.initWSN(databaseAPI)
32 wsnController=WSNController(WSN)
33
```

```

34
35
36
37 queueServer = ServerQueue(databaseAPI)
38
39 print("initializing_Done")
40
41
42
43
44
45
46
47 #start api
48 if __name__ == "__main__":
49     from ConfigurationAPI import Configuration
50
51     print("initializing_api")
52     Config=Configuration()
53     serverConf=Config.getServer();
54
55     uvicorn.run("Main:api",
56                 host=serverConf['IPAddress'],
57                 port=int(serverConf['port']),
58                 reload=serverConf['reload'],
59                 workers=int(serverConf['worker']))
60             )
61     print("initializing_api_done")
62
63
64
65
66
67
68 api=FastAPI()
69
70
71 serverAllowed=[*
72 ]
73
74
75
76 api.add_middleware(
77     CORSMiddleware,
78     allow_credentials=False,
79     allow_origins=serverAllowed,
80     allow_methods=["*"],
81     allow_headers=['Content-Type,Authorization,Content-Length,X-Requested-With,Accept'],
82 )
83
84 def raiseWrongArguments():
85     raise HTTPException(
86         status_code=status.HTTP_422_UNPROCESSABLE_ENTITY,
87         detail="wrong_arguments",
88     )
89
90 def raiseUnAuth():
91     raise HTTPException(
92         status_code=status.HTTP_401_UNAUTHORIZED,
93         detail="Key_mismatch",
94     )
95
96 # @api.get("/")
97 # async def Test():
98 #     from fastapi.responses import HTMLResponse
99
100
101 @api.post("/login")
102 async def login(value: Request):
103     data= await value.json()
104     try:
105         username=data['username']
106         password=data['password']
107         token=data['token']
108     except:
109         raiseWrongArguments()
110
111     result=databaseAPI.Login(username,password,token)
112     return {"result": result}
113
114
115 @api.post("/signup")
116 async def signUp(value: Request):
117     data= await value.json()
118     try :
119         username=data['username']
120         password=data['password']
121         email=data['email']
122     except:
123         raiseWrongArguments()
124
125     result=databaseAPI.signUP(username,password,email)
126     return {"result": result}
127
128
129 @api.post("/kota")
130 async def insertKota(value: Request):
131     data= await value.json()
132     try:

```

```
133     nama=data['name']
134 except:
135     raiseWrongArguments()
136
137 result=databaseAPI.insertKota(nama)
138 return {"result":result}
139
140 @api.get("/kota")
141 async def getBaseStasion():
142     result=databaseAPI.getKota()
143     return {"result":result}
144
145
146
147 #mengambil semua informasi bs (node sensor,kota)
148 @api.get("/bs")
149 async def getBS():
150     result=databaseAPI.getBaseStasion()
151     return {"result":result}
152
153
154 #menginput base stasion baru
155 @api.post("/bs")
156 async def insertBS(value: Request):
157     data= await value.json()
158     try:
159         nama=data['nama']
160         latitude=data['latitude']
161         longitude=data['longitude']
162         idKota=data['idKota']
163     except:
164         raiseWrongArguments()
165
166     result=databaseAPI.insertBaseStation(nama,latitude,longitude,idKota)
167     return {"result":result}
168
169
170 #mengambil node sensor
171 @api.get("/node")
172 async def getNodeSensor(idBS: str):
173     # data= await value.json()
174     # try:
175     #     idBS=data['idBS']
176     # except:
177     #     raiseWrongArguments()
178
179     if(idBS==None):
180         raiseWrongArguments()
181
182     result=databaseAPI.getNodeSensor(idBS)
183     return {"result":result}
184
185 #menginputkan node baru
186 @api.post("/node")
187 async def insertNodeSensor(value: Request):
188     data= await value.json()
189     try:
190         identifier=data['identifier'].lower()
191         idBS=data['idBS']
192         nama=data['nama']
193         indoor=int(data['indoor'])
194         interval=int(data['interval'])
195         tipeSensor=data['tipeSensor']
196     except:
197         raiseWrongArguments()
198
199     if(interval<1000):
200         raiseWrongArguments()
201     return None
202
203     result=wsnController.insertNewWSN(databaseAPI,identifier, nama, indoor, interval, tipeSensor, idBS)
204
205     # return True
206     if(result[0]):
207         res={
208             "success":True,
209             "key":result[1]
210         }
211         return {"result":res}
212     else:
213         return {"result":False}
214
215
216
217 @api.post("/tipe")
218 async def insertNodeSensor(value: Request):
219     data= await value.json()
220     try :
221         tipeSensor=data['tipeSensor']
222         identifier=data['node'].lower()
223     except:
224         raiseWrongArguments()
225
226
227     result=databaseAPI.insertTipe(tipeSensor,identifier)
228     return {"result":result}
229
230
231
```

```

232
233 @api.post("/update")
234 async def insertNodeSensor(value: Request):
235     data= await value.json()
236     try:
237         identifier=data['node'].lower()
238         command=data['command']
239     except:
240         raiseWrongArguments()
241
242     result=wsnController.updateQueue(databaseAPI,identifier.lower(),command)
243
244
245     return {"result":result}
246
247
248
249 @api.post("/sensing")
250 async def sensing(value: Request):
251     data= await value.json()
252     try:
253         identifier=data['id'].lower()
254         time=data['time']
255         sensingData=data['result']
256         token=data['key']
257     except :
258         raiseWrongArguments()
259
260     result=wsnController.sensingProcedure(databaseAPI,identifier, token,time,sensingData, data)
261
262     if(result==401):
263         raiseUnAuth()
264     else:
265         return result
266
267 @api.get("/interval")
268 async def changeInterval(node: str):
269     node=node.lower()
270     result=wsnController.getInterval(node)
271
272     return {"setInterval":result}
273
274
275
276 @api.post("/data")
277 async def getData(value: Request):
278     data= await value.json()
279     try:
280         identifier=data['node']
281         start=data['start']
282         end=data['end']
283         interval=int(data['interval']))
284     except :
285         raiseWrongArguments()
286
287         statistics=None;
288     try:
289         statistics=data['stat']
290     except:
291         statistics='avg'
292
293
294     result=wsnController.getData(databaseAPI,identifier,start,end,interval,statistics)
295
296     return {"result":result}
297
298
299
300 @api.get("/realTime")
301 async def getDataRealTime(node: str):
302     identifier=node.lower()
303     result=wsnController.getRealTime(identifier)
304     return result

```

Kode E.3: API - Initialization.py

```

1 from Model.WSN import WSN
2 from Controller.DatabaseController import DataBaseController
3
4 def initDatabase(Config):
5     print("initializing_database")
6     databaseConf=Config.getDataBase();
7
8     databaseAPI=DataBaseController(
9         databaseConf['ipAddress'],
10        int(databaseConf['port']),
11        databaseConf['database'],
12        databaseConf['username'],
13        databaseConf['password']
14    )
15    print("database_initialized")
16    return databaseAPI
17
18 def initWSN(databaseController):
19     wsnRegisterd=[]
20     print("load_WSN")
21     allNodes=databaseController.getNodeSensor(-1) #-1 tidak spesifik lokasi (ambil semua)
22     allQueue=databaseController.getQueue()

```

```

23| allSensor=databaseController.getSensorType(-1)
24| allSensingTable=databaseController.getTable()
25|
26|
27|
28| print(str(len(allNodes))+" _WSN_load")
29|
30| for i in range(0,len(allNodes)):
31|     currNodes=allNodes[i]
32|     listSensor=[]
33|
34|
35|     for sensor in allSensor:
36|         if(sensor['identifier']==currNodes['identifier']):
37|             listSensor.append(sensor['tipeSensor'])
38|
39|
40|     queuedWSN=None;
41|     for queue in allQueue:
42|
43|         if queue['idBS']==currNodes['identifier']:
44|             queuedWSN=queue
45|
46|
47|     newWSN=WSN(identifier= currNodes['identifier'], token= currNodes['token'],
48|                 ,sensorType=listSensor, interval=currNodes['interval'],
49|                 ,latitude=currNodes['latitude'], longitude=currNodes["longitude"]
50|                 ,kota=currNodes["namaKota"],queue=queuedWSN)
51|
52|
53|     wsnSensingTable=[]
54|
55|     for sensingTable in allSensingTable:
56|         if(sensingTable[0].split("-")[2]==currNodes['identifier'] or sensingTable[0].split("-")[0]==currNodes['identifier']):
57|             :
58|             wsnSensingTable.append(sensingTable[0])
59|
60|
61|     newWSN.setSensingTable(wsnSensingTable)
62|
63|
64|     wsnRegisterd.append(newWSN)
65|     print(str(i+1)+" .._wsn_ "+currNodes['identifier']+_"_loaded")
66| print(str(len(allNodes))+" _WSN_loaded")
67| return wsnRegisterd

```

Kode E.4: API - ServerQueue.py

```

1 class ServerQueue:
2     def __init__(self,database):
3         self.sensingTables=database.getTable()
4         print("getting_update_queue")
5         self.queue=database.getQueue() #id,command,idbs
6         print("update_queue_done",self.queue)
7
8
9
10    def getQueue(self,identifier):
11        for i in range(0,len(self.queue),1):
12            if(identifier==self.getQueue[i][2]):
13                return self.queue[1]
14
15    return None;
16
17    def insertNewQueue(self,identifier,command):
18        #mencari apakah udh ada command yang sama
19        overlap=self.overlapCommand(identifier,command)
20        if(bool(overlap)): #kalau overlap drop yg lama
21            self.database.deleteQueue(overlap)
22
23        newID=self.database.insertQueue(identifier,command)
24        if(newID !=-1):
25            self.queue.append((newID,command,identifier))
26            return True
27        else:
28            return False
29
30    def overlapCommand(self,identifier,command):
31        #mencari apakah udh ada command yang sama
32
33        commandBaru=command.split(";")[0]
34        for i in range(0,len(self.queue),1):
35            if(self.queue[i][2]==identifier):
36                commandQueue=self.queue[i][1].split(";")[0]
37                if(commandQueue==commandBaru): #kalau sama
38                    temp=self.queue[i][0]
39                    del self.queue[i]
40                    return temp
41
42    return 0

```

Kode E.5: API - ConfigurationAPI.py

```

1 import configparser
2 import os
3 class Configuration:
4
5     def __init__(self) :
6         print("getting_configuration")
7
8
9     dir = os.path.dirname(__file__)
10    filename = os.path.join(dir, 'Config/serverConf.ini')
11
12    self.config = configparser.ConfigParser()
13    self.config.read(filename)
14    if type(self.config) is None:
15        print("not_found_config_file")
16    server=self.config['API_SERVER']
17
18
19
20    def get DataBase(self):
21        database=self.config['Database']
22        return database
23
24
25    def get Server(self):
26        server=self.config['API_SERVER']
27        return server

```

Kode E.6: API - Database.py

```

1 import mysql.connector
2
3 class mysqlDB:
4     def __init__(self,ip,port,database,username,password) :
5         self.ip=ip
6         self.port=port
7         self.database=database
8         self.username=username
9         self.password=password
10
11     def connect(self):
12         self.mydb = mysql.connector.connect(
13
14             host=self.ip,
15             port=self.port,
16             database=self.database,
17
18             user=self.username,
19             password=self.password
20         )
21
22     def closeConnect(self):
23         self.mydb.close()
24
25     def executeSelectQuery(self,query,dictionary=True):
26         self.connect()
27         mycursor = self.mydb.cursor( dictionary=dictionary)
28         try:
29             mycursor.execute(query)
30             data=mycursor.fetchall()
31             return data
32         except mysql.connector.IntegrityError as err:
33             return False
34         finally:
35             self.closeConnect()
36
37
38     def executeNonSelectQuery(self ,query):
39         self.connect()
40         try:
41             mycursor = self.mydb.cursor()
42             mycursor.execute(query)
43             self.mydb.commit()
44             if mycursor.lastrowid == 0:
45                 return 1
46             return mycursor.lastrowid
47         except mysql.connector.IntegrityError as err:
48             print(err)
49             return None
50         finally:
51             self.closeConnect()
52

```

Kode E.7: API - DatabaseController.py

```

1 from Controller.Database import mysqlDB
2 import bcrypt
3
4
5 class DataBaseController:
6     def __init__(self, ip, port, database, username, password) :
7         self.db=mysqlDB(ip, port, database, username, password)
8
9     def getTables(self):
10        sql = "CALL `getAllTables`"

```

```

11     result=self.db.executeSelectQuery(sql, dictionary=False)
12     return result
13
14 def Login(self, username, password, token):
15     sql = "CALL_Login('{}')".format(username)
16     result=self.db.executeSelectQuery(sql)
17     if(len(result)==0):
18         return -8
19     hashPassword=(result[0]['password']).encode()
20     truePassword=bcrypt.checkpw(password.encode(), hashPassword)
21     if(not truePassword):
22         return -9;
23     else :
24         sql = "CALL_loginSuccess('{}','{}')".format(token, username)
25         self.db.executeNonSelectQuery(sql)
26     return result[0]['role']
27
28 def signUP(self, username, password, email, role=0):
29     hashedPassword = bcrypt.hashpw(password.encode(), bcrypt.gensalt())
30     sql = "CALL_signUp('{}','{}','{}',{})".format(username, hashedPassword.decode(), email, role)
31     try:
32         result=self.db.executeNonSelectQuery(sql)
33     except:
34         return False
35     if(result == None):
36         return False
37     else :
38         return True
39
40 def insertKota(self, namaKota):
41     sql="call_insertKota('{}')".format(namaKota)
42     row=self.db.executeSelectQuery(sql) #menggunakan execute select karena dia akan mengembalikan id
43     return {"result":bool(row),"id":row[0]["last_insert_id()"]}
44
45 def getKota(self):
46     sql="CALL_getKota();"
47     result=self.db.executeSelectQuery(sql)
48     return result
49
50 def getBaseStasion(self):
51     sql="CALL_getBaseStasion();"
52     resultLokasi=self.db.executeSelectQuery(sql)
53     return resultLokasi
54
55 def insertBaseStation(self, nama, latitude, longitude, fk):
56     sql="CALL_insertBaseStation('{}','{}','{}',{})".format(nama, latitude, longitude, fk)
57     row=self.db.executeSelectQuery(sql)
58     if(row ==False):
59         return row
60     else:
61         return {"result":bool(row),"id":row[0]["last_insert_id()"]}
62
63 def getNodeSensor(self, id):
64     sql="CALL_getNodeSensor({})".format(id);
65     result=self.db.executeSelectQuery(sql)
66     return result
67
68 def insertNodeSensor(self, identifier, nama, token, indooor, interval, idBS):
69
70     sql="CALL_insertNodeSensor('{}','{}','{}',{},{},{})".format(identifier, nama, token, indooor, interval, idBS)
71     row=self.db.executeSelectQuery(sql)
72     return row
73
74 def getSensorType(self,id):
75     sql="CALL_getTipeSensor({})".format(id)
76     result=self.db.executeSelectQuery(sql)
77     return result
78
79 def insertTipe(self,tipeSensor,idNode):
80     sql="INSERT INTO `tipesensor`(`tipeSensor`, `identifier`) VALUES "
81
82     for i in range(0,len(tipeSensor),1):
83         if(i!=0):
84             sql+=","
85             sql+="('{}','{}')".format(tipeSensor[i],idNode)
86     row=self.db.executeNonSelectQuery(sql)
87     return bool(row)
88
89 def insertSensing(self,time,identifier,data):
90     tahun=time[2:4]
91     bulan=time[5:7]
92     sql="INSERT INTO `{}`(`timeStamp`, `suhu`, `kelembapan`, `tekanan`, `akselerasi`) VALUES ".format(str(identifier)+"-"+bulan+"-"+tahun)
93     sql+="('{}',{},{},{},{}{})".format(time,data['T'],data['rh'],data['Pa'],data['a'])
94
95     row=self.db.executeNonSelectQuery(sql)
96     return bool(row)
97
98
99 def getQueue(self):
100    sql="CALL_getQueue()"
101    result=self.db.executeSelectQuery(sql,True)
102    return result
103
104 def insertQueue(self,identifier,command):
105
106     sql="CALL_insertQueue('{}','{}')".format(command,identifier)
107
108

```

```

109     row=self.db.executeNonSelectQuery(sql)
110     return row
111
112     def deleteQueue(self,id):
113         sql="CALL_deleteQueue('{})'.format(id)
114         self.db.executeNonSelectQuery(sql);
115
116     def getSensingData(self,namaTable,start,end):
117         sql="SELECT_*_FROM_{0}_WHERE_timeStamp>'{1}'_AND_timeStamp<'{2}'_ORDER_BY_timeStamp_ASC".format(namaTable,start,end)
118         # return sql
119
120         return self.db.executeSelectQuery(sql)
121
122
123     def executeDb(self,query):
124         return self.db.executeSelectQuery(query)
125
126     def updateInterval(self,identifier,interval):
127         query ="CALL_updateInterval('{}',{})".format(identifier,interval)
128         return self.db.executeNonSelectQuery(query)

```

Kode E.8: API - WSNController.py

```

1 import numpy as np
2 from Model.WSN import WSN
3
4
5 class WSNController:
6     def __init__(self,allWSN):
7         self.allWSN=allWSN
8
9     def __searchWSN(self,identifier):
10        for wsn in self.allWSN:
11            if(wsn.getIdentifier()==identifier):
12                return wsn
13        return None
14
15    def __nameTable(self,identifier,time,newFormat=False):
16        tahun=time[2:4]
17        bulan=time[5:7]
18        namaTabel=(str(identifier)+"-"+str(bulan)+"-"+str(tahun))
19
20        return namaTabel
21
22    def __isTableCreated(self, namaTabel,selectedWSN):
23
24        tableCreated=False
25        for table in selectedWSN.getSensingTable():
26            if(table==namaTabel):
27                tableCreated=True
28                break
29
30        return tableCreated
31
32    def __createTable(self, allSensor, identifier, time):
33
34        namaTabel=self.__nameTable(identifier,time)
35
36
37        sql="""CREATE TABLE '{0}' (
38            'timeStamp' datetime NOT NULL DEFAULT current_timestamp(),""".format(namaTabel)
39        concat=False
40        for sensor in allSensor:
41            if(concat):
42                sql+=","
43                if(sensor=="Suhu"):
44                    sql+="`suhu`_float_NOT_NULL"
45                    concat=True
46                elif(sensor == "Kelembapan"):
47                    sql+="`kelembapan`_int(11)_NOT_NULL"
48                    concat=True
49                elif(sensor=="Tekanan"):
50                    sql+="`tekanan`_int(11)_NOT_NULL"
51                    concat=True
52                elif(sensor=="Akselerasi"):
53                    sql+="`akselerasi`_varchar(20)_NOT_NULL"
54                    concat=True
55
56
57        sql+=",PRIMARY_KEY(`timeStamp`))"
58        return sql
59
60
61
62    def sensingProcedure(self, dbController, identifier, token, time, sensingData, dataJSON):
63        selectedWSN=self.__searchWSN(identifier)
64
65        if(selectedWSN==None):
66            return 401
67
68        if not (selectedWSN.getToken()==token):
69            return 401
70
71        selectedWSN.setLastData(dataJSON)
72        # time=ServerVariable.getTime(selectedWSN.getOffsetHour(), selectedWSN.getOffsetMinutes())
73        nameTable=self.__nameTable(identifier,time)
74        isCreated=self.__isTableCreated(nameTable,selectedWSN)

```

```

75
76     if(not isCreated):
77
78         sql=self.__createTable(selectedWSN.getSensorType(), identifier,time)
79         dbController.executeDb(sql)
80         selectedWSN.addSensingTable(nameTable)
81
82
83     result=dbController.insertSensing(time,identifier,sensingData)
84
85     if(selectedWSN.getQueue()==None):
86         return {"result":result}
87     else:
88         queue=selectedWSN.getQueue();
89
90         # hapus lagi queue
91         dbController.deleteQueue(selectedWSN.getQueue()['id'])
92         selectedWSN.setQueue(None)
93         command=queue['command'].split(":")
94
95     return {str(command[0]):int(command[1])}
96
97
98 def updateQueue(self, dbController, identifier, command):
99     selectedWSN=self.__searchWSN(identifier)
100    newID=dbController.insertQueue(identifier,command)
101
102    commandSplit=command.split(":")
103    if(commandSplit[0]=='setInterval'):
104        print("setNewInterval")
105        dbController.updateInterval(identifier,commandSplit[1])
106        selectedWSN.setInterval(commandSplit[1])
107
108    if(selectedWSN.getQueue()==None):
109
110        newQueue={
111            "id":newID
112            , "command":command
113            , "idBS":identifier
114        }
115        selectedWSN.setQueue(newQueue)
116
117    else:
118        queue=selectedWSN.getQueue()
119        dbController.deleteQueue(selectedWSN.getQueue()['id'])
120        queue['id']=newID
121        queue['command']=command
122
123    return True
124
125 def getInterval(self,identifier):
126     selectedWSN=self.__searchWSN(identifier)
127     return selectedWSN.getInterval()
128
129 def getData(self, dbController,identifier,start,end,interval,statistics):
130     if (type(identifier)==list):
131         result={}
132         for bs in identifier:
133             bsData=self.getData(dbController,bs,start,end,interval,statistics)
134             result[bs]=bsData
135
136
137     return result
138 else:
139     identifier=identifier.lower()
140     selectedWSN=self.__searchWSN(identifier)
141     if (selectedWSN==None):
142         return False
143
144     nameTable=self.__nameTable(identifier,start)
145     isCreated=self.__isTableCreated(nameTable,selectedWSN)
146
147     if(isCreated):
148         sensingData=dbController.getSensingData(nameTable,start,end)
149
150         if(len(sensingData)==0): #mengembalikan false kalau dia kosong
151             return False
152
153         if(statistics=='raw'):
154             output=self.rawData(sensingData,interval)
155         elif (statistics=='median'):
156             output=self.medianData(sensingData,interval)
157
158         elif(statistics=="inQ"):
159             output=self.statDataQuartile(sensingData)
160         elif("split" in statistics):
161             stat=statistics.split("-")
162             if(len(stat)==0):
163                 return False
164             try:
165                 interval=int(stat[1])
166                 output=self.statDataRange(sensingData,interval)
167             except:
168                 return False
169
170
171     else:
172         output=self.averageData(sensingData,interval)
173

```

