

**PENGEMBANGAN *DASHBOARD* RESPONSIF BERBASIS  
*WEB* UNTUK BANK AIR KAMI**



Disusun Oleh:

N a m a : Daud Hidayat Ramadhan  
NIM : 21523089

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
2026**



## HALAMAN PENGESAHAN DOSEN PENGUJI

**PENGEMBANGAN *DASHBOARD* RESPONSIF BERBASIS  
*WEB* UNTUK BANK AIR KAMI**

## TUGAS AKHIR

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 14 Januari 2026

Tim Penguji

Ari Sujarwo, S.Kom, M.I.T.

**Anggota 1**

Arrie Kurniawardhani, S.Si., M.Kom.

**Anggota 2**

Dr. Syarif Hidayat, S.Kom, M.I.T.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Ir. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. IPM., ASEAN Eng)

## HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Daud Hidayat Ramadhan

NIM : 21523089

Tugas akhir dengan judul:

### **PENGEMBANGAN *DASHBOARD* RESPONSIF BERBASIS *WEB* UNTUK BANK AIR KAMI**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 14 Januari 2026



Daud Hidayat Ramadhan

## **HALAMAN PERSEMBAHAN**

Alhamdulillahirabbil‘alamin, puji syukur ke hadirat Allah SWT atas segala nikmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini. Shalawat serta salam semoga senantiasa tercurah kepada Nabi Muhammad SAW sebagai suri teladan bagi umat manusia. Karya ini penulis persembahkan kepada keluarga tercinta atas doa, dukungan, dan kasih sayang yang tidak pernah putus; kepada dosen pembimbing atas arahan, kesabaran, dan bimbingan yang sangat berarti; serta kepada teman-teman seperjuangan yang telah kebersamai, membantu, dan menguatkan penulis selama proses penyusunan. Semoga skripsi ini membawa manfaat, menjadi amal kebaikan, dan memberikan keberkahan bagi semua pihak yang telah berperan dalam perjalanan ini.

## HALAMAN MOTO

*“Allah tidak membebani seseorang, kecuali menurut kesanggupannya.”*

QS. Al-Baqarah [2]: 286

*“Siapa yang bertawakal kepada Allah, Allah akan mencukupkannya.”*

QS. At-Talaq [65]: 3

## KATA PENGANTAR

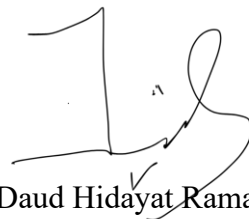
Puji syukur kehadirat Allah SWT atas rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Pengembangan *Dashboard* Responsif Berbasis *Web* Untuk Bank Air Kami”. Penulisan tugas akhir ini disusun sebagai salah satu syarat untuk menyelesaikan studi pada Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Dalam proses penyusunan, penulis menghadapi beberapa kendala, terutama dalam penyesuaian kebutuhan sistem dengan alur proses bisnis serta konsistensi penyusunan dokumen penelitian. Namun, kendala tersebut dapat diatasi berkat bantuan dan dukungan berbagai pihak. Penulis menyampaikan terima kasih kepada:

1. Bapak Ari Sujarwo selaku dosen pembimbing atas arahan, masukan, dan bimbingan selama proses penelitian.
2. Pihak Bank Air Kami serta peneliti sebelumnya terkait yang telah membantu memberikan dukungan data dan konteks kebutuhan sistem.
3. Keluarga serta rekan-rekan yang senantiasa memberikan doa dan dukungan.

Penulis berharap hasil penelitian ini dapat memberikan manfaat sebagai referensi pengembangan sistem monitoring kualitas air serta mendukung peningkatan proses bisnis pengelolaan layanan filter air.

Yogyakarta, 14 Januari 2026



Daud Hidayat Ramadhan

## SARI

Saat ini kondisi operasional bank air kami sedang dihentikan sementara dikarenakan adanya aktivitas pembangunan di sekitar Sungai Code. Penelitian ini bertujuan merancang dan mengembangkan dasbor responsif berbasis *web* yang mengintegrasikan data sensor IoT serta mengimplementasikan proses bisnis usulan pengelolaan filter air sebagai perluasan dari alur layanan yang sudah berjalan. Dasbor mencakup pengelolaan titik pemantauan, penyajian parameter kualitas air (pH, suhu, kekeruhan, dan TDS), pemetaan berbasis GIS, manajemen pelanggan dan filter (stok, pemasangan, serta riwayat penggunaan), serta pengelolaan pengguna berbasis peran (*Role-Based Access Control*). Penelitian pengembangan ini menggunakan model *waterfall* yang dimodifikasi, meliputi analisis kebutuhan, perancangan, implementasi menggunakan Laravel dan Filament, integrasi data melalui layanan *cloud*, serta pengujian sistem. Hasil pengembangan menunjukkan dasbor berjalan pada berbagai ukuran layar (*mobile, tablet, dan desktop*) dan mampu menerapkan pembatasan akses sesuai peran pengguna. Pengujian unit dan fitur menghasilkan 772 pengujian dengan 3.034 asersi yang seluruhnya lulus, sedangkan pengujian fungsional berbasis *black-box* mencakup 54 kasus uji dengan hasil memenuhi kriteria. Namun, karena pengujian menggunakan data sekunder dan data simulasi serta infrastruktur IoT tidak aktif secara waktu nyata, kesesuaian sistem terhadap proses bisnis usulan masih dinilai pada tingkat rancangan dan simulasi. Dengan demikian, dasbor ini dapat menjadi referensi awal bagi Bank Air Kami dalam mengadopsi dan mengimplementasikan perluasan proses bisnis pengelolaan filter air dari alur layanan yang telah berjalan.

Kata kunci: dasbor responsif, *monitoring* kualitas air, GIS, RBAC.

## GLOSARIUM

Cloud Computing	Paradigma komputasi berbasis internet yang menyediakan sumber daya seperti penyimpanan, pemrosesan, dan layanan aplikasi secara elastis sesuai kebutuhan sistem.
Continuous Delivery	Praktik pengembangan perangkat lunak yang memastikan aplikasi selalu berada dalam kondisi siap rilis melalui otomatisasi proses rilis dan validasi, sehingga pengiriman pembaruan dapat dilakukan secara lebih cepat dan konsisten.
Continuous Integration	Praktik penggabungan perubahan kode secara berkala ke repositori utama yang disertai pengujian otomatis untuk mendeteksi kesalahan lebih awal dan menjaga kualitas perangkat lunak.
Geographic Information System	Sistem yang digunakan untuk mengelola, menganalisis, dan menampilkan data berbasis lokasi, sehingga informasi kualitas air dapat dipetakan dan dipahami secara spasial.
Internet of Things	Konsep integrasi perangkat fisik seperti sensor dengan jaringan internet untuk mengumpulkan, mengirim, dan memantau data secara otomatis, termasuk data parameter kualitas air.
Minimum Viable Product	Versi awal produk dengan fitur inti minimum yang sudah cukup untuk diuji dan dievaluasi guna memastikan kesesuaian solusi dengan kebutuhan pengguna dan proses bisnis.
Responsive Web Design	Pendekatan perancangan antarmuka <i>web</i> agar tampilan dan fungsi aplikasi tetap optimal pada berbagai ukuran layar, seperti ponsel, tablet, dan desktop.
Role-based Access Control	Metode pengaturan hak akses pengguna berdasarkan peran tertentu dalam sistem, sehingga setiap pengguna hanya dapat mengakses fitur sesuai tanggung jawabnya.
Total Dissolved Solids	Parameter kualitas air yang menunjukkan jumlah zat padatan

terlarut dalam air dan sering digunakan sebagai indikator kelayakan atau perubahan kualitas air.

#### Turbiditas

Parameter kualitas air yang menggambarkan tingkat kekeruhan akibat partikel tersuspensi, yang dapat memengaruhi kejernihan dan mutu air.

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR .....	iv
HALAMAN PERSEMBAHAN .....	v
HALAMAN MOTO .....	vi
KATA PENGANTAR .....	vii
SARI .....	viii
GLOSARIUM.....	ix
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR .....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian .....	4
1.5 Manfaat Penelitian .....	4
1.6 Metodologi Penelitian.....	5
1.6.1 Analisis Kebutuhan .....	5
1.6.2 Perencanaan .....	5
1.6.3 Implementasi.....	5
1.6.4 Pengujian Aplikasi .....	6
1.6.5 Evaluasi dan Dokumentasi.....	6
1.7 Sistematika Penulisan .....	6
BAB II KAJIAN PUSTAKA.....	7
2.1 Air Bersih Layak Konsumsi.....	7
2.2 Standar Kualitas Air.....	7
2.3 Tata Kelola Infrastruktur.....	9
2.4 Proses Pengelolaan AMDK .....	12
2.5 Filter Air.....	16
2.5.1 Filter Fisik .....	17

2.5.2	Filter Adsorpsi.....	18
2.5.3	Filter Pertukaran Ion / Demineralization .....	18
2.5.4	Filter Membran (MF/UF/NF/RO).....	18
2.5.5	Filter Pelengkap (Disinfeksi UV) .....	18
2.6	Teknologi Pemantauan & Dasbor.....	19
2.7	<i>Responsive Web Design</i> (RWD) dalam Konteks Pengembangan Antarmuka <i>Multiplatform</i> .....	20
2.8	<i>Cloud Computing</i> Dalam Pengembangan Aplikasi <i>Web</i> .....	21
BAB III METODOLOGI.....		24
3.1	Kebutuhan sistem.....	25
3.1.1	Gambaran Sistem .....	25
3.1.2	Analisis Kebutuhan Sistem .....	26
3.2	Perencanaan .....	29
3.2.1	<i>Minimum Viable Product</i> (MVP) Penelitian.....	29
3.2.2	Arsitektur Perangkat lunak.....	30
3.2.3	Arsitektur dan Desain <i>Database</i> .....	31
3.2.4	Desain Infrastruktur .....	32
3.2.5	Rancangan Desain Antarmuka.....	33
3.3	Rencana Implementasi .....	42
3.3.1	Lingkungan Pengembangan .....	42
3.3.2	Manajemen Versi & CI/CD.....	43
3.3.3	Setup Infrastruktur .....	44
3.4	Pengujian.....	44
3.4.1	Pengujian Fungsional .....	44
3.4.2	Pengujian Non-Fungsional.....	46
3.5	Evaluasi dan Dokumentasi .....	47
BAB IV HASIL DAN PEMBAHASAN .....		48
4.1	Hasil Implementasi Sistem .....	48
4.1.1	Implementasi <i>Database</i> .....	48
4.1.2	Implementasi Aplikasi Website.....	50
4.1.3	Implementasi Infrastruktur & Automasi .....	53
4.2	Hasil Pengujian Aplikasi.....	55
4.2.1	Pengujian Fungsional .....	55
4.2.2	Pengujian Non-Fungsional .....	57