```

174         return output
175     else:
176         return False
177
178
179
180
181
182
183
184
185     def averageData(self,sensingData,interval):
186
187         currTime=sensingData[0]['timeStamp'];
188         currIntervalLeft=interval
189         output=[]
190         tempTime=[sensingData[0]['timeStamp']]
191         tempSuhu=[sensingData[0]['suhu']]
192         tempKelembapan=[sensingData[0]['kelembapan']]
193         tempTekanan=[sensingData[0]['tekanan']]
194
195         akselerasiData=sensingData[0]['akselerasi']
196         akselerasiData=akselerasiData.replace("[","")
197         akselerasiData=akselerasiData.replace("]","");
198
199         splitAkselerasi=akselerasiData.split(",")
200         tempX=[float(splitAkselerasi[0])]
201         tempY=[float(splitAkselerasi[1])]
202         tempZ=[float(splitAkselerasi[2])]
203
204         # tempAkselerasi=[sensingData[0]['alse;erasi']]
205         for i in range(1,len(sensingData)):
206             currData=sensingData[i]
207
208             diffInterval=(currData['timeStamp']-currTime).total_seconds()
209             currIntervalLeft-=diffInterval
210
211             currTime=currData['timeStamp']
212
213             if(currIntervalLeft <=0): #kalau habis
214
215                 currIntervalLeft=interval
216                 cell={}
217                 lengthData=len(tempSuhu)
218                 cell['timeStamp']= tempTime[int(lengthData/2)]
219                 cell['suhu']= round(sum(tempSuhu)/len(tempSuhu) ,2)
220                 cell['kelembapan']=round( sum(tempKelembapan)/len(tempKelembapan) ,2)
221                 cell['tekanan']= round( sum(tempTekanan)/len(tempTekanan),2)
222                 cell['akselerasi']={
223                     "x":round(sum(tempX)/len(tempX),2),
224                     "y":round(sum(tempY)/len(tempY),2),
225                     "z":round(sum(tempZ)/len(tempZ),2)
226                 }
227
228                 output.append(cell)
229                 tempTime=[currData['timeStamp']]
230                 tempSuhu=[currData['suhu']]
231                 tempKelembapan=[currData['kelembapan']]
232                 tempTekanan=[currData['tekanan']]
233
234                 akselerasiData=currData['akselerasi']
235                 akselerasiData=akselerasiData.replace("[","");
236                 akselerasiData=akselerasiData.replace("]","");
237
238                 splitAkselerasi=akselerasiData.split(",")
239
240                 tempX=[float(splitAkselerasi[0])]
241                 tempY=[float(splitAkselerasi[1])]
242                 tempZ=[float(splitAkselerasi[2])]
243
244             else:
245                 tempTime.append(currData['timeStamp'])
246                 tempSuhu.append(currData['suhu'])
247                 tempKelembapan.append(currData['kelembapan'])
248                 tempTekanan.append(currData['tekanan'])
249
250                 akselerasiData=currData['akselerasi']
251                 akselerasiData=akselerasiData.replace("[","");
252                 akselerasiData=akselerasiData.replace("]","");
253
254                 splitAkselerasi=akselerasiData.split(",")
255
256                 tempX.append(float(splitAkselerasi[0]))
257                 tempY.append(float(splitAkselerasi[1]))
258                 tempZ.append(float(splitAkselerasi[2]))
259
260
261             if(len(tempSuhu)>0):
262                 cell={}
263                 lengthData=len(tempSuhu)
264                 cell['timeStamp']= tempTime[int(lengthData/2)]
265                 cell['suhu']= round(sum(tempSuhu)/len(tempSuhu) ,2)
266                 cell['kelembapan']=round( sum(tempKelembapan)/len(tempKelembapan) ,2)
267                 cell['tekanan']= round( sum(tempTekanan)/len(tempTekanan),2)
268                 cell['akselerasi']={
269                     "x":round(sum(tempX)/len(tempX),2),
270                     "y":round(sum(tempY)/len(tempY),2),
271                     "z":round(sum(tempZ)/len(tempZ),2)
272                 }

```

```

273         output.append(cell)
274
275     return output
276
277 def medianData(self,sensingData,interval):
278
279     currTime=sensingData[0]['timeStamp'];
280     currIntervalLeft=interval
281     output=[]
282     tempTime=[sensingData[0]['timeStamp']]
283     tempSuhu=[sensingData[0]['suhu']]
284     tempKelembapan=[sensingData[0]['kelembapan']]
285     tempTekanan=[sensingData[0]['tekanan']]
286
287     akselerasiData=sensingData[0]['akselerasi']
288     akselerasiData=akselerasiData.replace("[","")
289     akselerasiData=akselerasiData.replace("]","");
290
291     splitAkselerasi=akselerasiData.split(",")
292     tempX=[float(splitAkselerasi[0])]
293     tempY=[float(splitAkselerasi[1])]
294     tempZ=[float(splitAkselerasi[2])]
295
296     # tempAkselerasi=[sensingData[0]['alse;erasi']]
297     for i in range(1,len(sensingData)):
298         currData=sensingData[i]
299
300         diffInterval=(currData['timeStamp']-currTime).total_seconds()
301         currIntervalLeft-=diffInterval
302
303         currTime=currData['timeStamp']
304
305         if(currIntervalLeft <=0 ): #kalau habis
306
307             currIntervalLeft=interval
308             cell={}
309             medPos=(int)(round((len(tempSuhu)+1)/2,0)) #kalau dia .5 maka ++
310
311             medSuhu=tempSuhu[medPos]
312             medKelembapan=tempKelembapan[medPos]
313             medTekanan=tempTekanan[medPos]
314             medX=tempX[medPos]
315             medY=tempY[medPos]
316             medZ=tempZ[medPos]
317
318             if(len(tempSuhu)%2==1):
319                 medPos=(int)(len(tempSuhu)/2)
320
321                 medSuhu+=tempSuhu[medPos]
322                 medKelembapan+=tempKelembapan[medPos]
323                 medTekanan+=tempTekanan[medPos]
324                 medX+=tempX[medPos]
325                 medY+=tempY[medPos]
326                 medZ+=tempZ[medPos]
327
328             cell['timeStamp']= tempTime[medPos]
329             cell['suhu']= medSuhu
330             cell['kelembapan']=medKelembapan
331             cell['tekanan']= medTekanan
332             cell['akselerasi']={
333                 "x":medX,
334                 "y":medY,
335                 "z":medZ
336             }
337
338             output.append(cell)
339
340             tempTime=[currData['timeStamp']]
341             tempSuhu=[currData['suhu']]
342             tempKelembapan=[currData['kelembapan']]
343             tempTekanan=[currData['tekanan']]
344
345             akselerasiData=currData['akselerasi']
346             akselerasiData=akselerasiData.replace("[","");
347             akselerasiData=akselerasiData.replace("]","");
348
349             splitAkselerasi=akselerasiData.split(",")
350
351             tempX=[float(splitAkselerasi[0])]
352             tempY=[float(splitAkselerasi[1])]
353             tempZ=[float(splitAkselerasi[2])]
354
355             else:
356                 tempTime.append(currData['timeStamp'])
357                 tempSuhu.append(currData['suhu'])
358                 tempKelembapan.append(currData['kelembapan'])
359                 tempTekanan.append(currData['tekanan'])
360
361                 akselerasiData=currData['akselerasi']
362                 akselerasiData=akselerasiData.replace("[","");
363                 akselerasiData=akselerasiData.replace("]","");
364
365                 splitAkselerasi=akselerasiData.split(",")
366
367                 tempX.append(float(splitAkselerasi[0]))
368                 tempY.append(float(splitAkselerasi[1]))
369
370
371

```

```

372         tempZ.append(float(splitAkselerasi[2]))
373
374
375     if(len(tempSuhu)>0):
376         medSuhu=tempSuhu[medPos]
377         medKelembapan=tempKelembapan[medPos]
378         medTekanan=tempTekanan[medPos]
379         medX=tempX[medPos]
380         medY=tempY[medPos]
381         medZ=tempZ[medPos]
382
383     if(len(tempSuhu)%2==1):
384         medPos=int(len(tempSuhu)/2)
385
386         medSuhu+=tempSuhu[medPos]
387         medKelembapan+=tempKelembapan[medPos]
388         medTekanan+=tempTekanan[medPos]
389         medX+=tempX[medPos]
390         medY+=tempY[medPos]
391         medZ+=tempZ[medPos]
392
393
394     cell['timeStamp']= tempTime[medPos]
395     cell['suhu']= medSuhu
396     cell['kelembapan']=medKelembapan
397     cell['tekanan']= medTekanan
398     cell['akselerasi']={
399         "x":medX,
400         "y":medY,
401         "z":medZ
402     }
403
404     output.append(cell)
405
406
407     return output
408
409
410
411
412 def rawData(self,sensingData,interval):
413
414     currTime=sensingData[0]['timeStamp'];
415     currIntervalLeft=interval
416     output=[]
417     tempTime=[sensingData[0]['timeStamp']]
418     tempSuhu=[sensingData[0]['suhu']]
419     tempKelembapan=[sensingData[0]['kelembapan']]
420     tempTekanan=[sensingData[0]['tekanan']]
421
422     akselerasiData=sensingData[0]['akselerasi']
423     akselerasiData=akselerasiData.replace("[","")
424     akselerasiData=akselerasiData.replace("]","");
425
426     splitAkselerasi=akselerasiData.split(",")
427     tempX=[float(splitAkselerasi[0])]
428     tempY=[float(splitAkselerasi[1])]
429     tempZ=[float(splitAkselerasi[2])]
430
431     # tempAkselerasi=[sensingData[0]['alse;erasi']]
432     for i in range(1,len(sensingData)):
433         currData=sensingData[i]
434
435         diffInterval=(currData['timeStamp']-currTime).total_seconds()
436         currIntervalLeft-=diffInterval
437
438         currTime=currData['timeStamp']
439
440         if(currIntervalLeft <=0 ): #kalau habis
441
442             currIntervalLeft=interval
443             cell={}
444             lengthData=len(tempSuhu)
445             cell['timeStamp']= str(tempTime[int(lengthData/2)])
446             cell['suhu']= tempSuhu
447             cell['kelembapan']=tempKelembapan
448             cell['tekanan']= tempTekanan
449             cell['akselerasi']={
450                 "x":tempX,
451                 "y":tempY,
452                 "z":tempZ
453             }
454
455
456             output.append(cell)
457             tempTime=[currData['timeStamp']]
458             tempSuhu=[currData['suhu']]
459             tempKelembapan=[currData['kelembapan']]
460             tempTekanan=[currData['tekanan']]
461
462             akselerasiData=currData['akselerasi']
463             akselerasiData=akselerasiData.replace("[","");
464             akselerasiData=akselerasiData.replace("]", "")
465
466             splitAkselerasi=akselerasiData.split(",")
467
468             tempX=[float(splitAkselerasi[0])]
469             tempY=[float(splitAkselerasi[1])]
470             tempZ=[float(splitAkselerasi[2])]
```

```

471
472     else:
473         tempTime.append(currData['timeStamp'])
474         tempSuhu.append(currData['suhu'])
475         tempKelembapan.append(currData['kelembapan'])
476         tempTekanan.append(currData['tekanan'])
477
478         akselerasiData=currData['akselerasi']
479         akselerasiData=akselerasiData.replace(",","")
480         akselerasiData=akselerasiData.replace("[",",")
481
482         splitAkselerasi=akselerasiData.split(",")
483
484         tempX.append(float(splitAkselerasi[0]))
485         tempY.append(float(splitAkselerasi[1]))
486         tempZ.append(float(splitAkselerasi[2]))
487
488     if(len(tempSuhu)>0):
489         lengthData=len(tempSuhu)
490         cell={}
491         cell['timeStamp']= str(tempTime[int(lengthData/2)])
492         cell['suhu']= tempSuhu
493         cell['kelembapan']=tempKelembapan
494         cell['tekanan']= tempTekanan
495         cell['akselerasi']={
496             "x":tempX,
497             "y":tempY,
498             "z":tempZ
499         }
500
501         output.append(cell)
502
503
504     return output
505
506 def statDataQuartile(self,sensingData):
507     result={}
508     sensors=['suhu','kelembapan','tekanan']
509     data={"suhu":[],"kelembapan":[],"tekanan":[]}
510     for i in range(0,len(sensingData)):
511         for sensor in sensors:
512             data[sensor].append(sensingData[i][sensor])
513
514     for sensor in sensors:
515         dataSensor=data[sensor]
516         q3 = np.percentile(dataSensor, 75)
517         q1 = np.percentile(dataSensor, 25)
518         jumlahQ3=0;
519         jumlahQ1=0;
520         for i in range(0,len(data)):
521             if(dataSensor[i]>=q3):
522                 jumlahQ3+=1
523             elif(dataSensor[i]<=q1):
524                 jumlahQ1+=1
525         jumlahQ2=len(dataSensor)-(jumlahQ3+jumlahQ1)
526         result[sensor]={
527             "Q3":q3
528             , "Q1":q1
529             , "numQ1":jumlahQ1
530             , "numQ2":jumlahQ2
531             , "numQ3":jumlahQ3
532         }
533     return result
534
535 def count_range(self, data, start, end):
536     count = 0
537     for num in data:
538         if start <= num < end:
539             count += 1
540     return count
541
542 def statDataRange(self, sensingData, span):
543     result={"suhu":[],"kelembapan":[],"tekanan":[]}
544
545     sensors=['suhu','kelembapan','tekanan']
546     data={"suhu":[],"kelembapan":[],"tekanan":[]}
547     for i in range(0,len(sensingData)):
548         for sensor in sensors:
549             data[sensor].append(sensingData[i][sensor])
550
551     jumlahData=(100/span)
552     print (jumlahData)
553     for sensor in sensors:
554         startNum=1
555         for p in range(int(jumlahData)):
556             end=startNum+span
557             jumlah=self.count_range( data[sensor], startNum,end)
558
559             result[sensor].append(jumlah)
560             startNum=startNum+span
561
562     return result
563
564
565 def getRealTime(self, identifier):
566     selectedWSN=self.__searchWSN(identifier)
567     return selectedWSN.getLastData()
568
569

```

```

570|
571| def insertNewWSN(self, dbController, identifier, nama, indoor, interval, tipeSensor, idBS):
572|     import random
573|     token=""
574|     availableCharacter="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"
575|     numAvailableCharacter=len(availableCharacter)-1
576|     for i in range(0,10):
577|         num=random.randint(0,numAvailableCharacter)
578|         token+=availableCharacter[num]
579|     result=dbController.insertNodeSensor(identifier, nama, token, indoor, interval, idBS)
580|
581|     if(not result):
582|         return False, False
583|
584|     dbController.insertTipe(tipeSensor, identifier)
585|
586|     latitude=result[0]['latitude']
587|     longitude=result[0]['longitude']
588|     kota=result[0]['namakota']
589|
590|     insertedWSN=WSN(identifier=identifier, sensorType=tipeSensor,
591|                       interval=interval, latitude=latitude,
592|                       longitude=longitude, kota=kota,
593|                       queue=None, token=token)
594|
595|     self.allWSN.append(insertedWSN)
596|     return True, token

```

Kode E.9: API - WSN.py

```

1 from datetime import datetime
2 class WSN:
3     def __init__(self, identifier:str, token:str,
4                  ,sensorType:list, interval:int
5                  ,latitude:str ,longitude:str
6                  , kota:str, queue:dict):
7
8         self.identifier=identifier
9         self.token=token
10
11        self.sensorType=sensorType
12        self.interval=interval;
13
14        self.latitude=latitude
15        self.longitude=longitude
16
17        self.kota=kota
18        self.queue=queue
19        self.offlineData={"time": "1970-01-01_00:00:00", "idBS":self.identifier, "result": {"kelembapan":None, "tekanan":None, "suhu":None, "akselerasi":None}}
20
21        self.lastData=self.offlineData
22
23
24    def setSensingTable(self, sensingTable):
25        self.sensingTable=sensingTable
26        print(self.sensingTable)
27
28    def setQueue(self, queue):
29        self.queue=queue
30
31    def addSensingTable(self, nameNewTable):
32        self.sensingTable.append(nameNewTable)
33
34    def setLastData(self, data):
35        timeLastData=self.lastData['time']
36        newDataTable=data['time']
37
38        lastTime = datetime.strptime(timeLastData,"%Y-%m-%d_%H:%M:%S")
39        newTime = datetime.strptime(newDataTable,"%Y-%m-%d_%H:%M:%S")
40
41        if(newTime>lastTime):
42            self.lastData=data;
43
44
45    def setInterval(self,interval):
46        self.interval=int(interval);
47
48
49    def getIdentifier(self):
50        return self.identifier
51
52    def getToken(self):
53        return self.token
54
55    def getSensorType(self):
56        return self.sensorType
57
58    def getSensingTable(self):
59        return self.sensingTable
60
61    def getLangitude(self):
62        return self.latitude
63
64    def getLongitude(self):
65        return self.longitude
66
67

```

```
68|     def getKota(self):
69|         return self.kota
70|
71|     def getQueue(self):
72|         return self.queue
73|
74|     def getInterval(self):
75|         return self.interval
76|
77|     def getLastData(self):
78|         timeLastData=self.lastData['time']
79|         lastTime = datetime.strptime(timeLastData,"%Y-%m-%d %H:%M:%S")
80|         timeNow= datetime.now()
81|         deltaTime=(timeNow-lastTime).seconds
82|         deltaTime*=1000
83|         if deltaTime> (self.interval*5):
84|             return self.offlineData
85|         else:
86|             return self.lastData
```

LAMPIRAN F

KODE PROGRAM WEBSITE

Kode F.1: Website - AdminController.php

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Routing\Controller;
7 use App\Models\API;
8
9 use App\Http\Controllers\GraphController;
10
11 class AdminController extends Controller
12 {
13     public function ganti(Request $request){
14         $bs=$request->post('bs');
15         $value=$request->post('newInterval');
16         $value*=1000;
17
18         $map = array();
19         $map['node']=$bs;
20         $map['command']="setInterval:".$value;
21         $map['token']=$request->session()->get('token', '');
22
23         $result=API::POST('update',$map);
24         return redirect('/admin');
25     }
26
27
28     public function kota(Request $request){
29         $nama=$request->post('nama');
30         // $hour=$request->post('hour');
31         // $minutes=$request->post('minutes');
32
33         $map = array();
34         $map['nama']=$nama;
35         $map['token']=$request->session()->get('token', '');
36
37         $result=API::POST('kota',$map);
38         return redirect('/admin');
39     }
40
41     public function addWSN(Request $request){
42
43         $sensor=array();
44
45         if($request->post('suhu')=="on"){
46             array_push($sensor, 'Suhu');
47         }
48
49         if($request->post('tekanan')=="on"){
50             array_push($sensor, 'Kelembapan');
51         }
52
53         if($request->post('kelembapan')=="on"){
54             array_push($sensor, 'Tekanan');
55         }
56
57         if($request->post('akselerasi')=="on"){
58             array_push($sensor, 'Akselerasi');
59         }
59
60
61         $indoors=($request->post('indoors')=="on";
62         $map = array();
63         $map['identifier']=$request->post('identifier');
64         $map['idBS']=$request->post('bs'); //del
65         $map['indoor']=$indoors; //del
66         $map['nama']=$request->post('nama');
67         $map['interval']=($request->post('interval')*1000);
68         $map['token']=$request->session()->get('token', '');
69         $map['tipeSensor']=$sensor;
70
71         $result=API::POST('node',$map)['result'];
72
73         if(!$result){
74             return redirect('/admin?wrongIdentifier=1');
```

```

76    }else{
77        $key=$result['key'];
78        return redirect("/admin?key=$key");
79    }
80}
81}
82}
83
84 public function addBase(Request $request){
85     $map = array();
86     $map['nama']=$request->post('nama');
87     $map['latitude']=$request->post('latitude');
88     $map['longitude']=$request->post('longitude');
89     $map['idKota']=$request->post('kota');
90     $map['token']=$request->session()->get('token'), '';
91
92     $idLokasi=(API::POST('bs',$map)['result'])['id'];
93
94     return redirect('/admin');
95 }
96
97
98
99 public function adminView(Request $request){
100    $mainPage=new GraphController();
101    return view('layout')
102        ->with('semuaKota',$mainPage->getCity())
103        ->with("semuaLokasi",$mainPage->getLocation())
104        ->with("base",$mainPage->getBS())
105        ->with('menu',true)
106        ->with('location',false)
107        ->with('admin',true)
108        ->with('page','/admin');
109
110}
111

```

Kode F.2: Website - UserController.php

```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Routing\Controller;
7 use App\Models\API;
8 use Illuminate\Support\Facades\Session;
9
10
11 use App\Http\Controllers\GraphController;
12
13 class UserController extends Controller
14 {
15
16     public function __construct()
17     {
18         // Session::startSession();
19
20     }
21
22
23     public function login(Request $request){
24         $ttoken=$request->post('_token');
25         $username=$request->post('username');
26         $password=$request->post('password');
27         $keepSignIn=$request->post('signIn');
28
29         $map = array();
30         $map['username']=$username;
31         $map['password']=$password;
32         $map['token']=$ttoken;
33
34
35         $result=API::POST('login',$map);
36
37
38
39
40         if($result['result']==0){
41             Session::put('username', $username);
42             Session::put('token', $ttoken);
43             Session::put('role', 0);
44             return redirect( 'user' );
45         }
46         elseif($result['result']==1) {
47             Session::put('username', $username);
48             Session::put('token', $ttoken);
49             Session::put('role', 1);
50             return redirect('admin');
51         }
52         else{
53             return redirect('/login?wrong=1');
54         }
55
56     }
57
58     public function logout(){
59

```

```

60     Session::flush();
61     return redirect('login');
62 }
63
64 public function signUp(Request $request){
65     $token=$request->post('_token');
66     $username=$request->post('username');
67     $email=$request->post('email');
68     $password=$request->post('password');
69
70     $map = array();
71     $map['username']=$username;
72     $map['password']=$password;
73     $map['email']=$email;
74
75     $result=API::POST('signup',$map);
76     if($result['result']){
77         return redirect('/main');
78     }
79     else{
80         return redirect('/signUP?wrong=1');
81     }
82 }
83
84
85 public function viewUser(Request $request){
86     $mainPage=new GraphController();
87     $username=$request->session()->get('username');
88     return view('layout')
89     ->with('semuaKota',$mainPage->getCity())
90     ->with("semuaLokasi",$mainPage->getLocation())
91     ->with('menu',true)
92     ->with('api', API::getURL())
93     ->with('page','graphSelection');
94 }
95
96 public function realTimeCont(Request $request){
97     $mainPage=new GraphController();
98     $username=$request->session()->get('username');
99     return view('layout')
100    ->with('semuaKota',$mainPage->getCity())
101    ->with("semuaLokasi",$mainPage->getLocation())
102    ->with('menu',true)
103    ->with('page','live')
104    ->with('api', API::getURL());
105 }
106
107 public function viewTable(Request $request){
108     $mainPage=new GraphController();
109     $username=$request->session()->get('username');
110     return view('layout')
111     ->with('semuaKota',$mainPage->getCity())
112     ->with("semuaLokasi",$mainPage->getLocation())
113     ->with('menu',true)
114     ->with('page','tabel')
115     ->with('api', API::getURL());
116 }
117
118 public function viewStatistik(Request $request){
119     $mainPage=new GraphController();
120     $username=$request->session()->get('username');
121     return view('layout')
122     ->with('semuaKota',$mainPage->getCity())
123     ->with("semuaLokasi",$mainPage->getLocation())
124     ->with('menu',true)
125     ->with('page','statistik')
126     ->with('api', API::getURL());
127 }
128
129 public function viewUserProfile(Request $request ){
130
131     return view('layout')
132     ->with('menu',true)
133     ->with('location',false)
134     ->with('page','userSetting');
135 }
136
137 }
138 }
```

Kode F.3: Website - API.php

```

1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 use Illuminate\Http\Request;
9 use Illuminate\Support\Facades\Http;
10
11 class API extends Model
12 {
13     use HasFactory;
14     // protected static $apiAddreses="http://192.168.101.95:5000";
15     protected static $apiAddreses="http://localhost:5000";
16 }
```

```

17 public static function GET($endpoint, $parameter){
18     $url=API::$apiAddreses;
19     $request = Http::get("$url/$endpoint?$parameter");
20     $response=$request->json();
21
22     return $response;
23 }
24
25 public static function getURL(){
26     return API::$apiAddreses;
27 }
28
29 public static function POST($endpoint, $parameter){
30     $url=API::$apiAddreses;
31     $request = Http::post("$url/$endpoint", $parameter);
32     $response=$request->json();
33     return $response;
34 }
35
36
37
38
39
40
41
42
43 }
44
45
46 ?>
```

Kode F.4: Website - web.php

```

1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\UserController;
5 use App\Http\Controllers\AdminController;
6 use App\Http\Controllers\GraphController;
7
8 /*
9 | -----
10 | Web Routes
11 | -----
12 |
13 | Here is where you can register web routes for your application. These
14 | routes are loaded by the RouteServiceProvider within a group which
15 | contains the "web" middleware group. Now create something great!
16 |
17 */
18
19 Route::get('/', function () {
20     return redirect("login");
21 });
22
23 Route::get('/login', function () {
24     return view('login');
25 });
26
27 Route::get('/signUP', function () {
28     return view('signup');
29 });
30
31
32 Route::get('/main',[ UserController::class , 'viewUser']);
33
34 Route::get('/realTime',[ UserController::class , 'realTimeCont']);
35
36 Route::get('user',[UserController::class,'viewUserProfile']);
37
38 Route::get('/admin', [AdminController::class, "adminView"]);
39
40
41 Route::get('/table',[ UserController::class , 'viewTable']);
42
43 Route::get('/statistic',[ UserController::class , 'viewStatistik']);
44
45 /*
46 | -----
47 | POST ROUTE
48 | -----
49 |
50 */
51
52 Route::post('/masuk', [UserController::class, 'login']);
53
54 Route::get('/keluar', [UserController::class, 'logout']);
55
56 Route::post('/daftar', [UserController::class, 'signup']);
57
58 Route::post('/data', [GraphController::class, 'data']);
59
60 Route::post('/ganti', [AdminController::class, 'ganti']);
61
62 Route::post('/kota', [AdminController::class, 'kota']);
63
64 Route::post('/sensorBaru', [AdminController::class, 'addWSN']);
```

```
66| Route::post('/baseBaru', [AdminController::class, 'addBase']);
```

Kode F.5: Website - accelerationInfo.blade.php

```
1 <br>
2 <div class="alert alert-info" role="alert" id="akelserasiInfo" style="display:none">
3 *Info Akselerasi:
4 <br>
5 X+ menandakan percepatan ke Timur
6 <br>
7 Y+ menandakan percepatan ke Utara
8 <br>
9 Z- menandakan percepatan ke Bawah
10</div>
```

Kode F.6: Website - admin.blade.php

```
1
2
3
4
5
6 <?php
7 use Illuminate\Support\Facades\Input; ?>
8
9
10 @if (!session()->has('role'))
11     <script>
12         // redirect tidak dilakukan langsung di dalam file Blade karena Blade hanya bertanggung jawab untuk merender tampilan
13         window.location.href = "{{url('/login')}}";
14     </script>
15 @else
16     @if (session('role')==0)
17         <script>
18             // redirect tidak dilakukan langsung di dalam file Blade karena Blade hanya bertanggung jawab untuk merender tampilan
19             window.location.href = "{{url('/user')}}";
20         </script>
21     @endif
22
23 @endif
24
25
26
27
28 <link rel="stylesheet" href="{{'css/location.css'}}">
29 <link rel="stylesheet" href="{{'css/admin.css'}}">
30 <script src="{{asset('js/mainAdmin.js')}}" defer></script>
31
32 <br>
33 <div class="container" id="workSheet">
34
35
36     @if (app('request')->has('wrongIdentifier')==1)
37     <br>
38     <div class="alert alert-danger" role="alert">
39         Node baru gagal disimpan, mohon coba lagi dengan identifier yang berbeda
40     </div>
41     @endif
42
43
44
45     @if (app('request')->has('key'))
46     <br>
47     <div class="alert alert-info" role="alert">
48         Node baru berhasil disimpan. Key: {{app('request')->get('key')}}
49     </div>
50     @endif
51
52
53     {{-- bagian update Config --}}
54     <a class="btn btn-primary" data-bs-toggle="collapse" href="#updateInterval" role="button" aria-expanded="false" aria-controls="collapseExample">
55         Update Interval
56     </a>
57     <a class="btn btn-primary" data-bs-toggle="collapse" href="#tambahKota" role="button" aria-expanded="false" aria-controls="collapseExample">
58         Tambah Kota
59     </a>
60     <a class="btn btn-primary" data-bs-toggle="collapse" href="#tambahBase" role="button" aria-expanded="false" aria-controls="collapseExample">
61         Tambah Base Station
62     </a>
63     <a class="btn btn-primary" data-bs-toggle="collapse" href="#tambahSensor" role="button" aria-expanded="false" aria-controls="collapseExample">
64         Tambah Node Sensor
65     </a>
66
67     @include('adminItem/updateInterval')
68
69
70     @include('adminItem/insertKota')
71
72     @include("adminItem/newBase")
73
```

```

75| @include('adminItem/newSensor')
76|
77|
78|
79|
80|
81{{-- {{\$request->input('wrongIdentifier')}} --}}
82
83</div>

```

Kode F.7: Website - header.blade.php

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Sensor Vision</title>
8
9     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLhlTQ8iRABdZLl603oVMWSktQOo6b7In1zL3/Jr59b6EGGoI1aFkw7cmDA6j6qD" crossorigin="anonymous">
10    <script defer src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UYJoMp4YLEuNSfAp+JcXn/tWtIxVXM" crossorigin="anonymous"></script>
11
12    <link rel="icon" href="assets/img/logo.png" type="image/icon">
13    <link rel="stylesheet" href="{{'css/layout.css'}}">
14
15
16</head>
17<body style="height:100%">
18    <div class="d-flex flex-row_menuAtas">
19        @php
20            if($menu){
21                echo "<button type='button' class='btn btn-link mx-2' data-bs-toggle='collapse' href='#navigation' aria-expanded='false' aria-controls='collapseExample'>
22                    <img src='assets/img/menu.png' height='25vh;' />
23                </button>";
24            }
25        @endphp
26
27        
28        <span id="title" class="align-self-center" >Sensor Vision</span>
29    </div>

```

Kode F.8: Website - layout.blad.php

```

1 @php
2     use Illuminate\Support\Facades\Session;
3 @endphp
4
5
6
7 @include('header',['menu'=> $menu ])
8 @include("navigation")
9
10
11 @if (Route::is('/login') || Route::is('/signup') )
12
13 @else
14     @if (!session()->has('token'))
15         <script>
16             // redirect tidak dilakukan langsung di dalam file Blade karena Blade hanya bertanggung jawab untuk merender tampilan
17             window.location.href = "{{url('/login')}}";
18         </script>
19     @endif
20
21
22     <div class="col-md-9_ms-sm-auto_col-lg-10_px-md-4">
23         @include($page)
24         {{-- @include('live') --}}
25     </div>
26     @include("footer")
27
28 @endif
29
30
31
32 @endif

```

Kode F.9: Website - live.blade.php

```

1 <script src="https://cdn.plot.ly/plotly-2.23.2.min.js" charset="utf-8"></script>
2 <script src="{{asset('js/live.js')}}" defer></script>
3
4 <link rel="stylesheet" href="{{'css/livePage.css'}}">
5
6 <script>
7 const urlAPI='{{$api}}'
8
9
10 </script>

```

```

12
13 <br>
14
15 <div class="row">
16   <div class="col-10">
17     <Select class="form-select" id="kotaSelection">
18       <option value="-1">Filter berdasarkan kota</option>
19       @foreach ($semuaKota as $kota)
20         <option value="{{ $kota['id'] }}>{{ $kota['nama'] }}</option>
21       @endforeach
22     </Select>
23
24   <div class="card_my-1">
25     <div class="card-body">
26       <h5 class="card-title">Pilih Diagram</h5>
27     </div>
28     <div class="card-text_border_liveData_align-text-bottom_liveSelection_text-center">
29       <p class="liveDataTitle">Kelembapan (%) </p>
30       <select class="mt-2" name="b" id="kelembapanSelection">
31         <option value="simpleText" >Teks Sederhana</option>
32         <option value="lines" >Diagram Garis</option>
33         <option value="scatter" >Diagram Area</option>
34         {{-- <option value="bar" > Diagram Batang --}}
35         <option value="gauge">Gauge </option>
36       </select>
37     </div>
38   <div class="card-text_border_liveData_align-text-bottom_liveSelection_text-center">
39     <p class="liveDataTitle">Suhu (°C) </p>
40     <select class="mt-2" name="b" id="suhuSelection">
41       <option value="simpleText" >Teks Sederhana</option>
42       <option value="lines" >Line Chart</option>
43       {{-- <option value="bar" > Diagram Batang --}}
44       <option value="scatter" >Area Chart</option>
45     </select>
46   </div>
47   <div class="card-text_border_liveData_align-text-bottom_liveSelection_text-center" >
48     <p class="liveDataTitle">Tekanan (kPa) </p>
49     <select class="mt-2" name="b" id="tekananSelection">
50       <option value="simpleText" >Teks Sederhana</option>
51       <option value="lines">Line Chart</option>
52       {{-- <option value="bar" > Diagram Batang --}}
53       <option value="scatter" >Area Chart</option>
54     </select>
55   </div>
56   <div class="card-text_border_liveData_align-text-bottom_liveSelection_text-center">
57     <p class="liveDataTitle_text-center">Akselerasi (x,y,z)</p>
58     <select class="mt-2" name="b" id="akselerasiSelection">
59       <option value="simpleText" >Teks Sederhana</option>
60       <option value="lines">Line Chart</option>
61       {{-- <option value="bar" > Diagram Batang --}}
62       <option value="scatter" >Area Chart</option>
63       {{-- <option value="scatter3d" >Area Chart --}}
64     </select>
65   </div>
66   </div>
67 </div>
68
69
70
71
72
73 @foreach ($semuaLokasi as $lokasi)
74   <div class="cardCollapse_my-1" id="{{ $lokasi['identifier'] }}">
75     <div class="card-body">
76       <h5 class="card-title">{{ $lokasi['namalokasi'] }}</h5>
77       <h6 id="timeText{{ $lokasi['identifier'] }}" class="card-text"></h6>
78     </div>
79     <div id="myPlotkelembapan{{ $lokasi['identifier'] }}" class="" style="display:inline-block"></div>
80     <div class="card-text_border_liveData_align-text-bottom">
81       <p class="liveDataTitle_text-center">Kelembapan </p>
82       <div class="text-center_text-center">
83         <p id="kelembapanText{{ $lokasi['identifier'] }}" class="liveDataText"></p>
84         <p class="liveDataText" hidden>%</p>
85       </div>
86     </div>
87
88
89     <div id="myPlotsuhu{{ $lokasi['identifier'] }}" style="display:inline-block"></div>
90     <div class="card-text_border_liveData_align-text-bottom">
91       <p class="liveDataTitle_text-center">Suhu </p>
92       <div class="text-center_text-center">
93         <p id="suhuText{{ $lokasi['identifier'] }}" class="liveDataText"></p>
94         <p class="liveDataText" hidden>°C</p>
95       </div>
96     </div>
97
98
99     <div id="myPlottekanan{{ $lokasi['identifier'] }}" style="display:inline-block"></div>
100    <div class="card-text_border_liveData_align-text-bottom" >
101      <p class="liveDataTitle_text-center">Tekanan </p>
102      <div class="text-center">
103        <p id="tekananText{{ $lokasi['identifier'] }}" class="liveDataText"></p>
104        <p class="liveDataText" hidden>kPa</p>
105      </div>
106    </div>
107
108
109    <div id="myPlotakselerasi{{ $lokasi['identifier'] }}" style="display:inline-block"></div>
110    <div class="card-text_border_liveData_align-text-bottom">
```

```

111     <p class="liveDataTitle_text-center">Akselerasi (x,y,z)</p>
112     <div class="text-center">
113         <p id="akselerasiText{$lokasi['identifier']}><span class="liveDataText" hidden>g</span></p>
114     </div>
115   </div>
116
117
118
119
120     </div>
121   </div>
122 </div>
123
124     @endforeach
125   </div>
126
127 <div class="col-2" id="lokSelect">
128   <h5>Pilih Lokasi</h5>
129   @foreach ($semuaLokasi as $lokasi)
130     <div class="optLokasi" city="{{$lokasi['idKota']}}" >
131       {{-- <input data-bs-toggle="collapse" href="#{{$lokasi['identifier']}}" aria-expanded="false" aria-controls="collapseExample" class="form-check-input" type="checkbox" name="{{$lokasi['identifier']}}" id="suhuCheck" onclick="initNewLive('{{$lokasi['identifier']}}',{{$lokasi['interval']}})" >
132
133       <label data-bs-toggle="collapse" href="#{{$lokasi['identifier']}}" aria-expanded="false" aria-controls="collapseExample" class="form-check-label" for="{{$lokasi['identifier']}}" onclick="initNewLive('{{$lokasi['identifier']}}',{{$lokasi['interval']}})" >{{$lokasi['nama']}}</label> --}}
134
135
136
137       <input data-bs-toggle="collapse" href="#{{$lokasi['identifier']}}" aria-expanded="false" aria-controls="collapseExample" onclick="initNewLive('{{$lokasi['identifier']}}',{{$lokasi['interval']}})" type="checkbox" id="select{{$lokasi['identifier']}}" >
138       <label data-bs-toggle="collapse" href="#{{$lokasi['identifier']}}" aria-expanded="false" aria-controls="collapseExample" class="form-check-label" for="select{{$lokasi['identifier']}}" >{{$lokasi['namaLokasi']}}</label>
139
140       <br><br>
141     </div>
142
143     @endforeach
144   </div>
145
146
147
148
149
150
151
152
153
154
155
156
157 {{--
158   <div class="row">
159     <div class="col-12">
160       @foreach ($semuaLokasi as $lokasi)
161         <input class="form-check-input" type="checkbox" name="{{$lokasi['identifier']}}" id="suhuCheck" checked>
162         <label class="form-check-label" for="{{$lokasi['identifier']}}>{{$lokasi['nama']}}</label>
163
164
165     @endforeach
166   </div>
167 </div> --}}

```

Kode F.10: Website - login.blade.php

```

1 @include('header', ['menu' => false ])
2 <link rel="stylesheet" href="{{'css/login.css'}}">
3
4
5 <div class="d-flex justify-content-end" style="margin: _30px,_50px,_0px,_0px">
6
7   
8
9   <div class="card_login" style="width: _18rem;">
10    <div class="card-body">
11      <div class="text-center">
12        
13
14      <h3>MASUK</h3>
15      <form class="form-signin" action="/masuk" method="post">
16        @csrf
17        {{-- frame work laravel yang membantu mengenerate token --}}
18        <div class="form-group">
19          <label>Username</label>
20          <input type="text" class="form-control" id="email" aria-describedby="emailHelp" placeholder="Enter_username" name="username">
21
22        </div>
23        <div class="form-group_mt-2">
24          <label>Password</label>
25          <input type="password" class="form-control" id="password" placeholder="Password" name="password">
26        </div>
27
28        Tidak punya akun? <a href="/signUP">Buat akun</a>
29
30        <button type="submit" class="btn btn-primary_mt-3">Masuk</button>
31      </form>

```

```

32         </div>
33     </div>
34   </div>
35 @include("footer")
36

```

Kode F.11: Website - navigation.blade.php

```

1 <link rel="stylesheet" href="{{'css/navigation.css'}}">
2
3 <div class="container-fluid" >
4   <div class="row">
5
6     <div class="col-lg-2" id="navigation" style="display:none !important;">
7       <div class="pt-3" style="height: 92vh;">
8         <ul class="nav flex-column" style="list-style-type: none; padding-left: 0;">
9
10
11       @if (session('role') == 1)
12         <li class="nav-item">
13           <a class="nav-link active" aria-current="page" href="admin">
14             <span>
15               <img alt="Admin icon" data-original="#000000" style="vertical-align: middle; height: 1em; margin-right: 0.2em;"/>
16               Admin
17             </span>
18           </a>
19         </li>
20
21       @endif
22
23
24
25
26         <li class="nav-item">
27           <a class="nav-link active" aria-current="page" href="realTime">
28             <span>
29               <img alt="Real Time icon" data-original="#000000" style="vertical-align: middle; height: 1em; margin-right: 0.2em;"/>
30               Real Time
31             </span>
32           </a>
33         </li>
34
35
36         <li class="nav-item">
37           <a class="nav-link active" aria-current="page" href="main">
38             <span>
39               <img alt="Data History icon" data-original="#000000" style="vertical-align: middle; height: 1em; margin-right: 0.2em;"/>
40               Data History
41             </span>
42           </a>
43         </li>
44
45

```