4.3	Pembahasan.....	57
4.3.1	Dasbor Responsif Untuk Pemantauan Kualitas Air.....	57
4.3.2	Mekanisme Simulasi Data IoT .....	59
4.3.3	Perluasan Sistem Untuk Proses Bisnis Pengelolaan Filter .....	61
BAB V KESIMPULAN DAN SARAN .....		68
5.1	Kesimpulan .....	68
5.2	Saran .....	68
DAFTAR PUSTAKA .....		69
LAMPIRAN.....		72

**DAFTAR TABEL**

Tabel 2.1 Perbandingan Standar Air Minum .....	8
Tabel 2.2 Perbandingan Filter air tradisional dan modern.....	15
Tabel 2.3 Perbandingan Deployment di <i>Cloud Computing</i> .....	22
Tabel 3.1 Komponen Aplikasi yang Digunakan.....	43
Tabel 3.2 Spesifikasi Infrastruktur.....	44
Tabel 3.3 Daftar <i>Test-case</i> untuk Pengujian Fungsional .....	45
Tabel 3.4 Daftar Pengujian Non-Fungsional .....	46
Tabel 4.1 Daftar Fitur yang Terimplementasi.....	50
Tabel 4.2 Daftar Ringkasan <i>Test</i> Secara Umum.....	55
Tabel 4.3 Daftar Ringkasan <i>Test</i> Secara Fungsional .....	56
Tabel 4.4 Daftar Ringkasan <i>Test</i> Secara Non-Fungsional .....	57
Tabel 4.5 Rangkuman Perubahan Proses Bisnis.....	62
Tabel 4.6 Rangkuman Perbandingan Sistem Berdasarkan Fitur .....	63

## DAFTAR GAMBAR

Gambar 2.1 Tren Distribusi Sumber Air (BPS 2018-2021) .....	12
Gambar 2.2 Proses pengelolaan air minum konvensional .....	13
Gambar 2.3 Proses Pengelolaan Air minum Modern Dengan Ultrafiltrasi .....	14
Gambar 2.4 Proses Pengelolaan Air Minum Modern Tanpa Ultrafiltrasi .....	14
Gambar 2.5 Penggunaan GIS untuk Tata Kelola Air .....	19
Gambar 3.1 Model Pengembangan <i>Waterfall</i> yang Dimodifikasi.....	24
Gambar 3.2 Gambaran IoT Sistem dari Penelitian Sebelumnya .....	26
Gambar 3.3 Diagram BPMN Proses Bisnis Usulan Pengelolaan Filter Air pada Bank Air Kami.....	27
Gambar 3.4 Arsitektur MVC .....	31
Gambar 3.5 Desain <i>Database</i> SQL.....	32
Gambar 3.6 Desain <i>Database</i> Dokumen .....	32
Gambar 3.7 Desain Infrastruktur Cloud.....	33
Gambar 3.8 Halaman Login.....	34
Gambar 3.9 Halaman Dasbor Utama .....	35
Gambar 3.10 Halaman Indeks Filter Air Pengguna .....	36
Gambar 3.11 Halaman Indeks filter Air.....	37
Gambar 3.12 Halaman Riwayat Penggantian Filter.....	38
Gambar 3.13 Halaman Riwayat Kualitas Air.....	39
Gambar 3.14 Halaman Indeks Sensor .....	40
Gambar 3.15 Halaman Indeks Stok .....	41
Gambar 3.16 Halaman Indeks User .....	42
Gambar 4.1 ERD Implementasi .....	49
Gambar 4.2 Struktur <i>Database</i> Dokumen di Firebase.....	50
Gambar 4.3 <i>Visual Table of Content</i> (VTOC) Aplikasi Hasil Implementasi .....	52
Gambar 4.4 Hasil Implementasi Pipeline Github Action.....	53
Gambar 4.5 Dasbor ukuran 768 px ( <i>tablet</i> ) .....	59
Gambar 4.6 Dasbor ukuran 480 px ( <i>mobile</i> ).....	59
Gambar 4.7 Perbandingan Mekanisme Ideal (kiri) dan Mekanisme Simulasi (kanan) .....	60
Gambar 4.8 Perbandingan Struktur <i>Database</i> Penelitian Sebelumnya (kiri) dan Sekarang (kanan) .....	61
Gambar 4.9 Pemantauan Kualitas Air pada Sistem Sebelumnya .....	64

Gambar 4.10 Pemantauan Kualitas Air pada Sistem yang Dikembangkan.....	65
Gambar 4.11 Riwayat Kualitas Air pada Sistem Sebelumnya .....	66
Gambar 4.12 Riwayat Kualitas Air pada Sistem yang Dikembangkan.....	66
Gambar 4.13 Pemasangan Filter Air pada Sistem Sebelumnya .....	67
Gambar 4.14 Pemasangan Filter Air pada Sistem yang Dikembangkan.....	67

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Kampung Terban di Yogyakarta merupakan salah satu kawasan perkotaan informal yang menghadapi tantangan besar dalam pengelolaan sumber daya air bersih. Sebagai wilayah yang padat penduduk dan berbatasan langsung dengan Sungai Code, kebutuhan air bersih menjadi salah satu isu utama bagi masyarakat. Sejak berdirinya, Bank Air Kami telah menjadi pusat berbagai penelitian yang berkontribusi pada pengembangan pengelolaan sumber daya air di kawasan ini. Mata air di kawasan ini telah dimanfaatkan sebagai sumber utama air minum warga, tetapi kualitas air sangat dipengaruhi oleh kondisi lingkungan seperti curah hujan, banjir, serta pencemaran dari aktivitas domestik. Selain itu, keterbatasan infrastruktur dasar di Kampung Terban menambah kompleksitas masalah air bersih. Banyak warga mengandalkan sistem tradisional untuk memastikan pasokan air, yang sering kali tidak cukup untuk memenuhi kebutuhan seluruh masyarakat.

Pada tahun 2013, inisiatif pendirian Bank Air Kami di Kampung Terban dimulai sebagai upaya mengelola sumber daya air dengan lebih efektif. Proyek ini bertujuan tidak hanya untuk menyediakan air minum berkualitas tinggi melalui teknologi filtrasi canggih, tetapi juga menciptakan ekosistem berbasis komunitas untuk pengelolaan sumber daya secara kolektif. Bank Air Kami menjadi contoh nyata kolaborasi antara masyarakat lokal, akademisi, dan pemangku kepentingan lain dalam menciptakan solusi inovatif untuk masalah air. Selain menyediakan fasilitas pemurnian air, proyek ini memanfaatkan teknologi *Internet of Things* (IoT) untuk memantau kualitas air secara *realtime*. Sistem ini mencatat parameter krusial seperti pH, kekeruhan, dan suhu yang memengaruhi kelayakan konsumsi air. Data tersebut kemudian diproses dan divisualisasikan untuk memberikan informasi relevan kepada masyarakat serta pemangku kepentingan. Hingga kini, Bank Air Kami telah mengimplementasikan sistem filtrasi modern berbasis *reverse osmosis* dan ozonisasi, serta mengintegrasikan data melalui Firebase Realtime Database untuk mempermudah pengelolaan informasi berbasis *cloud*.

Namun, meskipun Bank Air Kami telah memberikan dampak positif yang signifikan, sejumlah tantangan tetap muncul. Sistem monitoring yang ada menghadapi keterbatasan seperti gangguan sinyal, kapasitas perangkat keras yang tidak memadai, dan kurangnya kemampuan untuk menyajikan data secara intuitif kepada pengguna. Kesulitan dalam

memastikan distribusi data yang akurat dan tepat waktu kepada masyarakat juga menjadi kendala utama, sehingga memengaruhi pemanfaatan informasi untuk pengambilan keputusan.

Selain itu, keberlanjutan proyek menghadapi tekanan dari sisi pendanaan dan pengelolaan jangka panjang. Untuk menjawab kebutuhan air bersih yang terus meningkat akibat pertumbuhan populasi dan perubahan iklim, diperlukan solusi yang lebih adaptif. Dalam perjalanan pengembangannya, Bank Air Kami juga telah menciptakan aplikasi berbasis Android untuk mempermudah akses informasi bagi masyarakat, memperkenalkan *sistem knowledge management* berbasis *web* untuk dokumentasi dan analisis data, serta merancang prototipe dasbor untuk pengelolaan data secara visual dan interaktif.

Sebagai contoh, penelitian awal memanfaatkan teknologi Arduino untuk pemantauan kualitas air, menghasilkan model sistem monitoring menggunakan sensor pH, suhu, dan kekeruhan, dengan pengiriman data ke perangkat sentral melalui XBee. Teknologi ini menunjukkan potensi IoT untuk mendukung analisis data *realtime* dengan efisiensi yang tinggi. Selain itu, pengembangan aplikasi berbasis Android memanfaatkan prinsip user-centered design untuk memastikan kemudahan akses bagi masyarakat umum dalam memonitor kualitas air melalui perangkat mobile. Penelitian lain yang fokus pada *knowledge management* berbasis *web* memanfaatkan pendekatan data historis untuk memprediksi perubahan kualitas air berdasarkan faktor lingkungan seperti musim dan curah hujan. Hasil penelitian ini mendukung pengembangan sistem yang tidak hanya bersifat operasional tetapi juga analitis, memungkinkan kolaborasi multi-pihak untuk keberlanjutan pengelolaan sumber daya air.