```

46<li class="nav-item" id="navSensorAvailable" hidden>
47  <ul>
48    <li class="selected" box="suhu">
49      <a class="nav-link" aria-current="page" href="#" onclick="select(this);return false;" index=0>
50        <span>
51          <svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" version="1.1" id="Capa_1" x="0px" y="0px" viewBox="0_0_490_490" style="enable-background:new_0_0_490_490;" xml:space="preserve">
52            <g>
53              <g>
54                <path d="M262.096,490c-14.151,-0.27.883-2.774-40.818-8.245c-12.487-5.281-23.7-12.841-33.326-22.468
55                  -9.626-9.626-17.185-20.838-22.468-33.326c-5.47-12.933-8.244-26.666-8.244-40.816c-0.028,22.415,218.443,0,245.996,0h32.201c27.552,0,49.968,22.145,49.968,49.967v253.762c11.154,9.072,20.447,20.453,27.054,33.183c7.676,14.788,11.733,31.468,11.733,48.233c0,14.15-2.773,27.884-8.245,40.816c-5.282,12.488-12.84,23.7-22.467,33.326c-9.626,9.627-20.839,17.186-33.327,22.468c289.981,487.226,276.248,490,262.096,490z" M245.996,20.578c-16.205,-0.29.389,13.184-29.389,29.389v308.78c0,3.281-1.565,6.366-4.213,8.304c-21.649,15.84-34.574,41.283-34.574,68.06c0,46.47,37.886,84.276,84.277,84.276c46.471,0,84.277-37.887,84.277-84.276c0-26.777-12.925-52.221-34.575-68.06c-2.648-1.938-4.213-5.023-4.213-8.304v49.967c0-16.205-13.184-29.389-29.389-29.389H245.996z" fill="currentColor"/>
56            </g>
57            <path d="M283.943,325.025v-224.65c0-8.918-7.229-16.147-16.146-16.147h-11.402c-8.917,0-16.146,7.229-16.146,16.147v224.65c-24.561,8.927-42.105,32.475-42.105,60.121c0,35.32,28.632,63.953,63.952,63.953c35.32,0,63.953-28.633,63.953-63.953c326.048,357.5,368.505,333.952,283.943,325.025zM247.931,365.24c-4.783,4.243-7.878,9.827-8.949,16.148c-0.563,3.33-0.526,6.681,0.113,9.957c0.819,4.204-1.925,8.276-6.13,9.096c-0.501,0.099-1.0,1.46-1.492,0.146c-3.637,0.6-8.883-2.57-7.605-6.274c-0.997-5.117-1.057-10.337-0.18-15.514c1.643-9.707,6.598-18.643,13.95-25.164c3.205-2.842,8.107-2.549,10.95,0.656c251.429,357.496,251.136,362.397,247.931,365.24z" fill="currentColor"/>
58            <g>
59              <g>
60                <g>
61                  <path d="M157.262,75.302c0,4.658-3.776,8.436-8.436,8.436h-17.341c-4.66,0-8.437-3.778-8.437-8.436v-9.192c0-4.66,3.777-8.437,8.437-8.437h17.341c4.66,0,8.436,3.776,8.436,8.437V75.302z" fill="currentColor"/>
62                  <path d="M157.262,130.298c0,4.66-3.776,8.437-8.436,8.437h-17.341c-4.66,0-8.437-3.776-8.437-8.437v-9.192c0-4.66,3.777-8.437,8.437-8.437h17.341c4.66,0,8.436,3.776,8.436,8.437V130.298z" fill="currentColor"/>
63                  <path d="M157.262,185.294c0,4.66-3.776,8.437-8.436,8.437h-17.341c-4.66,0-8.437-3.776-8.437-8.437v-9.191c0-4.659,3.777-8.438,8.437-8.438h17.341c4.66,0,8.436,3.778,8.436,8.438V185.294z" fill="currentColor"/>
64                </g>
65                <path d="M157.262,240.291c0,4.659-3.776,8.438-8.436,8.438h-17.341c-4.66,0-8.437-3.778-8.437-8.438v-8.438h231.1c0-4.66,3.777-8.436,8.437-8.436h17.341c4.66,0,8.436,3.776,8.436,8.436V240.291z" fill="currentColor"/>
66                </g>
67                <path d="M157.262,295.287c0,4.661-3.776,8.438-8.436,8.438h-17.341c-4.66,0-8.437-3.776-8.437-8.438v-9.191c0-4.658,3.777-8.436,8.437-8.436h17.341c4.66,0,8.436,3.778,8.436,8.436V295.287z" fill="currentColor"/>
68              </g>
69            </svg>
70          </span>
71        Suhu
72      </a>
73    </li>
74
75    <li box="tekanan">
76      <a class="nav-link" aria-current="page" href="#" onclick="select(this);return false;" index=1>
77        <span>
78          <svg xmlns="http://www.w3.org/2000/svg" version="1.0" width="1280.000000pt" height="1280.000000pt"
79            viewBox="0_0_1280.000000_1280.000000" preserveAspectRatio="xMidYMid_meet">
80            <metadata fill="currentColor">
81              Created by potrace 1.15, written by Peter Selinger 2001-2017
82            </metadata>
83            <g transform="translate(0.000000,1280.000000)_scale(0.100000,-0.100000)" fill="currentColor" stroke="none">
84              <path d="M6095_12794_c-27_-2_-115_-8_-195_-14_-1870_-137_-3611_-1123_-4715_-2671_-1505_-2110_-1582_-4947_-193_-7132_-824_-1296_2092_-2261_3548_-2701_926_-279_1884_-347_2840_-200_761_117_1520_381_2180_759_1230_704_2185_1777_2734_3070_866_2040_607_4403_679_6204_-886_1241_-2181_2128_-3645_2496_-340_86_-633_135_-1005_171_-150_14_-756_-27_-870_182_m572_-984_c776_-36_1477_-217_2158_-556_560_-279_978_-580_1426_-1028_435_-435_743_-858_1008_-1386_325_-645_500_-1269_567_-2020_26_-288_14_-767_-27_-1095_-128_-1030_-548_-1992_-1224_-2805_-133_-160_-427_-458_-591_-601_-792_-688_-1757_-1133_-2774_-1278_-283_-41_-469_-53_-785_-53_-880_0_-1667_-187_-2450_582_-738_373_-1398_931_-1901_1605_-552_742_-909_1630_-1023_2550_-55_438_-55_912_0_1350_115_962_471_1810_1031_2560_163_219_293_368_522_596_241_240_338_325_576_504_863_649_1910_1026_2980_1075_238_10_281_10_507_02" fill="currentColor"/>
85              <path d="M6240_10996_l0_-214_-77_-6_c-431_-34_-740_-89_-1062_-191_-611_-193_-1123_-482_-1608_-908_l-130_-114_-155_148_-156_149_-116_-119_-111_-118_151_-151_152_-150_-75_-89_c-664_-778_-1037_-1786_-1039_-2805_10_-167_-52_-5_c-29_-3_-125_-11_-213_-17_-88_-6_-161_-112_-4_-5_21_-313_26_-319_3_-4_-96_0_205_8_110_8_205_13_213_10_7_-3_13_-20_13_-40_0_-63_60_-392_101_-552_80_-317_220_-686_361_-954_156_-298_322_-544_561_-831_82_-100_146_-184_140_-188_-25_-17_-313_-262_-313_-267_0_-11_204_-244_211_-242_4_-216_1365_303_197_165_364_305_372_310_11_9_-37_71_-233_305_-135_162_-281_344_-324_405_-372_523_-598_1102_-688_1765_-22_168_-26_712_-5_880_108_876_465_1631_1062_2244_-627_643_1406_1026_2320_1138_66_8_223_13_420_13_197_0_354_-5_420_-13_755_-93_1397_-361_1980_-826_-150_-120_433_-400_553_-547_423_-520_699_-1118_811_-1759_45_-260_51_-338_51_-680_-1_-280_-4_-347_-23_-481_-66_-466_-197_-882_-403_-1278_-151_-288_-279_-471_-609_-860_-192_-230_-239_-291_-228_-300_8_-5_175_-145_372_-310_196_-165_361_-301_365_-303_7_-2_211_231_-211_242_0_5_-288_251_-314_268_-5_3_44_71_110_150_203_244_309_388_426_578_245_399_417_806_529_1253_42_166_99_481_99_543_0_20_6_-37_13_40_8_3_103_-2_213_-10_109_-8_202_-12_205_-8_5_6_30_314_26_319_-1_1_-74_6_-162_12_-88_6_-183_14_-211_17_1_-51_5_-6_247_c-11_459_-76_838_-217_1262_-183_555_-439_1011_-824_1468_l-70_84_152_-150_151_151_-111_118_-110_119_-156_-149_-155_-148_-130_114_c-484_426_-997_715_-1608_908_-322_-102_-631_157_-1061_191_l-78_6_0_214_0_214_-160_0_-160_0_0_-214" fill="currentColor"/>
86              <path d="M8385_8348_c-165_-138_-605_-505_-978_-816_l_-679_-565_-56_27_c-31_15_-82_33_-112_41_-74_20_-246_19_-323_-1_-229_-59_-412_-243_-471_-472_-21_-80_-22_-229_-1_-317_8_-38_12_-71_7_-74_-5_-3_-151_-125_-326_-270_l_-317_-264_98_-101_c54_-56_156_-161_226_-233_70_-73_130_-133_132_-133_2_0_131_-140_286_311_198_218_287_309_298_305_118_-37_270_-48_371_-26_193_43_359_175_445_355_52_107_65_-166_65_285_0_103_-13_173_-47_249_l_-23_54_23_26_c121_131_1675_1850_1681_1859_4_7_6_12_4_11_-2_0_-" fill="currentColor"/>
87            </g>
88          </svg>
89        </span>
90      </a>
91    </li>

```

```

85           138_-113_-303_-251z" fill="currentColor"/>
86       </g>
87     </svg>
88   Tekanan
89 </a>
90 </li>
91 <li box="kelembapan">
92   <a class="nav-link" aria-current="page" href="#" onclick="select(this);return_false;" index=2>
93     <span>
94       <svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" version="1.1" id="Capa_1" x="0px" y="0px" viewBox="0_0_328.611_328.611" style="enable-background:new_0_0_328.611_328.611; xml:space="preserve">
95         <g>
96           <path d="M209.306,50.798c-2.452-3.337-7.147-4.055-10.485-1.602c-3.338,2.453-4.055,7.147-1.603,10.485
97             c-54.576,74.266,66.032,123.541,66.032,151.8c0,27.691-8.272,52.794-23.293,70.685c-17.519,20.866-42.972,31.446-75.651,31.446c-73.031,0-98.944-55.018-98.944-102.131c0-52.227,28.103-103.234,51.679-136.829c25.858-36.847,52.11-61.415,52.37-61.657c-3.035-2.819,3.209-7.565,0.39-10.6c-2.819-3.034-7.565-3.209-10.599-0.39c-1.11,1.031-27.497,25.698-54.254,63.765c-24.901,35.428-54.586,89.465-54.586,145.71c0,.31.062,9.673,.59.599,27.236,80.353c20.361,24.061,.50.345,36.779,86.708,36.779c36.794,0,66.926-12.726,87.139-36.801c17.286-20.588,26.806-49.117,26.806-80.33c278.25,156.216,240.758,93.597,209.306,50.798z" fill="currentColor"/>
98         <path d="M198.43,148.146l-95.162c95.162c-2.929,2.929-2.929,7.678,0,10.606c1.465,1.464,3.385,2.197,.5304,2.197c-.53,839-0.732,5.304-2.197c195.162c95.162c2.929-2.929-7.678,0-10.606c206.107,145.217,201.359,145.217,198.43,148.146z" fill="currentColor"/>
99         <path d="M191.965,207.899c-13.292,-0.24.106,10.814-24.106,24.106s10.814,24.106,24.106,24.106s24.106-10.814,24.106-24.106c-10.814,24.106-24.106,.5205.257,207.899,191.965,207.899c191.965,241.111c-5.021,0-9.106-4.085-9.106-9.106s4.085,9.106,9.106-9.106c-5.021,0-9.106-4.085-9.106-9.106s120.156,160.949,125.178,160.949z" fill="currentColor"/>
100        <path d="M125.178,194.162c13.292,0,24.106-10.814,24.106-24.106s-10.814,24.106-24.106-24.106s24.106-24.106,24.106-.5111.111,191.965,241.111c-111.111,191.965,241.111z" fill="currentColor"/>
101        <path d="M198.43,148.146l-95.162c95.162c-2.929,2.929-2.929,7.678,0,10.606c1.465,1.464,3.385,2.197,.5304,2.197c-.53,839-0.732,5.304-2.197c195.162c95.162c2.929-2.929-7.678,0-10.606c206.107,145.217,201.359,145.217,198.43,148.146z" fill="currentColor"/>
102        <path d="M198.43,148.146l-95.162c95.162c-2.929,2.929-2.929,7.678,0,10.606c1.465,1.464,3.385,2.197,.5304,2.197c-.53,839-0.732,5.304-2.197c195.162c95.162c2.929-2.929-7.678,0-10.606c206.107,145.217,201.359,145.217,198.43,148.146z" fill="currentColor"/>
103      </svg>
104    Kelembapan
105  </a>
106 </li>
107 <li box="akselerasi">
108   <a class="nav-link" aria-current="page" href="#" onclick="select(this);return_false;" index=3>
109     <span>
110       <svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" width="700pt" height="700pt" version="1.1" viewBox="0_0_700_700">
111         <defs fill="currentColor">
112           <symbol id="v" overflow="visible" fill="currentColor">
113             <path d="m18.766-1.125c-0.96875_0.5-1.9805_0.875-3.0312_1.125-1.043_0.25781-2.1367_0.39062-3.2812_0.39062-3.3984_0.6-0.0898-0.94531-8.0781-2.8438-1.9922-1.9062-2.9844-4.4844-2.9844-7.7344_0-3.2578_0.99219-5.8359_2.9844-7.7344_1.9883-1.9062_4.6797-2.8594_8.0781-2.8594_1.1445_0_2.2383_0.13281_0.39062_1.0568_0.25_0.625_0.625_0.3012_1.125v4.2188c-0.98047-0.65625-1.9453-1.1406-2.8906_1.4531-0.94922-0.3125-1.9492-0.46875-3-0.46875-1.875_0-3.3516_0.60547-4.4219_1.8125-1.0742_1.1992-1.6094_2.8555-1.6094_4.9688_0_2.1055_0.53516_3.7617_1.6094_4.9688_1.0703_1.1992_2.5469_1.7969_4.4219_1.7969_1.0508_0_2.0508-0.14844_3_0.45312_0.94531-0.3125_1.9102_0.80078_2.8906-1.4688z" fill="currentColor"/>
114           </symbol>
115           <symbol id="b" overflow="visible" fill="currentColor">
116             <path d="m13.734-11.141c-0.4375_0.19531-0.87109_0.34375-1.2969-0.4375-0.41797-0.10156-0.83984-0.15625-1.2656-0.15625-1.2617_0-2.2305_0.40625-2.9062_1.2188-0.67969_0.80469-1.0156_1.9531_1.0156_3.4531v7.0625h-1.3124.8906v2.5160c0.625_1.3124_1.3438-1.7266_2.1562-2.1875_0.82031-0.46875_1.8008_0.70312_2.9375-0.70312_0.16406_0_0.34375_0_0.011719_0.53125_0.03125_0.19531_0.011719_0.47656_0.039062_0.84375_0.078125z" fill="currentColor"/>
117           </symbol>
118           <symbol id="a" overflow="visible" fill="currentColor">
119             <path d="m17.641-7.7031v1.4062h-11.453c0.125_1.1484_0.53906_2.0078_1.25_2.5781_0.70703_0.57422_1.7031_0.85932_2.9844_0.85938_1.0312_0.2_0.082_0.14844_3.1562_1.45312_1.082-0.3125_2.1914_0.77344_3.3281_1.3906v3.7656c-1.1562_0.4375_2.3125_0.76562_3.4688_0.98438-1.1562_0.22656_2.3125_0.34375_3.4688_0.34375-2.7734_0-4.9297-0.70312-6.4688-2.1094-1.5312-1.4062_2.2969_3.3789-2.2969_5.9219_0-2.5_0.75391-4.4609_2.2656-5.8906_1.50878-1.4375_3.582-2.1562_6.2188_2.1562_2.4062_0.4_3.32_0.73047_5.7812_2.1875_1.4492_1.4492_2.1719_3.3828_2.1719_5.7969z-5.0312_1.625c0-0.92578_0.27344-1.6719_0.8125-2.2344-0.54297-0.57031-1.25_0.85938-2.125_0.85938-0.94922_0-1.7188_0.26562-2.3125_0.79688s-0.96484_1.2969-1.1094_2.2969z" fill="currentColor"/>
120           </symbol>
121           <symbol id="c" overflow="visible" fill="currentColor">
122             <path d="m9.2188-6.8906c-1.0234_0_1.793_0.17188-2.3125_0.51562-0.51172_0.34375-0.76562_0.85547_0.76562_1.5312_0_0.625_0.20703_1.1172_0.625_1.4688_0.41406_0.34375_0.98828_0.51562_1.7188_0.51562_0.92578_0_1.7031_0.3281_2.3281_0.98438_0.63281_0.66406_0.95312_1.4922_0.95312-2.4844v-0.5625z-7.4688_1.8438v8.7344h-4.9219v-2.2656c-0.65625_0.92969_1.3984_1.6055-2.2188_2.0312_0.82422_0.41406_1.8242_0.625_3.0_2.658_1.5859_0.2_8711_0.45703_3.8594_1.375_0.99219_0.92578_1.4844_2.1289_1.4844_3.6094_0_1.7891_0.61328_3.1016_1.8438_3.9375_1.2383_0.84375_3.1797_1.2656_5.8281_1.2656h2.8906v-0.3966c-0.76953_0.30859_1.332_0.92188_1.6875_0.61719_0.36328_1.5703_0.54688_2.8594_0.54688_1.0547_0_2.0312_0.10547_2.9375_0.3125_0.89844_0.21094_1.7305_0.52344_2.5_0.9375v-3.7344c1.0391_0.25_2.0859_0.44141_3.1406_0.57812_1.0625_0.13281_2.125_0.2031_3.1875_0.20312_2.7578_0_4.75_0.54688_5.9688_1.6406_1.2266_1.0859_1.8438_2.8555_1.8438_5.3125z" fill="currentColor"/>
123           </symbol>
124           <symbol id="d" overflow="visible" fill="currentColor">
125             <path d="m7.7031-19.656v4.3438h5.0469v3.5h-5.0469v6.5c0_0.71094_0.14062_0.1875_0.42188_1.4375s0_.83594_0.375_1.6719_0.375h2.5156v3.5h-4.1875c-1.9375_0-3.3125_0.39844_4.125_1.2031_0.80469_0.8125_1.2031_2.1797_1.2031_4.1094v-6.5h-2.4219v-3.5h2.4219v-4.3438z" fill="currentColor"/>
126           </symbol>
127           <symbol id="m" overflow="visible" fill="currentColor">
128             <path d="m12.766-13.078v-8.2031h4.9219v21.281h-4.9219v-2.2188c-0.66797_0.90625_1.4062_1.5703_2.2188_1.9844s-1.7578_0.625_2.8281_0.625c-1.8867_0_3.4336_0.75_4.6406_2.25_1.2109_1.5_1.8125_3.4258_1.8125_5.7812_0_2.3633_0.60156_4.2969_1.8125_5.7969_1.207_1.5_2.7539_2.25_4.6406_2.25_1.0625_0_2.2188_0.21484_2.8125_0.64062_0.82031_0.42969_1.5664_1.0859_2.2344_1.9688z-3.2188_9.9219c1.0391_0" fill="currentColor"/>
129           </symbol>

```

```

130      _1.8359_0.37891_2.3906_1.1406_0.55078_0.76953_0.82812_1.8828_0.82812_3.3438_0_1.457_0.27734-
131      2.5664_0.82812_3.3281_0.55469_0.76953_1.3516_1.1562_2.3906_1.1562_1.043_0_1.8398_0.38672-
132      2.3906_1.1562_0.55469_0.76172_0.82812_1.8711_0.82812_3.3281_0_1.4609_0.27344_2.5742_0.82812_-
3.3438_0.55078_0.76172_1.3477_1.1406_2.3906_1.1406_ fill="currentColor"/>
133  </symbol>
134  <symbol id="l" overflow="visible" fill="currentColor">
135    <path d="m10_5-3.1562c1.0508_0_1.8516_0.37891_2.4062_1.1406_0.55078_0.76953_0.82812_1.8828_0.82812_-
3.3438_0_1.457_0.27734_2.5664_0.82812_3.3281_0.55469_0.76953_1.3555_1.1562_2.4062_1.1562_-
1.0547_0_1.8594_0.38672_2.4219_1.1562_0.55469_0.77344_0.82812_1.8828_0.82812_3.3281_0_1.4492_-
0.27344_2.5586_0.82812_3.3281_0.5625_0.77344_1.3672_1.1562_2.4219_1.1562zmm-3.25_9.9219c_67578_-
0.88281_1.4219_1.5391_2.2344_1.9688_0.82031_0.42578_1.7656_0.64062_2.8281_0.64062_1.8945_0_-
3.4453_0.75_4.6562_2.25_1.207_1.5_1.8125_3.4336_1.8125_5.7969_0_2.3555_0.60547_4.2812_1.8125_-
5.7812_1.2109_1.5_2.7617_2.25_4.6562_2.25_1.0625_0_2.0078_0.21094_2.8281_0.625_0.8125_0.42578_-
1.5586_1.0859_2.2344_1.9844v2.2188h4.8906v21.281h4.8906z" fill="currentColor"/>
136  </symbol>
137  <symbol id="k" overflow="visible" fill="currentColor">
138    <path d="m0.34375_15.312h4.890614.125_10.391_3.5_10.391h4.89061-6.4375_16.766c_0.64844_1.6953_1.4023_-
2.8828_2.2656_3.5625_0.86719_0.6875_2_1.0312_3.4062_1.0312h-2.8438v-3.2188h1.5312c0.83203_0_-
1.4375_0.13672_1.8125_0.40625_0.38281_0.26172_0.67969_0.73047_0.89062_1.406210.14062_0.42188z_-
fill="currentColor"/>
139  </symbol>
140  <symbol id="j" overflow="visible" fill="currentColor">
141    <path d="m14.953_3.7188h-8.2188l-1.3125_3.7188h-5.2812l7.5625_20.406h6.2656l7.5625_20.406h-5.2812zm_-
6.9062_3.7812h5.5781l-2.7812-8.125z" fill="currentColor"/>
142  </symbol>
143  <symbol id="i" overflow="visible" fill="currentColor">
144    <path d="m17.75_9.3281v9.3281h-4.9219v-7.1406c0_1.3203_0.03125_2.2344_0.09375_2.7344s_0.16797_-
0.86719_0.3125_1.1094c_0.1875_0.3125_0.4492_0.55469_0.78125_0.73438_0.32422_0.17578_0.69531_-
0.26562_1.1094_0.26562_1.0234_0_1.8242_0.39844_2.4062_1.1875_0.58594_0.78125_0.875_1.8711_-
0.875_3.2656v7.5312h-4.8906v7.5312h4.8906v2.2344c0.73828_0.88281_1.5195_1.5391_2.3438_1.9688_-
0.83203_0.42578_1.75_0.64062_2.75_0.64062_1.7695_0_3.1133_0.54688_4.0312_1.6406_0.91406_1.0859_-
1.375_2.6562_1.375_4.7188z" fill="currentColor"/>
145  </symbol>
146  <symbol id="h" overflow="visible" fill="currentColor">
147    <path d="m2.1875_5.9688v-9.3438h4.9219v1.5312c0_0.83594_0.007813_1.875_0.015625_3.125_0.011719_1.25_-
0.015625_2.0859_0.015625_2.5_0_1.2422_0.03125_2.1328_0.09375_2.6719_0.070313_0.54297_0.17969_-
0.93359_0.32812_1.1719_0.20783_0.32422_0.47266_0.57422_0.79688_0.75_0.32031_0.16797_0.69141_-
0.25_1.1094_0.25_1.0195_0_1.8203_0.39062_2.4062_1.1719_0.58203_0.78125_0.875_1.8672_0.875_-
3.2656v7.5625h4.8906v15.312h-4.8906v-2.2188c-0.74219_0.89844_1.5234_1.5586_2.3438_1.9844_-
0.82422_0.41406_1.7344_0.625_2.7344_0.625_1.7617_0_3.1055_0.53906_4.0312_1.625_0.92969_1.082_-
1.3906_2.6602_1.3906_4.7344z" fill="currentColor"/>
148  </symbol>
149  <symbol id="u" overflow="visible" fill="currentColor">
150    <path d="m14.312_14.820v3.7188c_1.043_0.4375_2.0547_0.76562_3.0312_0.98438_0.98047_0.21875_1.9023_-
0.32812_2.7656_0.32812_0.92969_0.6211_0.11719_2.0781_0.34375_0.4492_0.23047_0.67188_0.58984_-
0.67188_1.0781_0_0.38672_0.17188_0.68359_0.51562_0.89062_0.34375_0.21094_0.95703_0.36719_-
1.8438_0.4687510.85938_0.125c2.5078_0.32422_4.1953_0.85156_5.0625_1.5781_0.86328_0.73047_-
1.2969_1.8711_1.2969_3.4219_0_1.6367_0.60547_2.8672_1.8125_3.6875_1.1992_0.8125_2.9922_1.2188_-
5.375_1.2188_1.0234_0_2.0742_0.078125_3.1562_0.23438_1.0742_0.15625_2.1797_0.39453_3.3125_-
0.71875v-3.7188c0.96875_0.48047_1.9609_0.83984_2.9844_1.0781_1.0312_0.23047_2.0781_0.34375_-
3.1406_0.34375_0.95703_0_1.6758_0.12891_2.1562_0.39062_0.47656_0.26953_0.71875_0.66406_0.71875_-
1.1875_0_0.4375_0.16979_0.75781_0.5_0.96875_0.33594_0.21875_0.99609_0.38281_1.9844_0.51_-
0.85938_0.10938c-2.1797_0.26953_3.7031_0.77344_4.5781_1.5156_0.875_0.73828_1.3125_1.8594_-
1.3125_3.3594_0_1.625_0.55078_2.8281_1.6562_3.6094_1.1133_0.78966_2.8203_1.1875_5.125_1.1875_-
0.89453_0.1_8359_0.074219_2.8281_0.21875_1_0.13672_2.082_0.35156_3.25_0.64062z" fill="-
currentColor"/>
151  </symbol>
152  <symbol id="g" overflow="visible" fill="currentColor">
153    <path d="m17.75_9.3281v9.3281h-4.9219v-7.1094c0_1.3438_0.03125_2.2656_0.09375_2.7656s_0.16797_-
0.86719_0.3125_1.1094c_0.1875_0.3125_0.4492_0.55469_0.78125_0.73438_0.32422_0.17578_0.69531_-
0.26562_1.1094_0.26562_1.0234_0_1.8242_0.39844_2.4062_1.1875_0.58594_0.78125_0.875_1.8711_-
0.875_3.2656v7.5312h-4.8906v-21.281h4.8906v8.2031c0.73828_0.88281_1.5195_1.5391_2.3438_1.9688_-
0.83203_0.42578_1.75_0.64062_2.75_0.64062_1.7695_0_3.1133_0.54688_4.0312_1.6406_0.91406_1.0859_-
1.375_2.6562_1.375_4.7188z" fill="currentColor"/>
154  </symbol>
155  <symbol id="f" overflow="visible" fill="currentColor">
156    <path d="m2.5781_20.406h5.87517.4219_14v-14h4.9844v20.406h-5.87517.4219_14v14h-4.9844z" fill="-
currentColor"/>
157  </symbol>
158  <symbol id="t" overflow="visible" fill="currentColor">
159    <path d="m0.42188_15.312h4.890613.8281_10.578_3.7969_10.578h4.9062l-6.0312_15.312h-5.375z" fill="-
currentColor"/>
160  </symbol>
161  <symbol id="s" overflow="visible" fill="currentColor">
162    <path d="m2.3594_21.281h4.8906v11.59415.625_5.625h5.625h5.625_7.4688_7.0312_8.0625_8.2812h-5.9375l_-
5.9688_6.3906v6.3906h-4.8906z" fill="currentColor"/>
163  </symbol>
164  <symbol id="r" overflow="visible" fill="currentColor">
165    <path d="m12.422_21.281v3.2188h-2.7031c_0.6875_0_1.1719_0.125_1.4531_0.375_0.27344_0.25_0.40625_-
0.6875_0_4.0625_1.3125v1.0625h4.1875v3.5h-4.1875v11.812h-4.8906v-11.812h-2.4375v-3.5h2.4375v-_
1.0625c_0.1644_0.46094_2.8984_1.3906_3.7031_0.92578_0.80078_2.3672_1.2031_4.3281_1.2031z" fill="-
currentColor"/>
166  </symbol>
167  <symbol id="e" overflow="visible" fill="currentColor">
168    <path d="m9.6406_12.188c_1.0859_0_1.9141_0.39062_2.4844_1.1719_0.57422_0.78125_0.85938_1.9062_-
0.85938_3.375s0.28516_2.5938_0.85938_3.375c0.57031_0.77344_1.3984_1.1562_2.4844_1.1562_1.0625_-
0_1.875_0.38281_2.4375_1.1562_0.57031_0.78125_0.85938_1.9062_0.85938_3.375s0.28906_2.5938_-
0.85938_3.375c_0.5625_0.78125_1.375_1.1719_2.4375_1.1719zmm-3.5c2.6328_0_4.6914_0.71484_6.1719_-
2.1406_1.4766_1.418_2.2188_3.3867_2.2188_5.9062_0_2.5117_0.74219_4.4805_2.2188_5.9062_1.4805_-
1.418_3.5391_2.125_6.1719_2.125_6.1719_2.4375_1.4922_1.4258_2.2344_3.3945_2.2344_5.9062_0_2.5195_0.74219_4.4883_2.2344_5.9062_1.4883_1.4258_3.5547_2.1406_6.2031_2.1406_ fill="currentColor"/>
169  </symbol>
170  <symbol id="q" overflow="visible" fill="currentColor">
171    <path d="m16.547_12.766c0.61328_0.94531_1.3477_1.6719_2.2031_2.1719_0.85156_0.5_1.7891_0.75_2.8125_-
0.75_1.7578_0_3.0977_0.54688_4.0156_1.6406_0.92578_1.0859_1.3906_2.6562_1.3906_4.7188v9.3281h-_
4.9219v-7.9844_0.35938_0.007813_0.13281_0.015625_0.32031_0.015625_0.5625_0_1.082_0.16406_-
1.8633_0.48438_2.3438_0.3125_0.48828_0.82422_0.73438_1.5312_0.73438_0.92969_0_1.6484_0.38672_-
2.1562_1.1562_0.51172_0.76172_0.77344_1.8672_0.78125_3.3125v7.5156h-4.9219v-7.9844c0_1.6953_-

```

```

0.14844-2.7852-0.4375-3.2656-0.29297-0.48828-0.8125-0.73438-1.5625-0.73438-0.9375_0-1.6641_
0.38672-2.1719_1.1562-0.51172_0.76172-0.76562_1.8594-0.76562_3.2969v7.5312h-4.9219v-15.312h4
.9219v2.2344c0_60156-0.86328_1.2891-1.5156_2.0625-1.9531_0.78125-0.4375_1.6406-0.65625_2.5781-
0.65625_1.0625_0_2_0.25781_2.8125_0.76562_0.8125_0.51172_1.4258_1.2305_1.8438_2.1562" fill="
currentColor"/>
169 </symbol>
170 <symbol id="p" overflow="visible" fill="currentColor">
171   <path d="m2.5781-20.406h8.7344c2.5938_0.4.582_0.57812_5.9688_1.7344_1.3945_1.1484_2.0938_2.7891_
2.0938_4.9219_0_2.1367-0.69922_3.7812-2.0938_4.9375-1.3867_1.1562-3.375_1.7344-5.9688_1.7344h-
3.4844v7.0781h-5.25zm5.25_3.8125v5.7031h2.9219c1.0195_0_1.8047-0.25_2.3594-0.75_0.5625-0.5_-
0.84375_1.2031_0.84375_2.1094_0_0.91406_0.28125-1.6172-0.84375_2.1094-0.55469-0.48828-1.3398-
0.73438-2.3594-0.73438z" fill="currentColor"/>
172 </symbol>
173 <symbol id="o" overflow="visible" fill="currentColor">
174   <path d="m2.3594-15.312h4.8906v15.031c0_2.0508-0.49609_3.6172-1.4844_4.7031-0.98047_1.082-2.4062-
1.625-4.2812_1.625h-2.4219v-3.8125v5.89593c0.92578_0_1.5625-0.21094_1.9062-0.625_0.35156-
0.41797_0.53125_1.2461_0.53125-2.4844zm0-5.9688h4.8906v4h-4.8906z" fill="currentColor"/>
175 </symbol>
176 <symbol id="n" overflow="visible" fill="currentColor">
177   <path d="m14.719-14.828v3.9844c-0.65625-0.45703-1.3242-0.79688-2-1.0156-0.66797-0.21875-1.3594-
0.32812-2.0781-0.32812-1.3672-0.2-4.336_-0.40234-3.2031_1.2031-0.76172_0.79297-1.1406_1.9062-
1.1406_3.3438_0_1.4297_0.37891_2.543_1.1406_3.3438_0.76953_1.8359_1.1875_3.2031_1.1875
_0.75781_0_1.4844-0.10938_2.1719_0.32812_0.6875-0.22656_1.3203_0.56641_1.9062-1.0156v4c-
0.76172_0.28125-1.5391_0.48828-2.3281_0.625-0.78125_0.14453-1.5742_0.21875-2.375_0.21875-
2.7617_0_4.9219_0.70703_6.4844-2.125_1.5547_1.4141_2.3281-3.3828_2.3281_5.9062_0_2.5312_-
0.77344-4.5639_2.3281-5.9219_1.5625-1.4141_3.7227-2.125_6.4844-2.125_0.80078_0_1.5938_0.074219
_2.375_0.21875_0.78125_0.13672_1.5547_0.35156_2.3281_0.64062z" fill="currentColor"/>
178 </symbol>
179 </defs>
180 <g>
181   <path d="m613.26_296.75c-9.2773_0-16.801-7.5273-16.801-16.801s7.5195-16.801_16.801-16.801c9.2773_0_-
16.801_7.5195_16.801_16.801-0.003906_0.2812-7.5234_16.801-16.801zm-28.059-5.7266h-11.199
c-3.0898_0-5.6016-2.5078-5.6016-5.6016v-11.199c0_3.0977_2.5078_5.6016-5.6016-5.6016h11.199c3
.0977_0_5.6016_2.5039_5.6016_5.6016v11.199c0_3.0938_2.5039_5.6016-5.6016_5.6016zm-238.04_196.17
c-109.73_0-204.36-95.949-204.36-207.2594.629-207.2c204.36-207.2c109.73_0_210.04_95.945_210.04_
207.25-100.3_22-210.04_207.2z0.085938-392c99.02_0-182.05_-0.84.281-182.05_184.8s83.027_184.8_
182.04_184.8c99.023_0.187.56-84.28c187.56-184.8_0-100.52-88.543-184.8-187.55-184.8zm-14.047_-
229.26v-16.789c6.3867_0.20312_12.852_0.33203_19.457_0.33203_84.883_0_148.54-16.801_148.54-28-
63.66-28.148.55-28c-6.6016_-0_13.07_0.12891-19.453_0.32422v-16.789c6.418-0.20703_12.914-0.33594_
25.531-95.199h16.734c6.3984_70.598-24.297_117.6-42.266_117.6-22.398_0-44.801-72.801-44.801-
173.6522.398-173.6_44.801-173.6c17.98_0_35.934_46.961_42.332_117.6h-16.781c-5.0703-56.891-
16.949-95.199-25.551-95.199zm-173.6_151.2c0-17.711_48.352-35.379_117.6-42.016v16.715c-55.832_
5.1953-95.199_16.84-95.199_25.301s39.367_20.105.95_199_25.301v16.715c-69.25-6.6367-117.6-24.305
-117.6-42.016zm100.8_0c0-4.082_6.3164-7.8164_16.801-10.734v21.465c-10.484-2.9141-16.801-6.6484-
16.801-10.73z#78.109-240.8c3.0781_0_5.5703_2.5078_5.5703_5.6016v11.199c0_3.0898-2.4922_5.6016-
5.5703_5.6016h-11.137c-3.0781_0-5.5703_2.5078_5.5703_5.6016l-0.03906-11.199c0-3.0898_2.4922-
5.6016_5.5703-5.6016zm-5.4375-5.6562c-9.2305_0_16.711-7.5195-16.711-16.801_0_9.2773_7.4805-
16.801_16.711-16.801_9.2305_0_16.711_7.5195_16.711_16.801s-7.4805_16.801_16.711_16.801zm-223.87
_257.66h-11.199c-3.0898_0-5.6016-2.5078_5.6016-5.6016v-11.199c0_3.0898_2.5078-5.6016_5.6016-
5.6016h11.199c3.0977_0_5.6016_2.5078_5.6016_5.6016v11.199c0_3.0938-2.5039_5.6016-5.6016_5.6016
zm-39.258_5.7305c-9.2734_0_16.801-7.5273-16.801_0_9.2773_7.5273-16.801_16.801_0_9.2773_7.5273-
16.801_16.801_0_9.2773_7.5195_16.801_16.801_16.801zm257.6_223.87c-3.0781_0_5.5703_2.5078_5.5703_5.6016v-11.199c0_3.0977_2.4922-5.6016_5.5703_5.6016h11.137c3.0781_0_
5.5703_2.5039_5.5703_5.6016v11.199c0_3.0898-2.4922_5.6016-5.5703_5.6016zm5.6992_5.6562c9.2305_0
_16.711_7.5195_16.711_16.801_0_9.2734-7.4805_16.801_16.711_16.801s-7.4805_16.801_16.711_16.801c-
0.003906-9.2812_7.4766-16.801_16.711-16.801" fill-rule="evenodd" fill="currentColor"/>
182   </g>
183 </svg>
184 </span>
185   Akeselerasi
186 </a>
187 </li>
188
189
190
191
192
193   </ul>
194 </li>
195
196 <li class="nav-item">
197   <a class="nav-link_active" aria-current="page" href="table">
198     <span>
199       <svg xmlns="http://www.w3.org/2000/svg" version="1.1" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:svgjs="
http://svgjs.com/svgjs" width="512" height="512" x="0" y="0" viewBox="0_0_512_512" style="enable-
background:new_0_0_512_512" xml:space="preserve" class=""><g><path d="M488.468_79.645H23.532c-8.854_0-
16.032_7.178-16.032_16.032v320.645c0_8.854_7.178_16.032_16.032v246.493c8.854_0_16.032-7.178_16.032-
16.032V95.677c0.001-8.854-7.177-16.032-16.031c-16.032zM127.742_111.71h256.516M440.371_143.774H7.5M504.5-
143.774z-32.065" style="stroke-width:15;stroke-linecap:round;stroke-linejoin:round;stroke-miterlimit:10;" fill="none" stroke="currentColor" stroke-width="15" stroke-linecap="round" stroke-linejoin="round" stroke-miterlimit="10" data-original="#000000" class=""></path><path d="M231.952_239.968h-64.129a8.016_8.016_0_0_1.8.016_8.016v32.065a8.016_0_0_1.8.016-
8.016_0_0_1.8.016_8.016M162.823_288.065h64.129M167.823_336.161h64.129M167.823_384.258h64.129M344.177-
239.968h-64.129a8.016_8.016_0_0_1.8.016-8.016v-32.065a8.016_8.016_0_0_1.8.016-8.016h64.129a8.016_8.016_0_0_1.8.016-
8.016h64.129a8.016_8.016_0_0_1.8.016-8.016v32.065a8.016_8.016_0_0_1.8.016-8.016M392.274_336.161h64.129M392.274_384.258h64.129M55.597_239.968a8.016_8.016_0_0_1.8.016-8.016v-32.065a8.016_8.016_0_0_1.8.016-8.016h64.129
M55.597_288.065h64.129M55.597_336.161h64.129M55.597_384.258h64.129" style="stroke-width:15;stroke-linecap:round;stroke-linejoin:round;stroke-miterlimit:10;" fill="none" stroke="currentColor" stroke-width="15" stroke-linecap="round" stroke-linejoin="round" stroke-miterlimit="10" data-original="#000000" class=""></path></g></svg>
```

```

200
201       
```

```

        none" stroke="currentColor" stroke-width="40" stroke-linecap="round" stroke-linejoin="round" stroke-miterlimit="10" stroke-dasharray="none;stroke-opacity:1" data-original="#000000" class="">></path><path d="M0_0h-271" style="stroke-width:40;stroke-linecap:round;stroke-linejoin:round;stroke-miterlimit:10;stroke-dasharray:none;stroke-opacity:1" transform="translate(487_255)" fill="none" stroke="currentColor" stroke-width="40" stroke-linecap="round" stroke-linejoin="round" stroke-miterlimit="10" stroke-dasharray="none" stroke-opacity="" data-original="#000000" class="">></path><path d="m0_0_44.786_44.787c11.716_11.716_30.71_0_42.426l0_132" style="stroke-width:40;stroke-linecap:round;stroke-linejoin:round;stroke-miterlimit:10;stroke-dasharray:none;stroke-opacity:1" transform="translate(438_427_189)" fill="none" stroke="currentColor" stroke-width="40" stroke-linecap="round" stroke-linejoin="round" stroke-miterlimit="10" stroke-dasharray="none" stroke-opacity="" data-original="#000000" class="">></path></g></g></g></svg>
255      </span>
256      Log out
257    </a>
258
259    </form>
260  </li>
261 </ul>
262 </div>
263 </div>
```

Kode F.12: Website - signUP.blade.php