Pada saat penelitian ini dilakukan, sebagian besar infrastruktur IoT Bank Air Kami yang pernah digunakan pada penelitian sebelumnya tidak lagi beroperasi secara kontinu. Perangkat sensor tidak lagi mengirimkan data kualitas air secara *realtime*, dan beberapa komponen perangkat lunak lama sulit dijalankan kembali karena ketergantungan pustaka yang sudah usang. Kondisi ini menyebabkan pengumpulan data primer baru dalam bentuk data sensor maupun uji coba langsung kepada pengguna tidak dapat dilakukan secara optimal. Oleh karena itu, penelitian ini memanfaatkan data sekunder berupa log IoT dan data *dummy*, serta data simulasi yang direkayasa berdasarkan standar baku mutu dan literatur, sebagai dasar untuk merancang dan menguji sistem yang dikembangkan.

Oleh karena itu, sistem monitoring perlu ditingkatkan melalui pengembangan dasbor responsif yang dirancang mampu memvisualisasikan data kualitas air baik historis maupun *realtime* ketika infrastruktur IoT kembali aktif, serta memungkinkan integrasi fitur-fitur analitik. Dalam konteks penelitian ini, pengujian dasbor dilakukan dengan menggunakan data sekunder dan data simulasi, sehingga fokus utama berada pada keandalan rancangan sistem

dan kesesuaian fitur dengan proses bisnis usulan, bukan pada representasi kondisi kualitas air aktual di lapangan. Penelitian ini juga berfokus pada kemampuan skalabilitas dari sistem, yang memungkinkan dasbor untuk berkembang sesuai dengan peningkatan sensor maupun filter.

Di luar perannya sebagai penyedia air minum dalam kemasan dan layanan isi ulang, Bank Air Kami memiliki potensi untuk berkembang menjadi penyedia solusi pengelolaan kualitas air yang lebih komprehensif. Salah satu peluang pengembangan tersebut adalah dengan tidak hanya mendistribusikan air minum, tetapi juga memproduksi dan mengelola filter air yang digunakan oleh pelanggan.

Gagasan pengembangan ini belum sepenuhnya terimplementasi dalam operasional Bank Air Kami saat ini, namun dapat dipandang sebagai arah perluasan proses bisnis yang tetap sejalan dengan visi awal Bank Air Kami dalam menyediakan air minum yang layak konsumsi. Dalam konteks penelitian ini, peneliti mengusulkan model proses bisnis baru yang memperluas layanan Bank Air Kami ke aspek manajemen filter air, tanpa mengubah atau mengganggu proses bisnis yang telah berjalan, melainkan memperluas dan memperbaharui alur yang sudah ada. Model proses bisnis usulan tersebut kemudian diwujudkan dalam bentuk dasbor *web* responsif yang mendukung pemantauan kualitas air sekaligus pengelolaan filter air.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang dapat disimpulkan sebuah rumusan masalah yaitu “Bagaimana merancang dan mengembangkan dasbor *web* pemantauan kualitas air yang responsif, simulasi data sensor IoT, serta dapat diperluas untuk mendukung proses bisnis usulan Bank Air Kami dalam pengelolaan filter air, dengan memanfaatkan data sekunder dan data simulasi sebagai dasar perancangan dan pengujian sistem?”.

## 1.3 Batasan Masalah

- a. Penelitian ini difokuskan pada pengembangan dasbor *web* yang mendukung pemantauan parameter kualitas air serta modul manajemen filter air yang berfokus pada pengelolaan stok dan penggunaan filter pada pelanggan.
- b. Proses bisnis pengelolaan filter air yang dimodelkan dalam penelitian ini merupakan proses bisnis usulan yang disusun oleh peneliti berdasarkan potensi pengembangan Bank Air Kami. Proses bisnis ini tidak dimaksudkan untuk menggantikan proses bisnis yang saat ini berjalan, melainkan untuk memperluas dan memperbarui alur layanan yang sudah ada, dan penelitian ini tidak mengevaluasi tingkat adopsinya secara nyata di lapangan.

- c. Ruang lingkup geografis penelitian dibatasi pada konteks Bank Air Kami di Kampung Terban, sehingga hasil penelitian tidak secara langsung digeneralisasikan untuk wilayah lain dengan kondisi sosial dan geografis yang berbeda.
- d. Penelitian ini tidak melakukan pengumpulan data primer baru dari sensor IoT maupun dari pengguna. Data yang digunakan berupa data sekunder dari penelitian dan sistem Bank Air Kami sebelumnya (seperti log IoT dan data *dummy*) serta data simulasi yang direkayasa peneliti berdasarkan standar baku mutu dan literatur, sehingga tidak merepresentasikan kondisi kualitas air aktual pada saat penelitian dilakukan.
- e. Pengujian sistem dalam penelitian ini dibatasi pada pengujian fungsional dan non-fungsional yang dilakukan oleh peneliti, baik secara manual maupun melalui pengujian otomatis. Penelitian ini tidak mencakup pengujian penerimaan pengguna (*user acceptance test*) maupun survei kepuasan pengguna.
- f. Sistem yang dikembangkan tidak digunakan untuk menetapkan atau menyatakan secara resmi kelayakan air aman atau tidak aman untuk diminum. Dasbor berperan sebagai alat bantu visualisasi dan pengelolaan informasi kualitas air dan filter, sedangkan keputusan terkait kelayakan air tetap berada pada pihak atau kajian lain di luar lingkup penelitian ini.
- g. Spesifikasi filter dan sensor yang digunakan dalam penelitian ini menggunakan informasi yang bersifat umum, terutama terkait jenis filter, parameter atau sensor yang digunakan dan rentang ukur sensor.

#### **1.4 Tujuan Penelitian**

Tujuan utama dari penelitian ini adalah merancang dan mengembangkan sebuah dasbor *web* pemantauan kualitas air dan dapat diperluas untuk mendukung proses bisnis usulan Bank Air Kami dalam pengelolaan filter air. Secara lebih khusus, penelitian ini bertujuan menyusun model proses bisnis usulan untuk memperluas layanan Bank Air Kami hingga mencakup pengelolaan filter air. Selain itu, penelitian ini mengembangkan dasbor *web* responsif yang mengintegrasikan pemantauan kualitas air dengan modul manajemen filter untuk pengelolaan stok dan penggunaan oleh pelanggan. Terakhir, fungsi-fungsi utama dasbor diuji secara fungsional dan non-fungsional menggunakan data sekunder dan simulasi guna memastikan kesesuaian sistem dengan rancangan proses bisnis.

#### **1.5 Manfaat Penelitian**

Manfaat utama dari penelitian ini adalah tersusunnya sebuah dasbor *web* yang responsif dan dapat diperluas untuk mendukung pemantauan kualitas air serta pengelolaan filter air pada

Bank Air Kami berdasarkan proses bisnis usulan yang telah dirancang. Dasbor ini diharapkan dapat membantu pengelola dalam mengorganisasi dan memvisualisasikan informasi parameter kualitas air serta kondisi penggunaan filter secara terstruktur. Hal ini bertujuan untuk mendukung pengambilan keputusan yang lebih terdokumentasi dan berbasis data.

Dasbor ini dirancang untuk mampu menampilkan data kualitas air dengan mensimulasikan data IoT pada lingkungan pengujian, sehingga alur integrasi data, penyimpanan, dan visualisasi dapat diuji tanpa bergantung pada ketersediaan perangkat sensor secara real-time. Pendekatan simulasi ini juga memungkinkan peneliti menyiapkan skenario data yang merepresentasikan variasi kondisi kualitas air sesuai standar baku mutu dan literatur, sehingga fitur pemantauan dapat dievaluasi dari sisi fungsionalitas dan keterbacaan informasi.

## **1.6 Metodologi Penelitian**

Penelitian ini termasuk dalam kategori penelitian dan pengembangan atau *Research and Development* (R&D) dengan produk utama berupa dasbor berbasis *web* untuk pemantauan kualitas air dan pengelolaan filter air di Bank Air Kami berdasarkan usulan proses bisnis. Pada tahap pengembangan sistem, penelitian ini menggunakan model waterfall yang dimodifikasi. Meskipun tahapan utama tetap mengikuti urutan waterfall, dalam praktiknya model ini memungkinkan adanya umpan balik antar-tahapan ketika diperlukan penyesuaian rancangan atau kebutuhan.

### **1.6.1 Analisis Kebutuhan**

Pada tahap ini, dilakukan identifikasi kebutuhan fungsional dan non-fungsional sistem dengan mengkaji artefak penelitian sebelumnya, kondisi terkini, potensi pengembangan Bank Air Kami, serta literatur terkait. Hasil analisis digunakan untuk merumuskan proses bisnis usulan dan fitur utama dasbor.

### **1.6.2 Perencanaan**

Tahap perencanaan berfokus pada penentuan ruang lingkup pengembangan dalam bentuk *Minimum Viable Product* (MVP), penyusunan arsitektur sistem, perancangan basis data, dan rancangan antarmuka dasbor *web* yang responsif sesuai kebutuhan pemantauan kualitas air dan pengelolaan filter.

### **1.6.3 Implementasi**

Tahap implementasi merealisasikan rancangan ke dalam bentuk aplikasi dasbor *web*, meliputi pengembangan *backend* dan *frontend*, pengelolaan data pada basis data, serta integrasi dengan data sekunder dan data simulasi yang disiapkan untuk kebutuhan pengujian sistem.

#### **1.6.4 Pengujian Aplikasi**

Pada tahap ini dilakukan pengujian fungsional dan non-fungsional untuk memastikan sistem berjalan sesuai kebutuhan yang telah dirumuskan. Pengujian dilakukan oleh peneliti menggunakan data sekunder dan data simulasi, tanpa melibatkan pengujian penerimaan pengguna secara langsung.