```

1 @include('header', ['menu'=> false ])
2 <link rel="stylesheet" href="{{'css/login.css'}}">
3
4
5 <div class="d-flex_justify-content-end" style="margin:_30px_50px_0px_0px">
6   
7   <div class="card_login" style="width:_18rem;">
8     <div class="card-body">
9       <div class="text-center">
10         
11
12       <h3>Sign UP</h3>
13       <form class="form-signin" action="/daftar" method="post">
14         @csrf
15         {{-- frame work laravel yang membantu mengenerate token --}}
16         <div class="form-group">
17           <label>Username</label>
18           <input type="text" class="form-control" id="username" aria-describedby="emailHelp" placeholder="Enter_username" name="username">
19         </div>
20         <div class="form-group">
21           <label>Email Address</label>
22           <input type="email" class="form-control" id="email" aria-describedby="emailHelp" placeholder="Enter_email" name="email">
23         </div>
24         <div class="form-group_mt-2">
25           <label>Password</label>
26           <input type="password" class="form-control" id="password" placeholder="Password" name="password">
27         </div>
28         <div>
29           Already have account? <a href="/">Login</a>
30         </div>
31         <button type="submit" class="btn btn-primary_mt-3">Sign Up </button>
32       </form>
33
34     </div>
35   </div>
36
37
38 </div>
39
40
41
42
43
44
45 @include("footer")
```

Kode F.13: Website - statistik.blade.php

```

1 <script src="https://cdn.plot.ly/plotly-2.23.2.min.js" charset="utf-8"></script>
2
3 <script>
4   const urlAPI='{{$api}}'
5 </script>
6 <link rel="stylesheet" href="{{'css/mainUser.css'}}">
7
8 <script src="{{asset('js/lokasiFilter.js')}}" defer></script>
9 <script src="{{asset('js/statistic.js')}}" defer></script>
10
11
12 <div class="container" id="workSheet">
13   <div class="row">
14     <br>
15     <div class="col-6">
16       <Select class="form-select" id="kotaSelection">
17         <option value="-1">Pilih Kota</option>
18         @foreach ($semuaKota as $kota)
19           <option value="{{$kota['id']}}">{{$kota['nama']}}</option>
20         @endforeach
21       </Select>
22     </div>
```

```

23  <div class="col-6">
24    <Select class="form-select" id="lokasiSelection" >
25      <option value="-1">Pilih Lokasi</option>
26      @foreach ($semualokasi as $lokasi)
27        <option hidden class="optLokasi" value="{{$lokasi['identifier']}}" city="{{$lokasi['idKota']}}>{{$lokasi['namaLokasi']}}</option>
28      {{($lokasi['indoors'])==1?"_Indoor":""}} {{("lat:".$lokasi['latitude'].",long:".$lokasi['longitude'])."}}</option>
29    @endforeach
30  </Select>
31  </div>
32  <div class="col-1">
33    <div class="pt-2" >Start Time </div>
34    <div class="pt-3">Interval </div>
35  </div>
36  <div class="col-3">
37    <input type="date" id="startDate" class="mt-2" value="2023-06-06">
38    <input type="time" id="startTime" class="mt-2" value="23:00">
39    <input type="number" placeholder="second" class="mt-2" id="intervalInput" value=60>
40  </div>
41  <div class="col-2">
42    <div class="pt-2" >End Time </div>
43    <div class="pt-3">Diagram Type</div>
44  </div>
45  <div class="col-3">
46    <div>
47      <input type="date" id="endDate" class="mt-2" value="2023-06-07">
48      <input type="time" id="endTime" class="mt-2" value="00:00">
49    </div>
50  </div>
51  <div>
52    <select class="mt-2" name="b" id="typeSelection">
53      <option value="1" for="suhu_kelembapan">Pilih Diagram</option>
54      <option value="scattered">Scatter plot</option>
55      <option value="boxFloating">Min-Max</option>
56      <option value="box">Box plot</option>
57      <option value="summary">Summary</option>
58    </select>
59  </div>
60 </div>
61 <br>
62
63 <br>
64
65 <div class="alert alert-secondary" role="alert" style="display:none">
66 Lokasi dipilih:
67 @foreach ($semualokasi as $lokasi)
68  {{-- <option hidden class="optLokasi" value="{{$lokasi['identifier']}}" city="{{$lokasi['idKota']}}>{{$lokasi['nama']}}</option>
69  {{-- >--}}}
70 <button type="button" hidden class="btn btn-secondary selectedLokasi" onclick="removeSelected(this)" value="{{$lokasi['identifier']}}>{{$lokasi['namaLokasi']}}</button>
71 @endforeach
72
73 </div>
74
75 <div class="alert alert-light" role="alert" id="scatter" hidden>
76   <h4>Konfigurasi Scatter</h4>
77   Pilih data untuk X dan Y
78
79   <select class="form-select" aria-label="Default_select_example" id="scatterX">
80     <option selected>Axis X</option>
81     <option value="suhu">Suhu</option>
82     <option value="kelembapan">Kelembapan</option>
83     <option value="Tekanan">Tekanan</option>
84   </select>
85
86   <br>
87
88   <select class="form-select" aria-label="Default_select_example" id="scatterY">
89     <option selected>Axis Y</option>
90     <option value="suhu">Suhu</option>
91     <option value="kelembapan">Kelembapan</option>
92     <option value="Tekanan">Tekanan</option>
93   </select>
94
95 </div>
96
97 <div class="alert alert-light" role="alert" id="summary" hidden>
98   <h4>Konfigurasi Summary</h4>
99
100  <select class="form-select" aria-label="Default_select_example" id="summaryType">
101    <option selected>Pilih Parameter</option>
102    <option value="in0">Kuartile</option>
103    <option value="split-1">Jumlah per 1</option>
104    <option value="split-2">Jumlah per 2</option>
105    <option value="split-5">Jumlah per 5</option>
106    <option value="split-10">Jumlah per 10</option>
107    <option value="split-20">Jumlah per 20</option>
108  </select>
109
110  <br>
111
112  <select class="form-select" aria-label="Default_select_example" id="summaryGraph">
113    <option selected>Pilih Tipe Visualisasi</option>
114    <option value="histogram">Histogram (Rekomendasi untuk Satu lokasi atau Jumlah per kecil)</option>
115    <option value="box">Diagram Batang (Rekomendasi untuk lebih dari Satu lokasi)</option>
116    <option value="pie">Diagram Lingkaran</option>
117  </select>

```

```

118 </div>
119
120
121 <button onclick="show()" class="btn btn-primary mt-2">Tampilkan Visualisasi</button>
122 {{-- <div id="myPlot" style="width:100%;_height:_30px;"></div> --}}
123 <div id="myPlotsuhu" class="chart" style="width:100%;_height:_30px;_display:none"></div>
124 <div id="myPlottekanan" class="chart" style="width:100%;_height:_30px;_margin-top:750px;_display:none"></div>
125
126
127 <div id="myPlotkelembapan" class="chart" style="width:100%;_height:_30px;_margin-top:750px;_display:none"></div>
128
129
130
131
132 <div id="myPlotakselerasi" class="chart" style="width:100%;_height:_30px;_margin-top:750px"></div> --}
133
134
135 {{--
136   <div id="myPlotakselerasi" class="chart" style="width:100%;_height:_30px;_margin-top:750px"></div> --}}
137
138
139   @include("accelerationInfo")
140 </div>

```

Kode F.14: Website - tabel.blade.php

```

1 <script>
2   const urlAPI='{{$api}}'
3
4   </script>
5 <script src="{{asset('js/lokasiFilter.js')}}" defer></script>
6 <script src="{{asset('js/table.js')}}" defer></script>
7
8 <div class="container" id="workSheet">
9   <br>
10  <div class="row">
11
12    <div class="col-6">
13      <Select class="form-select" id="kotaSelection">
14        <option value="-1">Pilih Kota</option>
15        @foreach ($semuaKota as $kota)
16          <option value="{{$kota['id']}}">{{$kota['nama']}}</option>
17        @endforeach
18      </Select>
19    </div>
20    <div class="col-6">
21      <Select class="form-select" id="lokasiSelection" >
22        <option value="-1">Pilih Lokasi</option>
23        @foreach ($semuaLokasi as $lokasi)
24          <option hidden class="optLokasi" value="{{$lokasi['identifier']}}" city="{{$lokasi['idKota']}}">{{$lokasi['namaLokasi']}}
25        @endforeach
26      </Select>
27    </div>
28    <div class="col-1">
29      <div class="pt-2" >Start Time </div>
30      <div class="pt-3">Interval </div>
31    </div>
32    <div class="col-3">
33      <input type="date" id="startDate" class="mt-2">
34      <input type="time" id="startTime" class="mt-2">
35      <input type="number" placeholder="Second" class="mt-2" id="intervalInput">
36    </div>
37
38    <div class="col-2">
39      <div class="pt-2" >End Time </div>
40    </div>
41
42    <div class="col-3">
43      <div>
44        <input type="date" id="endDate" class="mt-2">
45        <input type="time" id="endTime" class="mt-2">
46      </div>
47    </div>
48  </div>
49  <div class="row">
50    <div class="col-4" >
51      <button onclick="show()" class="btn btn-primary mt-2">Tampilkan Tabel</button>
52    </div>
53  </div>
54
55
56 <div class="row" style="background-color:white;_padding:50px_20px;_display:none">
57   <table id="tabellump" class="table_table-striped">
58     <thead>
59       <tr>
60         <th scope="col">Tanggal</th>
61         <th scope="col">Suhu</th>
62         <th scope="col">Kelembapan</th>
63         <th scope="col">Tekanan</th>
64         <th scope="col">Akselerasi (X,Y,Z)*</th>
65       </tr>
66     </thead>
67
68
69
70   </table>
71

```

```

72     </div>
73
74     @include("accelerationInfo")
75

```

Kode F.15: Website - live.js

```

1
2
3 const listLokasi=document.getElementsByClassName("optLokasi");
4 const chartKelembapan=document.getElementById("kelembapanSelection")
5 const chartSuhu=document.getElementById("suhuSelection")
6 const chartTekanan=document.getElementById("tekananSelection")
7 const chartAkselerasi=document.getElementById("akselerasiSelection")
8
9
10 const layoutLive = { width: 300, height: 150, margin: { t: 40, l: 30, b: 40, r:30 }};
11
12
13 class Live{
14     constructor(api,identifier){
15         this.url=api+identifier;
16         this.idBS=identifier;
17         this.time=null;
18
19         this.kelembapan=document.getElementById("kelembapanText"+identifier)
20         this.suhu=document.getElementById("suhuText"+identifier)
21         this.tekanan=document.getElementById("tekananText"+identifier)
22         this.akselerasi=document.getElementById("akselerasiText"+identifier)
23         this.timeDoc=document.getElementById("timeText"+identifier)
24
25
26         this.suhuFirstValue=true
27         this.kelembapanFirstValue=true
28         this.tekananFirstValue=true
29         this.akselerasiFirstValue=true
30         this.suhuType='text'
31         this.kelembapanType='text'
32         this.tekananType='text'
33         this.akselerasiType='text'
34
35     }
36
37     visibilityText(elem,visibility,plotName){
38         // try {
39             if(visibility){
40                 (elem.parentElement.parentElement).setAttribute("style", "display:inline-block!important");
41                 Plotly.purge(plotName)
42             }else{
43                 (elem.parentElement.parentElement).setAttribute("style", "display:none!important");
44             }
45         }
46     }
47
48     showTemperature(value){
49         let plotName="myPlotsuhu"+this.idBS //diconcat dengan id bs karena myplotsuhu banyak, bisa untuk banyak bs
50
51         switch(chartSuhu.value) {
52             case "lines":
53
54                 if(this.suhuType!='lines'){
55                     this.suhuFirstValue=true;// agar dia reset kalau yg berubah dari scatter
56                     this.suhuType='lines'
57                 }
58
59
60                 this.showLines(plotName,this.suhuFirstValue,value,'Suhu',"lines")
61                 this.visibilityText(this.suhu,false,plotName)
62                 this.suhuFirstValue=false;
63                 break;
64             case "simpleText":
65                 this.suhuType=undefined
66                 this.visibilityText(this.suhu,true,plotName)
67                 this.showSimpleText(this.suhu,value)
68                 this.suhuFirstValue=true;
69                 break;
70             case "scatter":
71
72                 if(this.suhuType!='scatter'){
73                     this.suhuFirstValue=true;// agar dia reset kalau yg berubah dari lines
74                     this.suhuType='scatter'
75                 }
76
77                 this.showLines(plotName,this.suhuFirstValue,value,'Suhu',"area")
78                 this.visibilityText(this.suhu,false,plotName)
79                 this.suhuFirstValue=false;
80                 break;
81             // code block
82         }
83     }
84
85     showPressure(value){
86         let plotName="myPlottekanan"+this.idBS //diconcat dengan id bs karena myplottekanan banyak, bisa untuk banyak bs
87
88         switch(chartTekanan.value) {
89             case "lines":
90
91

```

```

92
93     if(this.tekananType!='lines'){
94         this.tekananFirstValue=true;// agar dia reset kalau yg berubah dari scatter
95         this.tekananType='lines'
96     }
97
98     this.showLines(plotName,this.tekananFirstValue,value,'Tekanan',"lines")
99     this.visibilityText(this.tekanan,false,plotName)
100    this.tekananFirstValue=false;
101
102    break;
103  case "simpleText":
104    this.tekananType=undefined
105    this.visibilityText(this.tekanan,true,plotName)
106    this.showSimpleText(this.tekanan,value)
107    this.tekananFirstValue=true;
108    break;
109  case "scatter":
110
111    if(this.tekananType!='scatter'){
112        this.tekananFirstValue=true;// agar dia reset kalau yg berubah dari lines
113        this.tekananType='scatter'
114    }
115
116    this.showLines(plotName,this.tekananFirstValue,value,'Tekanan',"area")
117    this.visibilityText(this.tekanan,false,plotName)
118    this.tekananFirstValue=false;
119    break;
120
121    // code block
122  }
123
124}
125
126 showHumidity(value){
127
128   let plotName="myPlotkelembapan"+this.idBS //diconcat dengan id bs karena myplotsuhu banyak, bisa untuk banyak bs
129
130   switch(chartKelembapan.value) {
131     case "lines":
132
133       if(this.kelembapanType!='lines'){
134           this.kelembapanFirstValue=true;// agar dia reset kalau yg berubah dari scatter
135           this.kelembapanType='lines'
136       }
137       value=value==-1?0:value
138
139       this.showLines(plotName,this.kelembapanFirstValue,value,'Kelembapan',"lines")
140       this.visibilityText(this.kelembapan,false,plotName)
141       this.kelembapanFirstValue=false;
142       break;
143
144     case "simpleText":
145       value=value==-1?"Offline":value
146       this.kelembapanType=undefined
147       this.visibilityText(this.kelembapan,true,plotName)
148       this.showSimpleText(this.kelembapan,value)
149       this.kelembapanFirstValue=true;
150       break;
151
152     case "scatter":
153
154       if(this.kelembapanType!='scatter'){
155           this.kelembapanFirstValue=true;// agar dia reset kalau yg berubah dari lines
156           this.kelembapanType='scatter'
157       }
158
159       value=value==-1?0:value
160       this.showLines(plotName,this.kelembapanFirstValue,value,'Kelembapan',"area")
161       this.visibilityText(this.kelembapan,false,plotName)
162       this.kelembapanFirstValue=false;
163       break;
164
165     case "gauge":
166       if(this.kelembapanType!='gauge'){
167           this.kelembapanFirstValue=true;// agar dia reset kalau yg berubah dari lines
168           this.kelembapanType="gauge"
169       }
170
171       value=value==-1?0:value
172       this.showGauge(value)
173       this.visibilityText(this.kelembapan,false,plotName)
174       this.kelembapanFirstValue=false;
175       break;
176
177     case "bar":
178
179       value=value==-1?0:value
180       if(this.kelembapanType!='bar'){
181           this.kelembapanFirstValue=true;// agar dia reset kalau yg berubah dari lines
182           this.kelembapanType='bar'
183       }
184
185   }
186
187
188 showAceleration(value){
189   let plotName="myPlotakselerasi"+this.idBS //diconcat dengan id bs karena myplotsuhu banyak, bisa untuk banyak bs
190

```

```

191 switch(chartAkselerasi.value) {
192     case "lines":
193
194         if(this.akselerasiType!="lines"){
195             this.akselerasiFirstValue=true;// agar dia reset kalau yg berubah dari scatter
196             this.akselerasiType='lines'
197         }
198
199         this.showLines3D(plotName,this.akselerasiFirstValue,value,'Akselerasi',"lines")
200         this.visibilityText(this.akselerasi,false,plotName)
201         this.akselerasiFirstValue=false;
202
203         break;
204     case "simpleText":
205         this.akselerasiType=undefined
206         this.visibilityText(this.akselerasi,true,plotName)
207         this.showSimpleText(this.akselerasi,value)
208         this.akselerasiFirstValue=true;
209         break;
210     case "scatter":
211
212         if(this.akselerasiType!="scatter"){
213             this.akselerasiFirstValue=true;// agar dia reset kalau yg berubah dari lines
214             this.akselerasiType='scatter'
215         }
216         this.showLines3D(plotName,this.akselerasiFirstValue,value,'akselerasi',"area")
217         this.visibilityText(this.akselerasi,false,plotName)
218         this.akselerasiFirstValue=false;
219         break;
220
221         // code block
222     }
223
224 }
225
226 showSimpleText(elem,value){
227
228     elem.textContent=value
229     elem.nextElementSibling.hidden=value=='Offline'?true:false
230 }
231
232 showLines3D(plotName,create,value,title,type){
233
234     if(create){
235
236         let Xtrace = {
237             x: [this.time],
238             y: [value[0]],
239             name: "X",
240             type: 'lines'
241         };
242
243         let Ytrace = {
244             x: [this.time],
245             y: [value[1]],
246             name: "Y",
247             type: 'lines'
248         };
249
250         let Ztrace = {
251             x: [this.time],
252             y: [value[2]],
253             name: "Z",
254             type: 'lines'
255         };
256
257         let data=[Xtrace, Ytrace, Ztrace]
258
259         if(type=="area"){
260             data[0].fill='tonexty'
261             data[1].fill='tonexty'
262             data[2].fill='tonexty'
263         }
264
265         console.log(data)
266
267         let newLayout={...layoutLive}; //cara copy objek, karena layoutlive merupakan layout utama, sedangkan yang ini akan
268             berubah ubah sesuai tipe data
269
270         newLayout["title"]={text:title,font:{size:20}}
271
272         Plotly.newPlot(plotName, data,newLayout);
273     }else{
274
275         let update={y:[value[0]],[value[1]],[value[2]]] , x:[this.time],[this.time],[this.time]      }
276
277
278         Plotly.extendTraces(plotName,update, [0,1,2],30)
279     }
280
281 }
282
283
284 showLines(plotName,create,value,title,type){
285     if(create){
286         let data = [
287

```

```

289         x: [this.time],
290         y: [value],
291         type: type,
292     }
293   ];
294
295   if(type=="area"){
296     data[0]["fill"]='tonexty'
297   }
298
299
300   let newLayout={...layoutLive}; //cara copy objek, karena layoutlive merupakan layout utama, sedangkan yang ini akan
301   berubah ubah sesuai tipe data
302
303   newLayout["title"]={text:title,font:{size:20}}
304
305   Plotly.newPlot(plotName, data,newLayout);
306 }else{
307
308   let update = {
309     x: [[this.time]],
310     y: [[value]]
311   }
312
313   Plotly.extendTraces(plotName,update, [0],30)
314 }
315
316
317 }
318
319
320 showGauge(value){
321   if(this.kelembapanFirstValue){
322     let datachart = [
323       {
324         domain: { x: [0, 1], y: [0, 1] },
325         value: 0.00,
326         number: { suffix: "%", valueformat: ".2f" },
327         title: { text: "Kelembapan", font: { size: 20 } },
328         type: "indicator",
329         mode: "gauge+number",
330         gauge: { axis: {
331           range: [0, 100] ,
332           tickvals: [0, 20, 40, 60, 80, 100]
333         }
334       }
335     ]
336   ];
337
338   Plotly.newPlot('myPlotkelembapan${this.idBS}', datachart, layoutLive);
339 }else{
340   Plotly.update('myPlotkelembapan${this.idBS}', {value:value}, {}, [0])
341 }
342 }
343
344
345
346 getData(){
347   fetch(this.url,{
348     method: "GET",
349     headers: {
350       Accept: 'application.json',
351       'Content-Type': 'application/json'
352     },
353   }).then(response => response.json() .then(response=>{
354
355
356   if (response['time']=='1970-01-01T00:00:00'){
357     this.timeDoc.textContent='Offline'
358   }else{
359     this.timeDoc.textContent=response['time']
360     this.time=response['time']
361   }
362
363   if(response['result']['T']==undefined){
364     this.showTemperature('Offline')
365   }else{
366     this.showTemperature(response['result']['T'])
367   }
368
369   if(response['result']['Pa']==undefined){
370     this.showPressure('Offline')
371   }else{
372     this.showPressure(response['result']['Pa'])
373   }
374
375   if(response['result']['a']==undefined){
376     this.showAceleration('Offline')
377   }else{
378     this.showAceleration(response['result']['a'])
379   }
380
381   if(response['result']['rh']==undefined){
382     this.showHumidity(-1)
383   }else{
384     this.showHumidity(response['result']['rh'])
385   }
386 }

```

```

387
388     }).catch((error) => {
389         console.log(error)
390     });
391
392
393
394
395
396
397 }
398
399
400 stop(){
401     clearInterval(this.intervalLive)
402 }
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417 }
418
419
420 function initNewLive(idBS,interval){
421
422     let live=new Live('${urlAPI}/realTime?node=',idBS);
423     setInterval(function(){live.getData()},1000);
424
425
426     console.log(interval)
427
428     // live.live(interval);
429 }
430
431
432
433 document.getElementById('kotaSelection').addEventListener("change",function(event) {
434     let selectedCity=event.target.value
435     console.log(selectedCity)
436     for (let i=0;i<listLokasi.length;i++) {
437         if(selectedCity!=-1 && listLokasi[i].getAttribute('city')!=selectedCity ) {
438             listLokasi[i].style.display="none"
439             // listLokasi[i].hidden=false
440         }else{
441             listLokasi[i].style.display="block"
442             // listLokasi[i].hidden=true
443         }
444     }
445
446
447 });

```

Kode F.16: Website - lokasiFilter.js

```

1 const startDate=document.getElementById("startDate")
2 const startTime=document.getElementById("startTime")
3 const endDate=document.getElementById("endDate")
4 const endTime=document.getElementById("endTime")
5 const intervalInput=document.getElementById("intervalInput")
6
7 const optLokasi=document.getElementById('lokasiSelection')
8 const listLokasi=document.getElementsByClassName("optLokasi");
9 const accInfo=document.getElementById("akelserasiInfo")
10
11
12
13
14
15
16 document.getElementById('kotaSelection').addEventListener("change",function(event) {
17     let selectedCity=event.target.value
18     for (let i=0;i<listLokasi.length;i++) {
19         if(listLokasi[i].getAttribute('city')!=selectedCity){
20             listLokasi[i].hidden=true
21         }else{
22             listLokasi[i].hidden=false
23         }
24     }
25     optLokasi.value=-1
26     // navSensor.hidden=true
27
28 });
29 optLokasi.addEventListener("change",function(event) {
30     // navSensor.hidden=false
31 });
32
33 function labelsType(type){
34

```

```

35  if(type=='suhu'){
36    return 'Celcius'
37  }else if(type=='kelembapan'){
38    return '%'
39  }else if(type=='tekanan'){
40    return 'kPa'
41  }else if(type=='akselerasi'){
42    return 'g'
43  }
44 }

```

Kode F.17: Website - mainAdmin.js

```

1 const ctx = document.getElementById('myChart');
2 const listLokasi=document.getElementsByClassName("optLokasi");
3 const optLokasi=document.getElementById('lokasiSelection')
4 const navSensor=document.getElementById("navSensorAvailable")
5 const optType=document.getElementById("typeSelection")
6
7
8
9
10
11
12
13
14 //LISTENER
15 document.getElementById('kotaSelection').addEventListener("change",function(event) {
16   let selectedCity=event.target.value
17   for (let i=0;i<listLokasi.length;i++) {
18     if(listLokasi[i].getAttribute('city')!=selectedCity){
19       listLokasi[i].hidden=true
20     }else{
21       listLokasi[i].hidden=false
22     }
23   }
24   optLokasi.value=-1
25
26 });
27
28 const optLokasi2=document.getElementById('lokasiSelection2')
29 const listLokasi2=document.getElementsByClassName("optLokasi2");
30
31 document.getElementById('kotaSelection2').addEventListener("change",function(event) {
32   let selectedCity=event.target.value
33   for (let i=0;i<listLokasi2.length;i++) {
34     if(listLokasi2[i].getAttribute('city')!=selectedCity){
35       listLokasi2[i].hidden=true
36     }else{
37       listLokasi2[i].hidden=false
38     }
39   }
40   optLokasi2.value=-1
41
42 });
43
44 });

```

Kode F.18: Website - mainUser.js

```

1
2
3 const ctx = document.getElementById('myChart');
4 const navSensor=document.getElementById("navSensorAvailable")
5 const optType=document.getElementById("typeSelection")
6 const chartDiv=document.getElementsByClassName("chart")
7 const chartHeight=(document.getElementById('workSheet').offsetWidth/16)*9
8
9
10 const optTypeAcceleration=document.getElementById("typeSelectionAceleration")
11 let sensorSelected=[true,false,false,false]
12 let sensorOrder=['suhu','tekanan','kelembapan','akselerasi']
13 const selectedLokasi=document.getElementsByClassName("selectedLokasi")
14 let selectedKota=[]
15 let selectedKotaName=[]
16
17 let layout = {
18   width: document.getElementById('workSheet').offsetWidth,
19   height: chartHeight,
20   margin: {
21     l: 75,
22     r: 50,
23     b: 50,
24     t: 100,
25     pad: 4
26   },
27   paper_bgcolor: '#c7c7c7',
28   plot_bgcolor: '#ffffff',
29   title : {
30     font: {
31       family: 'Courier New, monospace',
32       size: 24
33     },
34     xref: 'paper',
35   },

```

```

36|     yaxis: {
37|       tickangle: -45,
38|       title: {
39|         font: {
40|           family: 'Courier-New,monospace',
41|           size: 18,
42|           color: '#000000'
43|         }
44|       }
45|     },
46|     xaxis: {
47|       side:"top",
48|     }
49|   };
50|
51|
52|
53|
54|
55| const config = {responsive: true}
56|
57|
58| function validInput(doAlert=true){
59|   if(startDate.value == ''){
60|     doAlert ? alert('Pilih Tanggal Mulai') : null
61|     return false;
62|   }else if(startTime.value == ''){
63|     doAlert ? alert('Pilih Waktu Mulai') : null
64|     return false;
65|   }else if(endDate.value == ''){
66|     doAlert ? alert('Pilih Tanggal Akhir') : null
67|     return false;
68|   }else if(endTime.value == ''){
69|     doAlert ? alert('Pilih Waktu Akhir') : null
70|     return false;
71|   }else if(selectedKota.length==0){
72|     doAlert ? alert('Pilih Lokasi lebih dahulu'):null
73|     return false;
74|   }else if(intervalInput.value == ''){
75|     doAlert ? alert('Pilih interval') : null
76|     return false;
77|   }else if(optType.value==1){
78|     doAlert ? alert('Pilih tipe grafik') : null
79|     return false;
80|   }else if(sensorSelected[3]==true && optTypeAceleration.value==1){
81|     doAlert ? alert('Pilih tipe grafik untuk akselerasi') : null
82|     return false;
83|   }
84|
85|
86|   if (startDate.value > endDate.value){
87|     doAlert ? alert('Tanggal Selesai Harus Sesudah Tanggal Mulai') : null
88|   }else if(startDate.value == endDate.value){
89|     if(startTime.value >= endTime.value ){
90|       doAlert ? alert('Waktu Selesai Harus Sesudah Waktu Mulai') : null
91|     }
92|   }
93|
94|
95|
96|   return true;
97| }
98|
99|
100|
101| function show3D(jsonData,judul){
102|   layout.yaxis.title.text="g"
103|   layout.title.text=judul+"*"
104|
105|   let tipe=optTypeAceleration.value;
106|
107|
108|   if (optType.value=='box'){
109|     alert("Akselerasi tidak dapat ditampilkan jika pemilihan visualisasi lainnya boxplot")
110|     return null;
111|   }
112|
113|
114|
115|   let data=[];
116|   for (let a=0;a<selectedKota.length;a++){
117|     let dataSensing=[];
118|     let time=[];
119|     let bs=selectedKota[a];
120|     let bsData=jsonData[bs];
121|
122|     let x=[];
123|     let y=[];
124|     let z=[];
125|
126|
127|     for(let i=0;i<bsData.length;i++){
128|       x.push(bsData[i]['akselerasi'][x])
129|       y.push(bsData[i]['akselerasi'][y])
130|       z.push(bsData[i]['akselerasi'][z])
131|       time.push(bsData[i]['timeStamp'])
132|     }
133|   }
134|

```

```

135|     if (tipe!="lines"){
136|       data.push (
137|         {
138|           x: x,
139|           y: y,
140|           z: z,
141|           type: tipe
142|           ,name:selectedKotaName[a]
143|         })
144|     }else{
145|       let Xtrace = {
146|         x: time,
147|         y: x,
148|         name:"X"+selectedKotaName[a],
149|         type: 'lines'
150|       };
151|
152|       let Ytrace = {
153|         x: time,
154|         y: y,
155|         name:"Y"+selectedKotaName[a],
156|         type: 'lines'
157|       };
158|
159|       let Ztrace = {
160|         x: time,
161|         y: z,
162|         name:"Z"+selectedKotaName[a],
163|         type: 'lines'
164|       };
165|
166|       data.push(Xtrace,Ytrace,Ztrace)
167|
168|     }
169|   }
170|
171|   console.log(data)
172|
173| // for (let x=0;x<selectedKota.length;x++){
174| //   let dataSensing=[]
175| //   let time=[]
176| //   let bs=selectedKota[x];
177| //   let bsData=jsonData[bs]
178| //   console.log(bsData)
179| //   for(let i=0;i<bsData.length;i++){
180| //     dataSensing.push(bsData[i][jenisData])
181| //     time.push(bsData[i]['timeStamp'])
182| //   }
183|
184|
185| //   data.push({x:time,y:dataSensing,type:tipe, name:selectedKotaName[x]})  

186| // }
187|
188| // }
189|
190| if(tipe=="scatter"){
191|   data[0].fill='tonexty'
192| }
193|
194|
195|
196|
197|
198|
199| Plotly.newPlot('myPlotakselerasi', data, layout);
200}
201|
202|
203| function show2D(jsonData,judul,jenisData,SelectedKota){
204|   layout.yaxis.title.text=labelsType(jenisData)
205|   layout.title.text=judul
206|
207|   let tipe=optType.value;
208|
209|
210|
211|
212|
213|   if(tipe=='box'){
214|     return box(jsonData,jenisData)
215|   }
216|
217|   let data=[]
218|
219|   for (let x=0;x<selectedKota.length;x++){
220|     let dataSensing=[]
221|     let time=[]
222|     let bs=selectedKota[x];
223|     let bsData=jsonData[bs]
224|
225|     for(let i=0;i<bsData.length;i++){
226|       dataSensing.push(bsData[i][jenisData])
227|       time.push(bsData[i]['timeStamp'])
228|     }
229|
230|     data.push({x:time,y:dataSensing,type:tipe, name:selectedKotaName[x]})  

231|   }
232|
233|

```

```

234 if(tipe=="scatter"){
235   for (let x=0;x<data.length;x++){
236     data[x].fill='tozeroy'
237   }
238 }
239
240 let newLayout={...layout}; //cara copy objek, karena layoutlive merupakan layout utama, sedangkan yang ini akan berubah ubah
241   sesuai tipe data
242
243
244
245 Plotly.newPlot('myPlot'+jenisData, data, newLayout);
246 }
247
248
249
250
251
252 function show(doAlert=true){
253 if (!validInput(doAlert)){
254   return null;
255 }
256
257 let type=document.getElementsByClassName("selected")[0].getAttribute("box")
258 let start='${startDate.value} ${startTime.value}'
259 let end='${endDate.value} ${endTime.value}'
260 let bs=optLokasi.options[optLokasi.selectedIndex].text
261 let tipe=optType.value;
262
263 let jsonString={
264   "node":selectedKota
265   , "start":`${start}:00'
266   , "end":`${end}:00'
267   , "interval":intervalInput.value
268   , "stat":"avg"
269 }
270
271
272
273 // if(selectedKota.length==1){
274 //   jsonString.idBS=selectedKota[0]
275
276 // }
277
278
279
280
281
282
283
284 //khusus untuk boxplot dia raw tidak di aggregasi
285 if(tipe=='box'){
286   jsonString.stat="raw"
287 }
288
289 fetch('${urlAPI}/data',{
290   method: "POST",
291   headers: {
292     Accept: 'application/json',
293     'Content-Type': 'application/json'
294   },
295   body: JSON.stringify(jsonString),
296 }).then(response => response.json() ).then(response=>{
297
298
299
300   if(response['result']=='false'){
301     alert("Data tidak ditemukan")
302   }else{
303
304     responseResult=response['result']
305     let first=true
306
307
308
309     for(let i=0;i<sensorSelected.length;i++){
310       if(sensorSelected[i]){
311
312         if(!first){
313           chartDiv[i].style.display="block"
314           chartDiv[i].style.marginTop="750px"
315         }
316         first=false;
317
318         type=sensorOrder[i]
319         let judul='Grafik ${type} dari ${start} sampai ${end}'
320
321         if(type=="akselerasi"){
322           accInfo.style.display="block"
323           accInfo.style.marginTop="750px"
324           show3D(responseResult,judul)
325         }else{
326           show2D(responseResult,judul,type)
327         }
328       }else{
329
330
331

```

```

332         }
333     }
334   }
335 }
336 }
337 }
338 }).catch((error) => {
339   console.log(error)
340 });
341 });
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 //LISTENER
351 document.getElementById('kotaSelection').addEventListener("change",function(event) {
352   let selectedCity=event.target.value
353   for (let i=0;i<listLokasi.length;i++) {
354     if(listLokasi[i].getAttribute('city')!=selectedCity){
355       listLokasi[i].hidden=true
356     }else{
357       listLokasi[i].hidden=false
358     }
359   }
360   optLokasi.value=-1
361   navSensor.hidden=true
362 });
363 }
364 }
365 }
366 optLokasi.addEventListener("change",function(event) {
367   navSensor.hidden=false
368   let lokasi=event.target.value
369   selectedKota.push(lokasi)
370   for (let i=0;i<selectedLokasi.length;i++){
371     if(selectedLokasi[i].value==lokasi){
372       selectedLokasi[i].hidden=false;
373       selectedKotaName.push(selectedLokasi[i].textContent)
374     }
375   }
376   for (let i=0;i<listLokasi.length;i++){
377     if(listLokasi[i].value==lokasi){
378       listLokasi[i].hidden=true;
379     }
380   }
381   optLokasi.value=-1
382 }
383 }
384 }
385 }
386 }
387 }
388 console.log(selectedKotaName)
389 });
390 }
391 function removeSelected(elem){
392   let lokasi=elem.value
393   let index = selectedKota.indexOf(lokasi);
394   if (index !== -1) {
395     selectedKota.splice(index, 1);
396     selectedKotaName.splice(index,1)
397   }
398   for (let i=0;i<selectedLokasi.length;i++){
399     if(selectedLokasi[i].value==lokasi){
400       selectedLokasi[i].hidden=true;
401     }
402   }
403 }
404 }
405 }
406 for (let i=0;i<listLokasi.length;i++){
407   if(listLokasi[i].value==lokasi){
408     listLokasi[i].hidden=false;
409   }
410 }
411 }
412 console.log(selectedKotaName)
413 }
414 }
415 }
416 }
417 //untuk auto show kalau ganti sensor
418 function select(elem){
419   //pindahin hightlight abu
420   // let selectedBefore=document.getElementsByClassName("selected")
421   // selectedBefore[0].classList.remove("selected")// hapus yang sebelumnya sudah masuk
422   elem.parentNode.classList.toggle("selected")
423   let index=elem.getAttribute("index")
424   sensorSelected[index]=!sensorSelected[index]
425   console.log(sensorSelected)
426 }
```

```

431 //untuk memfilter tipe grafik yang tidak sesuai
432 // let selectedTipe=elem.parentElement.getAttribute("box")
433 // let opsiTipe=optType.children;
434 // for(let i=0;i<opsiTipe.length ;i++){
435
436 // let tujuan=opsiTipe[i].getAttribute("for").split(" ");
437
438 // if( tujuan.includes(selectedTipe)){
439 //   opsiTipe[i].hidden=false;
440 // }else{
441 //   opsiTipe[i].hidden=true;
442 // }
443 // }
444
445 // if(selectedTipe=="akselerasi"){
446 //   optType.selectedIndex=4;
447 // }else{
448 //   if(optType.selectedIndex>=4){
449 //     optType.selectedIndex=1;
450 //   }
451 // }
452
453
454 }
455
456 // optType.addEventListener("change",function(event) {
457 //   show(false)
458 // });

```

Kode F.19: Website - statistic.js

```

1 const optType=document.getElementById("typeSelection");
2 const selectedLokasi=document.getElementsByClassName("selectedLokasi");
3 const scatterConf=document.getElementById('scatter');
4 const xScatter=document.getElementById("scatterX");
5 const yScatter=document.getElementById("scatterY");
6
7 const plotSuhu=document.getElementById("myPlotsuhu")
8 const plotTekanan=document.getElementById("myPlottekanan")
9 const plotKelembapan=document.getElementById("myPlotkelembapan")
10 const supportedSensor=['suhu','kelembapan','tekanan'];
11
12 const summaryConf=document.getElementById("summary")
13 const summaryType=document.getElementById("summaryType")
14 const summaryGraph=document.getElementById("summaryGraph")
15 let graph=document.getElementsByClassName("chart");
16 let selectedKota=[];
17 let selectedKotaName=[];
18
19
20 let layout = {
21   width: document.getElementById('workSheet').offsetWidth,
22   height: (document.getElementById('workSheet').offsetWidth/16)*9,
23   margin: {
24     l: 75,
25     r: 50,
26     b: 50,
27     t: 100,
28     pad: 4
29   },
30   paper_bgcolor: '#c7c7c7',
31   plot_bgcolor: '#ffffff',
32   title : {
33     font: {
34       family: 'Courier\u201dNew,\u201dmonospace',
35       size: 24,
36     },
37     xref: 'paper',
38   },
39   xaxis: {
40     tickangle: -45,
41     title: {
42       font: {
43         family: 'Courier\u201dNew,\u201dmonospace',
44         size: 18,
45         color: '#000000',
46       }
47     }
48   },
49 },
50   xaxis: {
51     automargin: true,
52     side:"top",
53   },
54
55
56
57 };
58
59 function validInput(doAlert=true){
60   if(startDate.value == ''){
61     doAlert ? alert('Pilih Tanggal Mulai') : null
62     return false;
63   }else if(startTime.value == ''){
64     doAlert ? alert('Pilih Waktu Mulai') : null
65     return false;
66   }else if(endDate.value == ''){
67

```

```

68    doAlert ? alert('Pilih Tanggal Akhir') : null
69    return false;
70 }else if(endTime.value == ''){
71    doAlert ? alert('Pilih Waktu Akhir') : null
72    return false;
73 }else if(selectedKota.length==0){
74    doAlert ? alert('Pilih Lokasi lebih dahulu'):null
75    return false;
76 }else if(intervalInput.value == ''){
77    doAlert ? alert('Pilih interval') : null
78    return false;
79 }else if(optType.value==1){
80    doAlert ? alert('Pilih tipe grafik') : null
81    return false;
82 }
83
84 if (startDate.value > endDate.value){
85    doAlert ? alert('Tanggal Selesai Harus Sesudah Tanggal Mulai') : null
86 }else if(startDate.value == endDate.value){
87    if(startTime.value >= endTime.value ){
88        doAlert ? alert('Waktu Selesai Harus Sesudah Waktu Mulai') : null
89    }
90 }
91
92
93
94     return true;
95 }
96
97
98 function showScatter(jsonData){
99     let data=[]
100
101    for (let x=0;x<selectedKota.length;x++){
102        let xValue=[]
103        let yValue=[]
104        let bs=selectedKota[x];
105        let bsData=jsonData[bs]
106
107        for(let i=0;i<bsData.length;i++){
108            xValue.push(bsData[i][xScatter.value])
109            yValue.push(bsData[i][yScatter.value])
110        }
111
112        data.push({x:xValue,y:yValue,type:"scatter", mode: 'markers', name:selectedKotaName[x]})}
113
114    }
115
116
117
118
119 let newLayout={...layout}; //cara copy objek, karena layoutlive merupakan layout utama, sedangkan yang ini akan berubah ubah
120             sesuai tipe data
121 newLayout.yaxis.title.text=(yScatter.value)
122 newLayout.xaxis['title']= {text: xScatter.value, standoff: 2}
123 newLayout.title.text='Grafik Perbandingan antara ${xScatter.value} dengan ${yScatter.value}'
124
125 Plotly.newPlot("myPlotsuhu", data, newLayout);
126
127
128 }
129
130
131 function showMinMax(jsonData,jenisData){
132     layout.yaxis.title.text="Waktu"
133     layout.xaxis['title']= {text: jenisData, standoff: 2}
134     layout.title.text='Grafik Min-max ${jenisData} dengan interval ${intervalInput.value} detik'
135
136     let data=[]
137
138     for (let x=0;x<selectedKota.length;x++){
139         let start=[]
140         let plus=[]
141         let bs=selectedKota[x];
142         let bsData=jsonData[bs]
143         let date=[]
144
145         for(let i=0;i<bsData.length;i++){
146             date.push(bsData[i]['timeStamp'])
147
148
149             let sortedData=(bsData[i][jenisData]).sort()
150             start.push(sortedData[0])
151             plus.push(sortedData[sortedData.length-1]-sortedData[0])
152
153         }
154
155         data.push(
156             {x:plus, //ambah brp
157              base:start, //mulai di mana
158              orientation: 'h',
159              y:date,
160              hovertemplate:'Min: %{base:.2f}' +
161                  '  
Max: %{x:.2f}' +
162                  '  
Time: %{y}' ,
163              type:"bar",
164              mode: 'markers',
165              name:selectedKotaName[x]})}

```