#### **1.6.5 Evaluasi dan Dokumentasi**

Tahap evaluasi dan dokumentasi bertujuan menilai kesesuaian antara rancangan dan implementasi sistem, mengidentifikasi keterbatasan, serta menyusun dokumentasi terkait rancangan, implementasi, dan hasil pengujian sebagai dasar penyusunan kesimpulan dan saran pengembangan lebih lanjut.

### **1.7 Sistematika Penulisan**

Penulisan laporan tugas akhir ini disusun ke dalam lima bab yang saling berkaitan. Bab I Pendahuluan menyajikan latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, serta metodologi penelitian secara umum untuk memberikan gambaran konteks, urgensi, dan arah penelitian. Bab II Tinjauan Pustaka memuat kajian teoretis dan penelitian terdahulu yang relevan, mencakup konsep kualitas air, teknologi IoT, dasbor *web*, manajemen filter air, serta penelitian terkait Bank Air Kami sebagai landasan teoretis untuk mengidentifikasi kesenjangan penelitian.

Bab III Metodologi menjelaskan pendekatan penelitian pengembangan (*Research and Development*), model pengembangan sistem, jenis dan sumber data, analisis kebutuhan, perancangan sistem, serta tahapan implementasi dan pengujian. Bab IV Hasil dan Pembahasan menyajikan hasil implementasi dasbor *web* beserta hasil pengujiannya. Bab ini juga membahas keterkaitan sistem dengan proses bisnis usulan Bank Air Kami, termasuk analisis kelebihan, keterbatasan, dan kontribusi penelitian. Bab V Kesimpulan dan Saran memuat rangkuman temuan utama untuk menjawab rumusan masalah, serta saran untuk pengembangan sistem dan penelitian lanjutan di masa mendatang.

## **BAB II**

### **KAJIAN PUSTAKA**

#### **2.1 Air Bersih Layak Konsumsi**

Ketersediaan air bersih layak konsumsi menjadi isu yang terus menguat karena perubahan iklim, pertumbuhan penduduk, dan urbanisasi dapat memengaruhi debit, distribusi, serta mutu air baku. Dalam berbagai laporan, risiko krisis air diproyeksikan meningkat apabila pengelolaan sumber daya air tidak dibarengi intervensi yang memadai, baik pada sisi konservasi maupun pengendalian kualitas.

Berdasarkan laporan *World Meteorological Organization* (WMO), lebih dari dua miliar orang hidup dalam kondisi kekurangan air, sedangkan lebih dari lima miliar orang diprediksi akan mengalami krisis air pada tahun 2050 jika tidak ada intervensi signifikan dalam pengelolaan sumber daya air (World Meteorological Organization, 2021). Tantangan ini tidak hanya berdampak pada kesehatan masyarakat, tetapi juga pada ketahanan pangan, ekonomi, dan keamanan global. Oleh karena itu, menjaga kualitas dan ketersediaan air layak konsumsi menjadi prioritas utama berbagai negara di dunia.

Pada level lokal, tantangan tersebut relevan dengan kondisi Kampung Terban, Yogyakarta, yang berada di sekitar Sungai Code dan memanfaatkan mata air sebagai sumber utama kebutuhan harian. Meski menjadi sumber vital, mutu air pada konteks seperti ini cenderung dinamis dan dipengaruhi cuaca, kondisi lingkungan, serta aktivitas manusia di sekitar wilayah tangkapan. Karena itu, upaya menjaga kelayakan air perlu mengacu pada parameter kualitas yang terukur dan dapat dipantau secara berkelanjutan, sehingga perubahan kondisi bisa terdeteksi lebih dini dan ditindaklanjuti secara operasional.

#### **2.2 Standar Kualitas Air**

Kualitas air minum umumnya dievaluasi melalui kelompok parameter mikrobiologi, fisik, kimia, dan radiologi. Dari sisi kesehatan masyarakat, parameter mikrobiologi menjadi indikator kritis karena berkaitan langsung dengan risiko patogen. Di sisi lain, parameter fisik-kimia (misalnya kekeruhan dan pH) sering digunakan sebagai indikator praktis untuk menilai perubahan mutu air yang cepat teramati dan relevan untuk pengendalian proses.

Berdasarkan standar air minum di berbagai negara dan *World Health Organization* (WHO), air minum harus bebas dari kontaminan mikrobiologis, seperti *Escherichia coli* dan bakteri patogen lainnya (European Economic Community, 1980; Levin et al., 2024;

‘Permenkes-No-492-Tahun-2010-tentang-Persyaratan-Kualitas-Air-Minum’, 2010.; United States Environmental Protection Agency, 2024; World Health Organization, 2011). Selain itu, parameter fisik seperti kekeruhan (turbiditas) dan rasa juga menjadi indikator penting. WHO menetapkan ambang batas kimia untuk zat seperti arsenik, klorin, dan fluorida yang jika jumlahnya berlebih dapat menyebabkan masalah kesehatan serius (World Health Organization, 2011).

Di Amerika Serikat, USEPA menerapkan standar melalui *National Primary Drinking Water Regulations* (NPDWRs) yang menetapkan *Maximum Contaminant Levels* (MCLs) untuk berbagai bahan kimia berbahaya, seperti timbal, arsenik, nitrat, dan merkuri. USEPA juga mengatur tindakan pencegahan berbasis teknik pengolahan jika tingkat pencemaran kimia tertentu sulit diukur langsung.

Sementara itu, *Bureau of Indian Standards* (BIS), melalui standar IS 10500:2012, mengatur bahwa air minum harus memenuhi ambang batas kontaminan mikrobiologis, kimia, serta parameter fisik seperti kekeruhan. Di India, perhatian besar diberikan pada kontaminasi bahan kimia seperti fluorida dan pestisida yang kerap menjadi masalah di wilayah pedesaan.

Di Eropa, *Directive 98/83/EC* menetapkan standar air minum yang ketat untuk mikroorganisme patogen, logam berat, dan pestisida. Adapun di Indonesia, pemerintah melalui Permenkes No. 2 Tahun 2023 mengikuti pedoman WHO dan menyelaraskan beberapa parameter internasional. Namun, standar ini masih terbatas dalam pengaturan parameter kimia, khususnya pestisida.

Berikut ini adalah tabel perbandingan standar kualitas air bersih berdasarkan beberapa parameter yang diterapkan oleh WHO, USEPA, BIS, Eropa, dan Indonesia berdasarkan parameter utama:

Tabel 2.1 Perbandingan Standar Air Minum

Parameter	WHO	USEPA	BIS(IS 10500:2012)	Eropa (Directive 98/83/EC)	Indonesia (Permenkes 2023)
E. coli	0 mg/L	0 mg/L	0 mg/L	0 mg/L	0 mg/L
Coliforms	0 mg/L	0 mg/L	0 mg/L	0 mg/L	0 mg/L
Kekeruhan (Turbidity)	Maksimum 5 NTU	Maksimum 1 NTU	Maksimum 5 NTU	Maksimum 1 NTU	Maksimum 5 NTU
Arsenik	0.01 mg/L	0.01 mg/L	0.01 mg/L	0.01 mg/L	0.01 mg/L
Nitrat	50 mg/L	10 mg/L	45 mg/L	50 mg/L	50 mg/L
Fluorida	1.5 mg/L	4.0 mg/L	1.0 mg/L	1.5 mg/L	1.5 mg/L
Timbal	0.1 mg/L	0.15 mg/L	0.01 mg/L	0.1 mg/L	0.1 mg/L
besi	0.3 mg/L	0.3 mg/L	0.3 mg/L	0.2 mg/L	0.3 mg/L

Trihalometana Total	0.1 mg/L	0.08 mg/L	Tidak diatur	0.1 mg/L	0.1 mg/L
PH	6.5 - 8.5	6.5 - 8.5	6.5 - 8.5	6.5 - 9.5	6.5 - 8.5

Tabel di atas menggambarkan bahwa standar kualitas air minum di seluruh dunia memiliki banyak kesamaan, terutama dalam parameter-parameter mikrobiologis dan kimia utama seperti E. coli, arsenik, dan coliforms. Perbedaan terjadi seperti pada batas maksimal fluorida dan nitrat, yang mungkin mencerminkan kondisi lingkungan yang berbeda di masing-masing negara atau wilayah.

USEPA dan Eropa memiliki standar yang lebih ketat dan komprehensif, mencakup pengaturan lebih banyak parameter, termasuk pestisida dan senyawa organik volatil (VOCs), yang tidak diatur dalam standar Indonesia dan BIS (India). Indonesia, meskipun telah selaras dengan standar internasional untuk parameter utama seperti E. coli, arsenik, dan nitrat, masih bisa memperluas standar dengan menambahkan batasan untuk senyawa-senyawa lain guna meningkatkan perlindungan kesehatan. Dibandingkan dengan standar-standar lainnya, Indonesia sudah memadai untuk memenuhi kebutuhan dasar, tetapi peningkatan standar untuk mencakup kontaminan kimia yang lebih luas akan memberikan jaminan kualitas air yang lebih baik. Standar ini tidak hanya menjadi acuan teknis, tetapi juga memengaruhi strategi tata kelola air minum di berbagai kawasan.

### 2.3 Tata Kelola Infrastruktur

Di Amerika Serikat, tata kelola air bersih dikendalikan oleh *Environmental Protection Agency* (EPA) melalui regulasi *Safe Drinking Water Act* (SDWA). Pendekatan ini berfokus pada pengelolaan kualitas air dengan penerapan teknologi canggih seperti *Water Quality Surveillance and Response Systems* (SRS). Sistem ini dirancang untuk mendeteksi anomali kualitas air secara dini dengan menggabungkan data *realtime* dari sensor kualitas air yang terpasang di berbagai titik jaringan distribusi (U.S. Environmental Protection Agency, 2015).

Integrasi *Geographic Information System* (GIS) menjadi komponen penting dalam tata kelola ini, memungkinkan visualisasi dan analisis spasial data kualitas air. Selain itu, *cloud computing* digunakan untuk memproses data dalam skala besar, memungkinkan akses yang cepat dan efisien bagi para pemangku kepentingan. Model hidrolis seperti *Contamination Dissemination Look-Ahead Simulation* (CDLAS) menjadi alat utama dalam mensimulasikan penyebaran kontaminasi, sehingga langkah mitigasi dapat dilakukan dengan cepat dan tepat (U.S. Environmental Protection Agency, 2015).

Di Eropa, tata kelola air minum dikelola melalui *Directive* (EU) 2020/2184 yang menekankan pendekatan manajemen risiko di seluruh rantai pasokan air. Teknologi *Continuous Water-Quality Monitoring* (CWQM) diterapkan untuk memantau parameter seperti pH, suhu, dan kekeruhan secara *realtime*. GIS memainkan peran penting dalam memetakan jaringan distribusi air, yang memungkinkan pengelolaan lebih efisien dan responsif terhadap potensi masalah (Carmi & Theocharidou, 2019).

Eropa juga memanfaatkan teknologi optimasi penempatan sensor untuk meningkatkan efektivitas sistem pemantauan. Dengan perangkat lunak khusus, mereka dapat mengidentifikasi titik-titik kritis dalam jaringan distribusi. Pendekatan ini mendukung pengambilan keputusan berbasis data yang lebih baik, memastikan kualitas air tetap terjaga bahkan dalam kondisi darurat (Carmi & Theocharidou, 2019).

Penelitian seperti yang diulas oleh (Wen et al., 2020) menyoroti pentingnya indikator mikrobiologis dalam pengelolaan air, dengan menyarankan peningkatan sistem pemantauan yang mencakup berbagai indikator mikrobiologis untuk memastikan deteksi dini kontaminasi patogen.

Bank Air Kami di Indonesia mengadopsi pendekatan yang memadukan teknologi lokal dan berbasis komunitas untuk memastikan ketersediaan air bersih bagi masyarakat sekitar. Sistem pemantauan kualitas air berbasis IoT diterapkan untuk mengukur parameter seperti pH, suhu, turbiditas, dan konduktivitas. Data yang dikumpulkan oleh sensor dikirim ke Firebase Realtime Database untuk pemantauan *realtime* melalui aplikasi seluler (Pratama & Sujarwo Ari, 2020).

Namun, terdapat beberapa kelemahan dalam infrastruktur Bank Air Kami dibandingkan dengan sistem di Amerika Serikat dan Eropa. Kelemahan ini meliputi keterbatasan dalam skalabilitas infrastruktur, kurangnya integrasi GIS yang mendalam, dan kendala dalam stabilitas koneksi internet, yang dapat memengaruhi keandalan pengiriman data (Atalaric & Sujarwo, 2024).

Pendekatan Amerika Serikat dan Eropa menunjukkan perbedaan yang menarik. Amerika Serikat lebih berfokus pada respons cepat melalui teknologi seperti SRS dan CDLAS, sementara Eropa mengedepankan pencegahan dengan pendekatan manajemen risiko yang terintegrasi. Bank Air Kami, meskipun telah mengadopsi teknologi IoT, masih memiliki ruang untuk perbaikan, terutama dalam pengelolaan data mikrobiologis dan integrasi GIS yang lebih luas.

Kombinasi dari pendekatan ini dapat menciptakan sistem yang lebih holistik, dengan respons yang cepat sekaligus pencegahan yang efektif. Misalnya, penggunaan simulasi

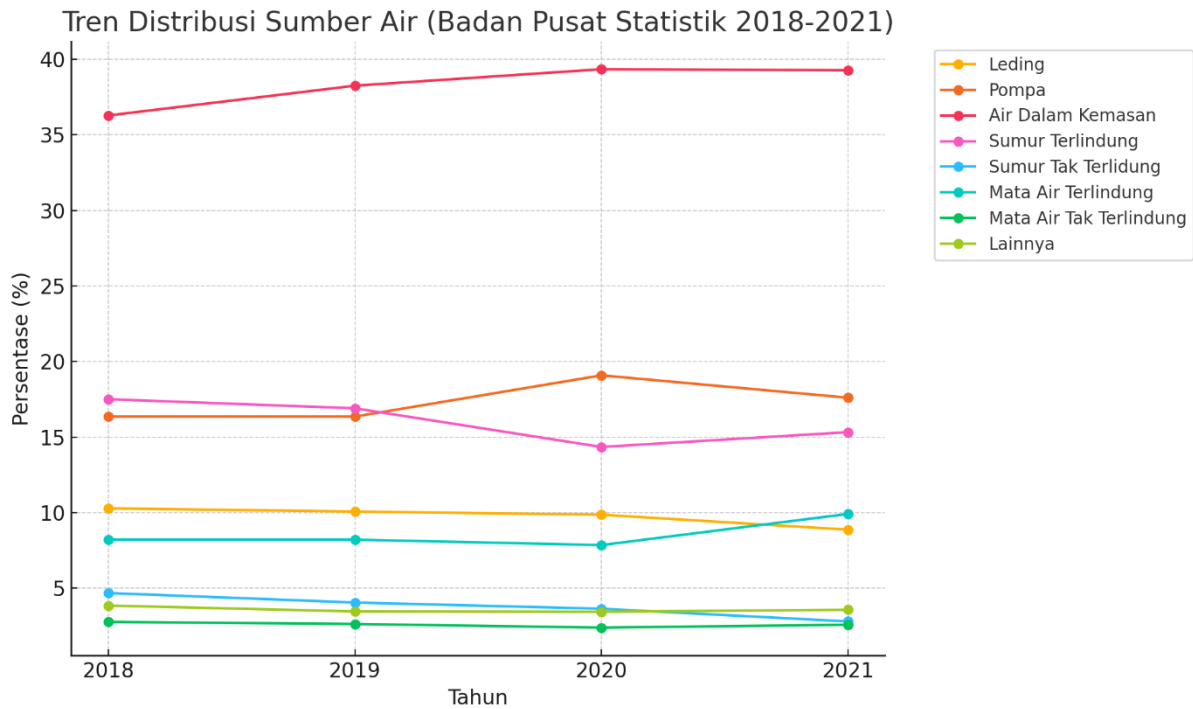
CDLAS dari Amerika Serikat dapat dikombinasikan dengan optimasi penempatan sensor ala Eropa dan pengembangan aplikasi lokal seperti yang diterapkan di Bank Air Kami untuk menciptakan jaringan distribusi air yang lebih tahan terhadap insiden kontaminasi.

Tantangan besar dalam tata kelola infrastruktur air minum adalah skalabilitas. Baik di Amerika Serikat maupun Eropa, penerapan teknologi IoT harus mampu menangani pertumbuhan data yang signifikan dari berbagai perangkat. Teknologi seperti edge computing dan arsitektur IoT modular memberikan solusi untuk mengatasi tantangan ini. Sistem ini tidak hanya meningkatkan kapasitas pengolahan data, tetapi juga memungkinkan integrasi perangkat yang lebih efisien (Javed, Malhi, Kinnunen, & Framling, 2020)

(Li, Yen, Lin, Tung, & Huang, 2018) juga menunjukkan bagaimana integrasi platform seperti Firebase dapat memungkinkan pengelolaan data *realtime*, termasuk pemantauan menggunakan perangkat dengan sumber daya terbatas seperti Arduino. Sistem ini mendukung keamanan data dengan protokol enkripsi dan memungkinkan analitik berbasis *machine learning* untuk pengambilan keputusan yang lebih cerdas.

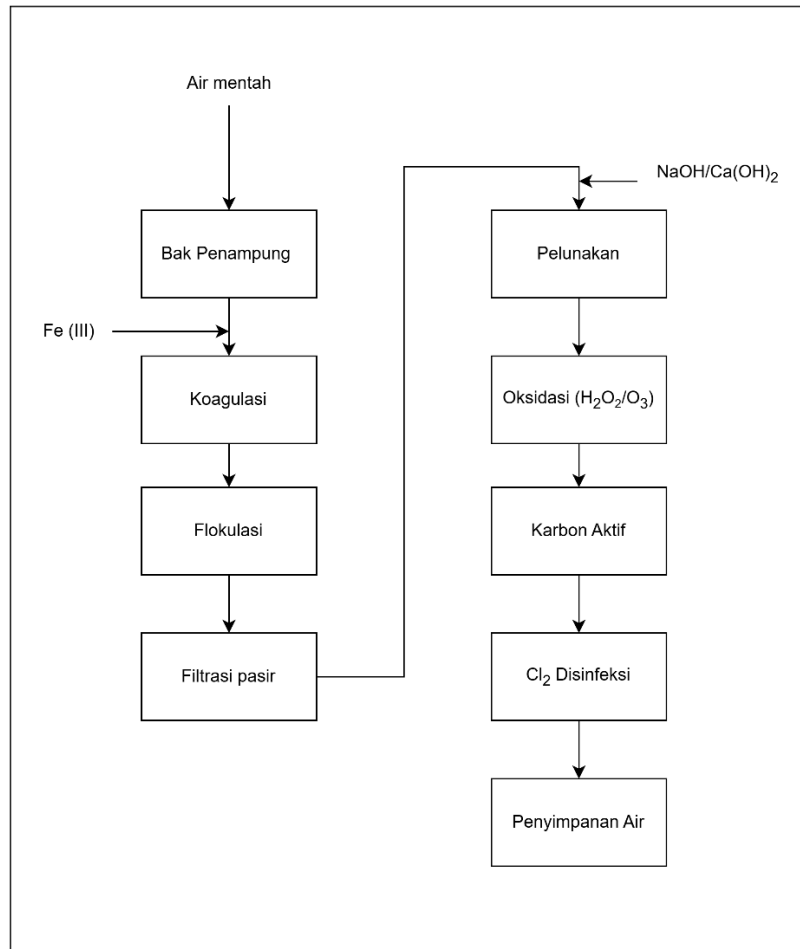
Dalam menangani insiden, baik Amerika Serikat maupun Eropa memanfaatkan teknologi tinggi. Amerika Serikat menggunakan *Event Management System* (EMS) untuk merespons insiden dengan cepat, sementara Eropa mengandalkan simulasi berbasis GIS untuk memprediksi dan memitigasi dampak insiden. Bank Air Kami, meskipun memiliki sistem pemantauan *realtime*, masih menghadapi tantangan dalam merespons insiden secara cepat karena keterbatasan infrastruktur komunikasi dan integrasi (Böhlen & Maharika, 2018). Berlandaskan kerangka kebijakan serta praktik tata kelola yang telah dijabarkan, perlu dicermati bagaimana ketentuan tersebut diterapkan dalam proses pengolahan air minum pada tingkatan industri maupun tradisional.