```

166      }
167
168
169
170
171 let newLayout={...layout}; //cara copy objek, karena layoutlive merupakan layout utama, sedangkan yang ini akan berubah ubah
172     sesuai tipe data
173 newLayout.yaxis.title.text="Waktu"
174 newLayout.xaxis['title']={text: jenisData, standoff: 2}
175 newLayout.title.text='Grafik Min-max ${jenisData} dengan interval ${intervalInput.value} detik'
176
177 newLayout.xaxis["type"]="date"
178
179 Plotly.newPlot('myPlot'+jenisData, data, newLayout);
180
181
182
183 function showBox(jsonData,jenisData){
184
185 let data = [];
186
187 for (let x=0;x<selectedKota.length;x++){
188     let dataSensing=[]
189     let time=[]
190     let bs=selectedKota[x];
191     let bsData=jsonData[bs]
192
193
194     for(let i=0;i<bsData.length;i++){
195         let sense=bsData[i][jenisData]
196
197         for (let u=0;u<sense.length;u++){
198             dataSensing.push(bsData[i][jenisData][u])
199         }
200     }
201
202     for (let u=0;u<sense.length;u++){
203         time.push(bsData[i]['timeStamp'])
204     }
205 }
206
207     data.push(
208     {
209         y: dataSensing,
210         x: time,
211         name: selectedKotaName[x],
212         type: "box"
213     }
214 )
215
216 }
217
218 let newLayout={...layout}; //cara copy objek, karena layoutlive merupakan layout utama, sedangkan yang ini akan berubah ubah
219     sesuai tipe data
220 newLayout.yaxis.title.labelsType(jenisData)
221 newLayout.xaxis['title']={text: "waktu", standoff: 1}
222 newLayout.title.text='Grafik Min-max ${jenisData} dengan interval ${intervalInput.value} detik'
223
224 newLayout.xaxis["type"]="log"
225 newLayout.xaxis["type"]="date"
226
227 Plotly.newPlot('myPlot'+jenisData, data, layout);
228
229 }
230
231 function showPieold(jsonData){
232     let colors = ['rgb(42,190,37)', 'rgb(37,107,190)', 'rgb(190,49,37)'];
233
234 for (let p=0;p<selectedKota.length;p++){
235     let bs=selectedKota[p];
236     let bsData=jsonData[bs]
237
238
239     for(let i=0;i<supportedSensor.length;i++){
240         let data=[]
241         let format;
242         let title
243         let addInfo=undefined;
244         if(summaryType.value=="inQ"){
245             format=getInQData(bsData, supportedSensor[i])
246             title='Grafik Perbandingan kuartil data ${supportedSensor[i]}'
247             addInfo='Q1=${Math.round(bsData[supportedSensor[i]]["Q1"]*100)/100}'+
248                 '<br>Q3=${Math.round(bsData[supportedSensor[i]]["Q3"]*100)/100}'
249         }else{
250             format=getStatData(bsData, supportedSensor[i], removeZeros=true)
251             title= 'Grafik rasio data ${supportedSensor[i]}'
252         }
253
254         data.push({values: format[0], labels:format[1]
255             ,type:"pie"
256             ,textinfo:"label+percent+value"
257             ,textposition:"outside"
258             ,marker: {
259                 colors: colors
260             },
261             rotation: 90,
262         })
263     }
264 }
265
266 
```

```

263    })
264
265    let newLayout={...layout}; //cara copy objek, karena layoutlive merupakan layout utama, sedangkan yang ini akan berubah ubah
266        sesuai tipe data
267    newLayout.title.text=title
268
269    if(addInfo!=undefined){
270        newLayout['annotations']= [
271            {
272                text: addInfo,
273                showarrow: false,
274                x: 1.0,
275                y: 0.5,
276                font: {
277                    family: 'Arial',
278                    size: 14,
279                    color: 'black'
280                }
281            }
282        ]
283        Plotly.newPlot("myPlot"+supportedSensor[i], data, newLayout);
284    }
285}
286
287
288 function showPie(jsonData, jenisData){
289     let colors = ['rgb(42,190,137)', 'rgb(37,107,190)', 'rgb(190,49,137)'];
290
291     let data=[];
292     let addInfo="";
293     let midText=[];
294
295     let row=0;
296     let col=0;
297     let numberOfRow=round(selectedKota.length/2)
298     let distanceY=(1-0.4)/numberOfRow // 0.4 itu pad atas+bawah
299
300     for (let p=0;p<selectedKota.length;p++){
301         let bs=selectedKota[p];
302         let bsData=jsonData[bs]
303
304         if(col==2){
305             col=0;
306             row=row+1
307         }
308
309
310         if(summaryType.value=="inQ"){
311             format=getInQData(bsData, jenisData)
312             title='Grafik Perbandingan kuartil data ${jenisData}'
313             addInfo+=`<br>Kuartil untuk ${selectedKotaName[p]}'+
314                 '<br>Q1=${Math.round(bsData [jenisData]["Q1"]*100)/100}'+
315                 '<br>Q3=${Math.round(bsData [jenisData]["Q3"]*100)/100}'
316
317         }else{
318             format=getStatData(bsData, jenisData, removeZeros=true)
319             title= 'Grafik rasio data ${jenisData}'
320         }
321
322         let posX=col==0?0.20:0.80
323         let poxy=row==0?0.80:0.20
324
325         midText.push({font: {
326             size: 20
327         },
328         text: selectedKotaName[p],
329         showarrow: false,
330         x: posX,
331         y: poxy
332     })
333
334
335
336
337
338
339
340         data.push({values: format[0], labels:format[1]
341             ,type:"pie"
342             ,textinfo:"label+percent+value"
343             ,textposition:"outside"
344             ,marker: {
345                 colors: colors
346             },
347             rotation: 90,
348             domain: {
349                 row: row,
350                 column: col
351             },
352             name: selectedKotaName[p]
353         })
354         col+=1
355     }
356     let newLayout={...layout}; //cara copy objek, karena layoutlive merupakan layout utama, sedangkan yang ini akan berubah ubah
357         sesuai tipe data
358     newLayout.title.text=title
359     newLayout['grid']={rows:row+1,columns:2}

```

```

360
361     if(selectedKota.length==1){
362         newLayout.grid.columns=1 //biar ketengah
363     }
364
365     if(addInfo!=undefined){
366         newLayout['annotations']= [
367             {
368                 text: addInfo,
369                 showarrow: false,
370                 x: 0.5,
371                 y: 1.0,
372                 font: {
373                     family: 'Arial',
374                     size: 14,
375                     color: 'black'
376                 }
377             }
378         ]
379
380         for (let i=0;i<midText.length;i++){
381             newLayout.annotations.push(midText[i])
382         }
383         newLayout.annotations.push(midText)
384         console.log(newLayout)
385     }
386     Plotly.newPlot("myPlot"+jenisData, data, newLayout);
387
388 }
389
390 function showHistogram(jsonData,jenisData,gap=0){
391
392     let data = [];
393
394     for (let x=0;x<selectedKota.length;x++){
395         let bs=selectedKota[x];
396         let bsData=jsonData[bs]
397
398         let sense=bsData[jenisData]
399         let format;
400         if(summaryType.value=="inQ"){
401             format=getInQData(sense,undefined)
402             addInfo='Q1=${Math.round(bsData [jenisData]["Q1"]*100)/100}'+
403                 '<br>Q3=${Math.round(bsData [jenisData]["Q3"]*100)/100}'
404         }else{
405             format=getStatData(sense,undefined,false)
406         }
407
408         data.push(
409             {
410                 y: format[0],
411                 x: format[1],
412                 name: selectedKotaName[x],
413                 type: "bar",
414             }
415         )
416     }
417 }
418
419
420 let newLayout={...layout}; //cara copy objek, karena layoutlive merupakan layout utama, sedangkan yang ini akan berubah ubah
421     sesuai tipe data
422
423 newLayout.title.text='Grafik Histogram untuk ${jenisData}'
424
425 newLayout.yaxis.title.text="Frekuensi"
426 newLayout.xaxis['title']= {text: labelsType(jenisData), standoff: 1}
427
428 newLayout.xaxis.side="bottom"
429 newLayout.yaxis["type"]="linear"
430 newLayout.xaxis["type"]="category"
431 newLayout['bargap']=gap
432
433
434 if(summaryType.value=="inQ"){
435     newLayout.title.text='Grafik Histogram interkuartil untuk ${jenisData}'
436 }
437
438
439 Plotly.newPlot('myPlot'+jenisData, data, newLayout);
440 }
441
442
443 function getInQData(indata,tipe){
444     let x;
445     if(tipe!=undefined){
446         x=[indata[tipe]["numQ1"], indata[tipe]["numQ2"],indata[tipe]["numQ3"]]
447     }else{
448         x=[indata["numQ1"], indata["numQ2"],indata["numQ3"]]
449     }
450
451
452     let y=["Q1", "Q2", "Q3"]
453
454     return [x,y]
455 }
456
457 function getStatData(indata,tipe, removeZeros=false){

```

```

458 let x;
459 if(tipe!=undefined){
460     x=indata[tipe]
461 }else{
462     x=indata
463 }
464
465 let incerement= (100/x.length)
466 let y=[];
467 for (let i=1;i<=100;i+=incerement){
468     y.push(i+"▬▬"+(i+incerement))
469 }
470
471
472
473
474 if(removeZeros){
475     for (let i=0;i<100;i++){
476         let pos=x.indexOf(0)
477         console.log(pos)
478         if(pos>-1){
479             x.splice(pos,1)
480             y.splice(pos,1)
481         }
482     }
483 }
484 return [x,y]
485 }
486
487
488
489
490 function show(doAlert=true){
491
492
493 if (!validInput(doAlert)){
494     return null;
495 }
496
497 for (let i=0;i<graph.length;i++){
498     graph[i].style.display="none";
499 }
500
501
502 let start='${startDate.value} ${startTime.value}'
503 let end='${endDate.value} ${endTime.value}'
504 let bs=optLokasi.options[optLokasi.selectedIndex].text
505 let tipe=optType.value;
506
507
508 let jsonString={
509     "node":selectedKota
510     , "start":`${start}:00'
511     , "end":`${end}:00'
512     , "interval":intervalInput.value
513     , "stat":"avg"
514 }
515
516
517 if(tipe=='box' || tipe=="boxFloating"){
518     jsonString.stat="raw"
519 }else if(tipe=="summary"||tipe== "split-1" ||
520          tipe=="split-2"||tipe== "split-5" ||
521          tipe=="split-10"||tipe== "split-20"
522 ){
523     jsonString.stat=summaryType.value
524 }
525
526
527 fetch('${urlAPI}/data',{
528     method: "POST",
529     headers: {
530         Accept: 'application.json',
531         'Content-Type': 'application/json'
532     },
533     body: JSON.stringify(jsonString),
534 }).then(response => response.json() .then(response=>{
535
536
537
538 if(response['result']=='false'){
539     alert("Data tidak ditemukan")
540 }else{
541     responseResult=response['result']
542     if(tipe=="scatter"){
543         graph[0].style.display="block";
544         Plotly.purge(document.getElementById("myPlotsuhu"))
545         Plotly.purge(document.getElementById("myPlottekanan"))
546         Plotly.purge(document.getElementById("myPlotkelembapan"))
547         showScatter(responseResult)
548     }else if(tipe=="boxFloating"){
549         for (let m=0; m <supportedSensor.length;m++){
550             graph[(m)].style.display="block";
551             new showMinMax(responseResult,supportedSensor[m])
552         }
553     }else if(tipe=="box"){
554         for (let m=0; m <supportedSensor.length;m++){
555             graph[(m)].style.display="block";
556         }
557     }
558 }
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
26
```

```

557             new showBox(responseResult,supportedSensor[m]);
558         }
559     }else if(type=="summary"){
560         for (let m=0; m <supportedSensor.length;m++){
561             graph[(m)].style.display="block";
562         }
563     if(summaryGraph.value=="pie"){
564         for (let m=0; m <supportedSensor.length;m++){
565             graph[(m)].style.display="block";
566             showPie(responseResult,supportedSensor[m])
567         }
568     }else if(summaryGraph.value=="histogram"){
569         for (let m=0; m <supportedSensor.length;m++){
570             graph[(m)].style.display="block";
571             new showHistogram(responseResult,supportedSensor[m]);
572         }
573     }else if(summaryGraph.value=="box"){
574         for (let m=0; m <supportedSensor.length;m++){
575             graph[(m)].style.display="block";
576             new showHistogram(responseResult,supportedSensor[m],0.7);
577         }
578     }
579 }
580
581 }
582
583 }).catch((error) => {
584     console.log(error)
585 });
586 }
587 }
588
589
590
591 optLokasi.addEventListener("change",function(event) {
592     let lokasi=event.target.value
593
594     selectedKota.push(lokasi)
595
596     for (let i=0;i<selectedLokasi.length;i++){
597         if(selectedLokasi[i].value==lokasi){
598             selectedLokasi[i].hidden=false;
599             selectedKotaName.push(selectedLokasi[i].textContent)
600         }
601     }
602
603     for (let i=0;i<listLokasi.length;i++){
604         if(listLokasi[i].value==lokasi){
605             listLokasi[i].hidden=true;
606         }
607     }
608
609     optLokasi.value=-1
610
611
612     console.log(selectedKotaName)
613 });
614
615
616
617 function removeSelected(elem){
618     let lokasi=elem.value
619     let index = selectedKota.indexOf(lokasi);
620     if (index != -1) {
621         selectedKota.splice(index, 1);
622         selectedKotaName.splice(index,1)
623     }
624
625     for (let i=0;i<selectedLokasi.length;i++){
626         if(selectedLokasi[i].value==lokasi){
627             selectedLokasi[i].hidden=true;
628         }
629     }
630
631
632     for (let i=0;i<listLokasi.length;i++){
633         if(listLokasi[i].value==lokasi){
634             listLokasi[i].hidden=false;
635         }
636     }
637 }
638
639
640 optType.addEventListener("change",function(event) {
641     console.log(event.target.value)
642     scatterConf.hidden=true
643     switch (event.target.value){
644         case "scatter":
645             scatterConf.hidden=false
646             break;
647         case "summary":
648             summaryConf.hidden=false;
649             break;
650     }
651
652
653
654
655

```

```

656
657
658
659 });

```

Kode F.20: Website - table.js

```

1 const tabel=document.getElementById('tabeldorf')
2
3
4 function validInput(doAlert=true){
5     if(startDate.value == ''){
6         doAlert ? alert('Pilih Tanggal Mulai') : null
7         return false;
8     }else if(startTime.value == ''){
9         doAlert ? alert('Pilih Waktu Mulai') : null
10        return false;
11    }else if(endDate.value == ''){
12        doAlert ? alert('Pilih Tanggal Akhir') : null
13        return false;
14    }else if(endTime.value == ''){
15        doAlert ? alert('Pilih Waktu Akhir') : null
16        return false;
17    }else if(intervalInput.value == ''){
18        doAlert ? alert('Pilih interval') : null
19        return false;
20    }else if(optLokasi.value == ''){
21        doAlert ? alert('Pilih Lokasi') : null
22        return false;
23    }
24
25    if (startDate.value > endDate.value){
26        doAlert ? alert('Tanggal Selesai Harus Sesudah Tanggal Mulai') : null
27    }else if(startDate.value == endDate.value){
28        if(startTime.value >= endTime.value ){
29            doAlert ? alert('Waktu Selesai Harus Sesudah Waktu Mulai') : null
30        }
31    }
32
33
34    return true;
35 }
36
37
38
39
40 var bodytable=undefined;
41
42 function show(doAlert=true){
43     if (!validInput(doAlert)){
44         return null;
45     }
46
47     let start='${startDate.value} ${startTime.value}'
48     let end='${endDate.value}-${endTime.value}'
49     let bs=optLokasi.options[optLokasi.selectedIndex].text
50
51
52     let jsonString={
53         "node":`${optLokasi.value}`,
54         "start":`${start}:00`,
55         "end":`${end}:00`,
56         "interval":`${intervalInput.value}`,
57         'stat':'avg'
58     }
59
60
61
62
63
64     fetch(`${urlAPI}/data',{
65         method: "POST",
66         headers: {
67             Accept: 'application/json',
68             'Content-Type': 'application/json'
69         },
70         body: JSON.stringify(jsonString),
71     }).then(response => response.json() .then(response=>{
72
73
74         if(response['result']=='false'){
75             alert("Data tidak ditemukan")
76         }else{
77             //buang yang sudah ada
78
79             if(bodytable!=undefined){
80                 bodytable.remove()
81             }
82
83
84             bodytable=document.createElement("tbody")
85
86             tabel.parentElement.style.display="block"
87
88             responseResult=response['result']
89             console.log(responseResult[0]['timeStamp'])
90
91

```

```

92         for(let i=0;i<responseResult.length;i++){
93             let row=document.createElement('tr')
94
95             let time=document.createElement('td')
96             time.textContent=responseResult[i]['timeStamp']
97             time.classList.add("tabelData")
98             row.appendChild(time)
99
100            let suhu=document.createElement('td')
101            suhu.textContent=responseResult[i]['suhu']+°C
102            suhu.classList.add("suhu")
103            row.appendChild(suhu)
104
105            let kelembapan=document.createElement('td')
106            kelembapan.textContent=responseResult[i]['kelembapan']+%
107            kelembapan.classList.add("tabelData")
108            row.appendChild(kelembapan)
109
110            let tekanan=document.createElement('td')
111            tekanan.textContent=responseResult[i]['tekanan']+kPa
112            tekanan.classList.add("tabelData")
113            row.appendChild(tekanan)
114
115            let akselerasi=document.createElement('td')
116            akselerasi.textContent=(+responseResult[i]['akselerasi'][x]+,++responseResult[i]['akselerasi'][y]+
117                ,,+responseResult[i]['akselerasi'][z]+)
118            akselerasi.classList.add("tabelData")
119            row.appendChild(akselerasi)
120
121            bodytable.appendChild(row)
122        }
123
124        tabel.appendChild(bodytable)
125
126        // tabel
127
128        accInfo.style.display="block"
129
130
131    }
132
133
134
135    }).catch((error) => {
136        console.log(error)
137    });
138}

```

Kode F.21: Website - admin.css

```

1 .partBox{
2     background-color:#antiquewhite;
3     padding:20px,10px;
4     border-radius:10px;
5     margin:10px,0px;
6 }
7
8 .conf{
9     height:40px;
10    vertical-align:middle,!important;
11 }
12
13 label.conf{
14     padding-top:10px;
15 }
16
17 input[type=checkbox].indoor{
18     margin-top:10px;
19 }

```

Kode F.22: Website - layout.css

```

1 @import url('https://fonts.googleapis.com/css?family=Prata&family=Rokkitt:wght@300;400;500;600;700;800&display=swap');
2
3 *{
4     margin:0px;
5     padding:0px;
6     font-family:rokkitt;
7 }
8
9 body{
10     background-color:#F0D1A2;
11 }
12
13
14
15 .menuAtas {
16     background-color:#antiquewhite;
17 }
18
19
20 .debug{
21     border:1px solid red;
22 }

```

```

23| #logo{
24|   margin: -5px 0px;
25|   height: 9vh;
26|
27}
28
29#title{
30  color: #555154;
31
32  line-height: 1.5;
33  font-weight: 500;
34  font-size: 38px;
35  height: 5vh;
36}

```

Kode F.23: Website - livePage.css

```

1 #lokSelect{
2   background-color: white;
3   border-radius: 8px;
4 }
5
6 .aClean{
7   color: black;
8 }
9
10 .liveData{
11   width: 300px;
12   height: 152px;
13   display: inline-block !important;
14 }
15
16
17 .liveDataTitle{
18   font-size: 18pt;
19 }
20
21 .liveDataText{
22   font-size: 16pt;
23   display: inline-block;
24 }
25
26 .midData{
27   padding-left: 10px;
28 }
29
30
31 .liveSelection{
32   height: 100px !important;
33 }
34
35 #suhuSelection, #kelembapanSelection, #tekananSelection, #akselerasiSelection{
36   min-width: 250px !important;
37 }

```

Kode F.24: Website - login.css

```

1 .login{
2   width: 30% !important;
3 }
4
5 #gambar{
6   width: 40vw;
7   position: absolute;
8   left: 80px;
9   bottom: 25vh;
10}

```

Kode F.25: Website - mainUser.css

```

1 #mainLayout{
2   margin-top: 20px;
3   background-color: white;
4   border: 1px solid red;
5
6   min-height: 200px;
7 }
8
9 .container{
10  margin-top: 1vh;
11 }
12
13 .waktu{
14  margin-top: 0.4vw;
15 }
16

```

Kode F.26: Website - navigation.css

```

1 span svg {
2

```

```
3   color :#555154;
4  /* color:white; */
5
6  width:30px;
7  height:30px;
8 }
9
10 ul {
11   list-style-type:@"";
12 }
13
14 #navigation{
15   background-color:#antiquewhite;
16 }
17
18
19 .selected{
20   background-color:#aaaaaa;
21 }
22
23 .nonClick{
24   margin-left:2vw;
25 }
```