## 2.4 Proses Pengelolaan AMDK



Gambar 2.1 Tren Distribusi Sumber Air (BPS 2018-2021)

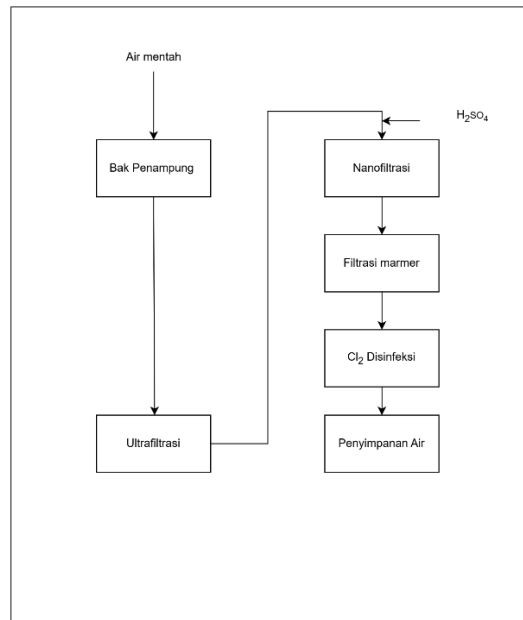
Data Badan Pusat Statistik 2018 - 2021 menunjukkan bahwa kebutuhan air minum meningkat, diiringi pergeseran signifikan ke air minum dalam kemasan (AMDK). Berdasarkan data tersebut, preferensi masyarakat dalam memenuhi kebutuhan air minum kini lebih condong kepada air minum dalam kemasan yang dianggap praktis dan higienis, sementara pemanfaatan sumber tradisional seperti air pompa, leding, serta sumur dan mata air cenderung menurun. Perubahan ini menyebabkan terjadinya peningkatan permintaan terhadap AMDK sehingga menimbulkan dua tuntutan sekaligus: kapasitas produksi yang lebih besar dan mutu produk yang konsisten sesuai standar keselamatan pangan. Sebelum membahas pengelolaan AMDK secara modern, penting untuk meninjau terlebih dahulu rantai proses konvensional yang masih digunakan di banyak pabrik air minum di Indonesia termasuk Bank Air Kami.



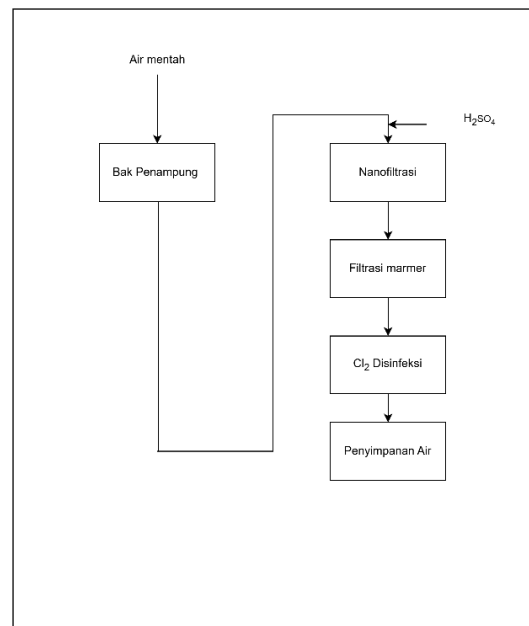
Gambar 2.2 Proses pengelolaan air minum konvensional

Skema pada Gambar 2.2 Proses pengelolaan air minum konvensional menyajikan urutan proses pengolahan air minum secara konvensional yang terdiri atas sembilan unit operasi utama. Pertama, air baku dialirkan ke bak penampung guna menstabilkan debit dan homogenitas mutu awal. Selanjutnya, air dipompa ke unit koagulasi, tempat koagulan besi (III) ditambahkan untuk menetralkan muatan koloid sehingga terbentuk mikrofluk. Mikrofluk tersebut diperbesar di unit flokulasi melalui pengadukan lambat, kemudian disisihkan pada filtrasi pasir cepat sehingga kekeruhan menurun secara signifikan. Filtrat yang dihasilkan memasuki tahap pelunakan; penambahan NaOH atau Ca(OH)<sub>2</sub> menaikkan pH, memicu pengendapan ion kalsium dan magnesium sebagai karbonat atau hidroksida, dan menurunkan kesadahan. Air yang telah dilunakkan selanjutnya diproses pada reaktor oksidasi menggunakan hidrogen peroksida atau ozon untuk mengoksidasi senyawa organik tersisa, logam terlarut, serta mematikan sebagian mikroorganisme. Produk oksidasi dialirkan melalui kolom karbon aktif granular guna mengadsorpsi senyawa penyebab rasa dan bau serta prekursor pembentuk produk samping desinfeksi. Setelah zat organik di dalam air berkurang, air diberi klorin pada

tahap akhir. Penambahan klorin ini bertujuan untuk membunuh kuman yang masih tersisa dan meninggalkan sedikit kadar klorin supaya air tetap aman hingga sampai ke pengguna (Darise, 2016; Peraturan Menteri Perindustrian R.I, 2011; Van Der Bruggen, Verberk, & Verhack, 2004).



Gambar 2.3 Proses Pengelolaan Air minum Modern Dengan Ultrafiltrasi



Gambar 2.4 Proses Pengelolaan Air Minum Modern Tanpa Ultrafiltrasi

Skema pada Gambar 2.3 Proses Pengelolaan Air minum Modern Dengan Ultrafiltrasi menyajikan urutan proses pengolahan air minum secara konvensional yang terdiri dari 6 unit operasi utama. Air baku terlebih dahulu masuk ke bak penampung untuk menstabilkan debit dan mutu awal. Dari unit ini, air dipompa ke sistem ultrafiltrasi (UF); membran UF berpori sangat halus sehingga mampu menahan kekeruhan, koloid, bakteri, dan sebagian besar virus, menghasilkan air yang jauh lebih jernih. Tahapan ini sering kali di lewatkan untuk menekan biaya produksi seperti pada Gambar 2.4 Proses Pengelolaan Air Minum Modern Tanpa Ultrafiltrasi. Filtrat UF selanjutnya dialirkan ke unit nanofiltrasi (NF). Sebelum memasuki membran NF, larutan akan ditambahkan sedikit asam sulfat ( $H_2SO_4$ ) yang berguna untuk menurunkan pH dan mencegah pengendapan kalsium atau magnesium. Membran NF menyisihkan ion-ion hardness, senyawa organik berukuran kecil, serta sisa mikroorganisme, tetapi tetap mempertahankan sebagian mineral ringan sehingga rasa air tetap “segar” (Van Der Bruggen et al., 2004; Zhang et al., 2021).

Proses selanjutnya air akan melewati filtrasi marmer (remineralisasi). Media marmer terlarut secara perlahan untuk membahkan kalsium serta menetralkan pH, sehingga air menjadi stabil dan tidak korosif bagi pipa maupun kemasan. Setelah komposisi mineral dikoreksi, air diberi dosis klorin pada tahap disinfeksi akhir untuk memastikan tidak ada kuman tersisa dan menjaga kadar klorin residual selama distribusi.

Berdasarkan literatur, proses pengelolaan air minum konvensional sudah sesuai dengan peraturan Menteri Perindustrian Nomor 96/m-ind/per/12/2011 Tahun 2011 Tentang Persyaratan Teknis Industri Air Minum dalam Kemasan. Namun, kualitas produk sangat dipengaruhi fluktuasi mutu air baku serta bergantung pada dosis bahan kimia yang relatif tinggi. Sebaliknya, skema modern berbasis membran (UF–NF) menawarkan konsistensi mutu lebih baik karena membran ultrafiltrasi menahan padatan tersuspensi, bakteri, dan sebagian virus, sedangkan nanofiltrasi menyisihkan ion kesadahan dan jejak pestisida tanpa menghilangkan seluruh mineral yang berkontribusi pada rasa segar.

Tabel 2.2 Perbandingan Filter air tradisional dan modern

Aspek	Proses Tradisional	Proses Modern (UF–NF)
Mutu produk	Sensitif terhadap fluktuasi air baku	Lebih stabil
Limbah	Sludge koagulasi	Hanya concentrate dan tanpa sludge kimia
Konsumsi energi	Rendah (pompa tekanan kecil)	Sedang–tinggi
Biaya pembangunan	Lebih murah	mahal
Perawatan		Pencucian (CIP) berkala dan modul diganti 5–7 tahun sekali.

	Cuci filter pasir, ganti karbon aktif cukup sering.	
Risiko re-kontaminasi	Relatif tinggi pada media terbuka	Rendah Sekitar 20–50 % lebih mahal
Skalabilitas	Penambahan kapasitas memerlukan bangunan baru	kapasitas mudah ditambah

Dari Tabel 2.2 Perbandingan Filter air tradisional dan modern menunjukkan bahwa proses tradisional unggul dalam investasi awal dan kebutuhan energi yang lebih sedikit. Dibandingkan dengan proses modern yang memiliki konsistensi mutu yang lebih baik, *scalable*, dan menghasilkan limbah yang lebih sedikit. Namun, keputusan memilih teknologi tidak dapat semata-mata berdasarkan satu atau dua faktor. Bagi perusahaan yang beroperasi di daerah dengan pasokan listrik terbatas dan modal awal terbatas, proses tradisional mungkin tetap menjadi pilihan paling realistis. Sebaliknya, jika sasaran pasar adalah konsumen yang menuntut mutu sangat stabil, ruang lahan pabrik terbatas, atau jika regulasi lingkungan kian ketat terhadap penggunaan bahan kimia dan pembuangan lumpur, maka investasi pada sistem membran modern menjadi lebih masuk akal dalam jangka panjang. Selain itu, skema UF–NF yang bersifat modular memudahkan penambahan kapasitas seiring pertumbuhan permintaan AMDK tanpa perlu perluasan bangunan besar.

Berdasarkan uraian proses konvensional dan modern, dapat disimpulkan bahwa unit filtrasi (media pasir, karbon aktif, dan membran) merupakan komponen yang menentukan kestabilan mutu air. Kinerja filter bersifat menurun seiring waktu pemakaian (*fouling*/saturasi), sehingga diperlukan pengelolaan yang mencakup pencatatan jenis filter, status pemakaian, jadwal perawatan/penggantian, serta keterkaitan dengan indikator kualitas air.

## 2.5 Filter Air

Istilah filter umumnya merujuk pada unit atau media penyisih yang bekerja melalui mekanisme tertentu untuk menurunkan kandungan kontaminan pada air. Secara konseptual, filtrasi dapat dipahami sebagai proses pemisahan yang memanfaatkan hambatan fisik (misalnya penyaringan partikel berdasarkan ukuran pori), interaksi kimia (misalnya adsorpsi pada media tertentu), maupun pemisahan berbasis membran yang memiliki karakteristik pemisahan lebih ketat. Dengan demikian, “filter” dalam pembahasan ini dipahami sebagai komponen teknis yang memiliki fungsi spesifik sesuai mekanisme kerjanya (Ireland’s Environmental Protection Agency, 2020; U.S. Environmental Protection Agency, 2005).

Pada praktik penggunaan unit pengolahan air skala rumah tangga atau point-of-use (POU), filter tidak selalu berdiri sendiri. Banyak sistem POU disusun sebagai rangkaian beberapa tahap untuk mencapai tujuan kualitas air tertentu, misalnya tahap awal untuk mengurangi sedimen/partikel, tahap berikutnya untuk mengurangi kontaminan terlarut tertentu, dan pada beberapa kasus dilengkapi tahap lanjutan seperti membran atau disinfeksi. Pola bertahap ini menunjukkan bahwa efektivitas filter sangat bergantung pada kombinasi teknologi, serta tujuan pengolahan dan karakteristik air baku (Wu, Cao, Tong, Finkelstein, & Hoek, 2021).

Pemilihan filter juga berkaitan dengan prinsip bahwa tidak ada satu jenis filter yang efektif untuk semua kontaminan. Menurut CDC (*Centers for Disease Control and Prevention*) menekankan bahwa pemilihan filter perlu mempertimbangkan kontaminan target, karena teknologi berbeda memiliki kemampuan reduksi yang berbeda. Dengan kata lain, filter perlu dipahami sebagai unit yang “sesuai tujuan”, bukan solusi generik untuk seluruh permasalahan kualitas air (Centers for Disease Control and Prevention, 2024).

Selain aspek fisik-kimia, pengolahan air minum juga memiliki dimensi mikrobiologis. Pedoman evaluasi opsi pengolahan air tingkat rumah tangga menekankan pentingnya melihat kinerja opsi pengolahan terhadap pengurangan risiko kesehatan, terutama ketika tujuan pengolahan dikaitkan dengan pengendalian patogen. Hal ini menguatkan bahwa “efektivitas filter” tidak hanya diartikan sebagai kejernihan atau perubahan parameter fisik, tetapi juga dapat dikaitkan dengan kemampuan memenuhi target kinerja yang relevan terhadap keselamatan (World Health Organization, 2011).

Secara umum, filter air dapat diklasifikasikan berdasarkan mekanisme penyisihan kontaminan dan tujuan pengolahan. Klasifikasi berbasis mekanisme dipandang lebih mudah diterapkan pada konteks pengolahan air minum karena setiap mekanisme memiliki karakteristik, batas kemampuan, dan kebutuhan operasional yang berbeda (Ireland’s Environmental Protection Agency, 2020; U.S. Environmental Protection Agency, 2005). Pada sistem POU, literatur juga menunjukkan bahwa beberapa mekanisme sering dikombinasikan dalam satu rangkaian bertahap untuk mencapai mutu air (Wu et al., 2021). Sejalan dengan hal tersebut, klasifikasi filter pada subbab ini disusun ke dalam beberapa kelompok utama sebagai berikut.

### **2.5.1 Filter Fisik**

Kelompok ini bekerja terutama melalui mekanisme penyaringan fisik, sehingga efektif untuk menurunkan kontaminan berupa partikel tersuspensi yang mempengaruhi

kekeruhan/kejernihan air. Filtrasi media (misalnya filtrasi pasir) banyak dibahas pada konteks instalasi pengolahan air minum, sementara sediment prefilter umum ditemui pada unit POU sebagai tahap awal (Ireland's Environmental Protection Agency, 2020; Wu et al., 2021).

### 2.5.2 Filter Adsorpsi

Kelompok ini mengandalkan proses adsorpsi pada permukaan media untuk menurunkan senyawa tertentu, termasuk komponen yang memengaruhi rasa dan bau serta sebagian senyawa organik terlarut.

### 2.5.3 Filter Pertukaran Ion / Demineralization

Kelompok ini memanfaatkan reaksi pertukaran ion untuk menurunkan ion terlarut tertentu (ion atau mineral seperti garam, dan magnesium). Mekanisme ini berbeda dari filtrasi fisik dan adsorpsi karena target utamanya adalah komponen terlarut spesifik yang dapat ditukar pada media resin (U.S. Environmental Protection Agency, 2005; Wu et al., 2021).

### 2.5.4 Filter Membran (MF/UF/NF/RO)

Kelompok ini menggunakan pemisahan berbasis membran dan dapat diklasifikasikan menjadi *Microfiltration* (MF), *Ultrafiltration* (UF), *Nanofiltration* (NF), dan *Reverse osmosis* (RO). Perbedaannya terutama terkait ukuran pori/kemampuan menahan partikel hingga kontaminan terlarut serta kebutuhan operasi dan kontrol kinerja yang lebih ketat dibanding filtrasi media. Pada sistem POU, RO sering dipadukan dengan tahapan lain (prefilter dan/atau karbon) serta tahap pascaperlakuan (Wu et al., 2021).

### 2.5.5 Filter Pelengkap (Disinfeksi UV)

Sebagian sistem POU melengkapi rangkaian filtrasi dengan disinfeksi, misalnya ultraviolet (UV). Tahap ini ditujukan untuk pengendalian mikrobiologis melalui inaktivasi patogen, sehingga diperlakukan sebagai tahap pelengkap yang melengkapi rangkaian filter (World Health Organization, 2011; Wu et al., 2021).

Table 2.1 Perbandingan Klasifikasi dan Perbedaan Antar Filter Air

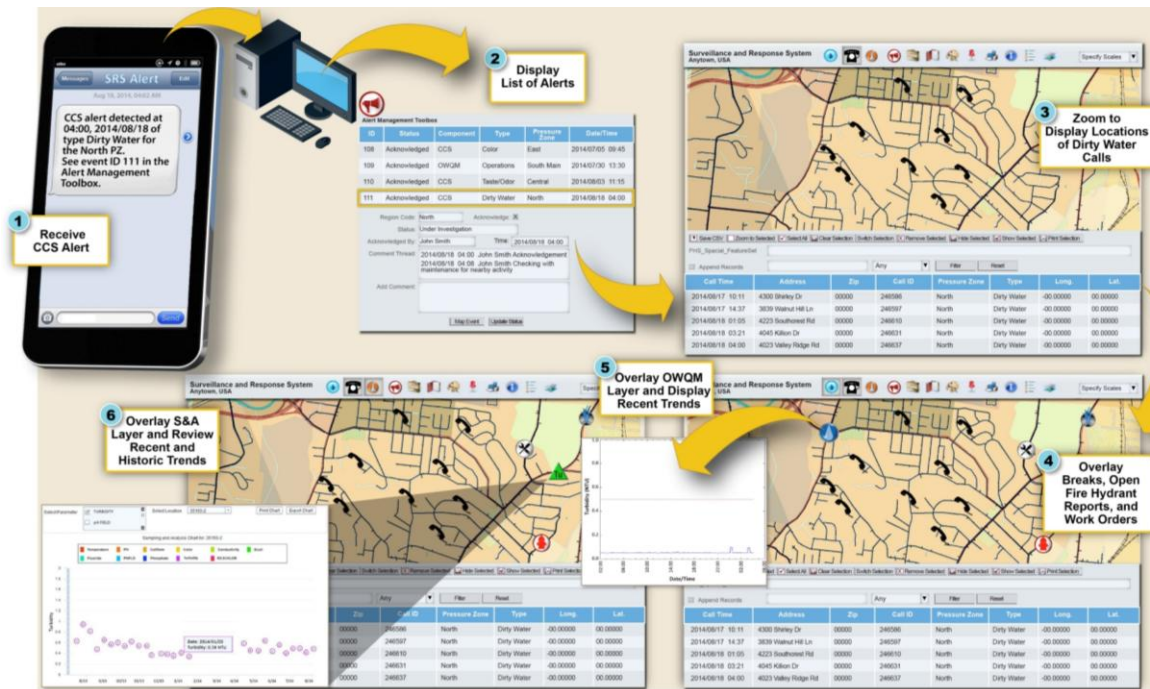
Kelompok filter	Mekanisme	Tujuan
Filter fisik	Penyaringan fisik / penahanan partikel (bergantung ukuran pori/media & kondisi operasi)	Partikel tersuspensi, kekeruhan

Filter adsorpsi	Adsorpsi pada permukaan media	Umumnya senyawa yang memengaruhi rasa/bau
Filter Pertukaran Ion / Demineralization	Pertukaran ion	Menghilangkan mineral
Filter membran (MF/UF/NF/RO)	Pemisahan berbasis membran	Penyisihan partikel halus hingga kontaminan terlarut
Tahap pelengkap / disinfeksi (UV)	Inaktivasi mikroorganisme	Menghilangkan mikrobiologis/patogen

## 2.6 Teknologi Pemantauan & Dasbor

Dasbor pemantauan kualitas air minum menjadi salah satu alat utama dalam memastikan kualitas air yang aman dan layak dikonsumsi. Berbagai pendekatan telah diterapkan di berbagai belahan dunia, termasuk Amerika Serikat dan Eropa, yang menunjukkan perbedaan dalam bentuk dasbor, cara kerja, serta pemanfaatannya.

Di Amerika Serikat, dasbor seperti SRS dasbor mengintegrasikan data *realtime* dari sensor kualitas air yang tersebar di jaringan distribusi. Sistem ini didukung oleh algoritma deteksi anomali dan visualisasi berbasis peta interaktif melalui GIS. Visualisasi ini memungkinkan identifikasi cepat terhadap potensi masalah seperti kebocoran atau kontaminasi. Selain itu, penggunaan *cloud computing* memperkuat kemampuan pemrosesan data dalam jumlah besar, memberikan hasil analisis yang mendalam untuk mendukung mitigasi insiden kualitas air (U.S. Environmental Protection Agency, 2015). Di Eropa, dasbor kualitas air berfokus pada deteksi dini dengan pemanfaatan CWQM. Sistem ini menekankan pengawasan berkelanjutan menggunakan sensor cerdas dan analitik prediktif. GIS memainkan peran penting dalam memetakan distribusi parameter kualitas air, yang memberikan visibilitas tinggi terhadap jaringan distribusi dan mendukung pengambilan keputusan berbasis (Carmi & Theocharidou, 2019; U.S. Environmental Protection Agency, 2015).



Gambar 2.5 Penggunaan GIS untuk Tata Kelola Air

Sumber: U.S. EPA (2015)

Selain GIS, jenis visualisasi yang digunakan dalam dasbor juga memainkan peran penting. Visualisasi seperti Time-Series Plots sangat membantu dalam memantau tren parameter kualitas air dari waktu ke waktu (U.S. Environmental Protection Agency, 2015). Gauge Displays digunakan untuk menampilkan kondisi terkini parameter dengan jelas dan ringkas, memberikan indikasi cepat apakah nilai berada dalam ambang batas yang diizinkan (U.S. Environmental Protection Agency, 2015). Box-and-Whisker Plots berguna untuk menggambarkan variasi data statistik, membantu analisis distribusi nilai dan deteksi outlier (U.S. Environmental Protection Agency, 2015). Ada juga penelitian yang menggunakan metode eDNA (environmental DNA) mulai diterapkan untuk memantau perubahan komunitas mikroorganisme dalam air minum secara cepat dan akurat, memberikan informasi penting terkait risiko mikrobiologis (Watson et al., 2024).

Meski telah menunjukkan efektivitasnya, penerapan dasbor masih menghadapi sejumlah tantangan. Di Amerika, pengelolaan data dari perangkat IoT dalam skala besar memerlukan infrastruktur yang andal dan kapasitas pemrosesan data yang tinggi. Sementara itu, di Eropa, biaya implementasi jaringan sensor cerdas menjadi kendala utama. Peluang perbaikan dapat melibatkan integrasi teknologi seperti edge computing untuk mengurangi beban jaringan dan peningkatan analitik berbasis machine learning untuk mendukung deteksi dini dan prediksi lebih akurat (Carmi & Theodoridou, 2019; Li et al., 2018).

## **2.7 Responsive Web Design (RWD) dalam Konteks Pengembangan Antarmuka Multiplatform**

Dalam konteks pengembangan dasbor dinamis seperti yang dirancang untuk *Bank Air Kami*, RWD memiliki peran signifikan untuk memastikan aksesibilitas pada berbagai platform. Lynch dan Horton dalam *Web Style Guide* menyoroti bahwa penerapan RWD harus memperhatikan prinsip-prinsip pengalaman pengguna yang baik, seperti kemudahan navigasi, keterbacaan, dan pengelolaan konten yang efisien (Patrick J. Lynch & Sarah Horton, 2016). Mereka juga mencatat bahwa desain responsif harus mempertimbangkan kebutuhan pengguna yang terus berubah, sekaligus memastikan antarmuka tetap intuitif meskipun diakses melalui perangkat dengan layar yang lebih kecil (Noorkaran Bhanarkar, Aditi Paul, & Dr. Ashima Mehta, 2023; Patrick J. Lynch & Sarah Horton, 2016).

Namun, penerapan elemen-elemen RWD sering kali menjadi subjek perdebatan. Marcotte, misalnya, berpendapat bahwa media query adalah kunci utama dalam mendukung fleksibilitas desain, karena memungkinkan pengembang menetapkan aturan CSS spesifik untuk berbagai resolusi layer (Marcotte & Keith, 2011). Di sisi lain, Lynch dan Horton lebih menekankan pentingnya mengintegrasikan strategi desain berbasis pengguna, yang tidak hanya berfokus pada aspek teknis, tetapi juga pada pengalaman visual dan kenyamanan interaksi pengguna (Patrick J. Lynch & Sarah Horton, 2016). Penelitian oleh Bhanarkar et al. juga mendukung pandangan ini dengan menyoroti bahwa desain responsif dapat mengurangi interaksi yang tidak perlu, seperti menggulir horizontal atau klik berlebih, sehingga meningkatkan efisiensi pengguna. Selain itu, mereka menekankan pentingnya menyederhanakan elemen antarmuka untuk memastikan keterbacaan pada perangkat kecil (Noorkaran Bhanarkar et al., 2023).

Keuntungan strategis lain dari RWD adalah penghematan waktu pengembangan karena tidak memerlukan versi situs terpisah untuk perangkat tertentu. Fitur seperti penyederhanaan elemen halaman dan penyesuaian ukuran font serta gambar memungkinkan pengguna untuk mengakses informasi dengan nyaman tanpa harus menggulir secara horizontal. Namun, ada tantangan dalam memastikan bahwa elemen penting tetap terlihat, terutama pada perangkat dengan layar kecil (Noorkaran Bhanarkar et al., 2023). Hal ini menunjukkan pentingnya pendekatan desain yang seimbang antara kebutuhan teknis dan kepuasan pengguna.

Secara keseluruhan, penerapan RWD pada dasbor *Bank Air Kami* tidak hanya akan mendukung fleksibilitas desain, tetapi juga menjawab kebutuhan aksesibilitas lintas perangkat yang semakin relevan di era digital saat ini. Dengan memadukan aspek teknis dan strategis,

RWD memastikan bahwa dasbor dapat berkembang secara dinamis sesuai dengan perubahan kebutuhan pengguna dan teknologi. Hal ini sejalan dengan tujuan menciptakan antarmuka yang inklusif, efisien, dan responsif terhadap tantangan masa depan.

## 2.8 *Cloud Computing* Dalam Pengembangan Aplikasi Web

*Cloud computing* telah menjadi elemen esensial dalam pengembangan aplikasi web modern, terutama dalam mendukung skalabilitas proses bisnis. *Cloud computing* memungkinkan organisasi untuk memanfaatkan sumber daya komputasi secara dinamis sesuai kebutuhan, mengurangi biaya operasional, dan meningkatkan efisiensi. Menurut Ahmad dan Andras (2019), kemampuan skalabilitas pada platform *cloud* bergantung pada kemampuannya untuk menyesuaikan kapasitas layanan berdasarkan kebutuhan pengguna secara *realtime*, memastikan efisiensi penggunaan sumber daya (Al-Said Ahmad & Andras, 2019).

*Cloud computing* mendukung pengembangan aplikasi web dengan kemampuan untuk menyesuaikan skala sumber daya tanpa memerlukan konfigurasi fisik tambahan. Dengan fitur *auto-scaling*, platform *cloud* dapat meningkatkan jumlah instans *server* selama periode permintaan tinggi dan menurunkannya saat tidak diperlukan, memastikan penghematan biaya operasional. Hal ini memungkinkan bisnis untuk mengakomodasi fluktuasi beban kerja tanpa perlu melakukan investasi infrastruktur besar sebelumnya (Al-Said Ahmad & Andras, 2019; Gupta, Mittal, & Mufti, 2021).

Pendekatan *serverless* atau *Function as a Service* (FaaS) memberikan paradigma baru dalam pengembangan aplikasi web. Dalam model ini, pengembang hanya perlu fokus pada pembuatan fungsi yang akan dijalankan sesuai permintaan, sementara penyedia *cloud* bertanggung jawab atas pengelolaan sumber daya komputasi yang mendasarinya (Rajan, 2020). *Serverless* mendukung efisiensi biaya karena pengguna hanya membayar waktu eksekusi dan sumber daya yang digunakan, berbeda dengan pendekatan tradisional yang memerlukan pembayaran untuk kapasitas yang dialokasikan tetapi tidak selalu digunakan (Al-Said Ahmad & Andras, 2019). Namun, *serverless* memiliki keterbatasan, seperti latensi awal (*cold start*) yang dapat memengaruhi performa aplikasi berbasis waktu nyata. Selain itu, fungsi *serverless* cenderung bersifat *stateless*, sehingga tidak cocok untuk beban kerja yang memerlukan pengelolaan data secara berkelanjutan di antara pemanggilan fungsi (Malawski, Gajek, Zima, Balis, & Figiela, 2020; Rajan, 2020).

Teknologi *container*, seperti yang dijelaskan oleh Mufti et al. (2020), memberikan solusi optimal untuk aplikasi yang membutuhkan isolasi dan portabilitas. Dalam konteks pengembangan aplikasi web, *container* memungkinkan pengemasan aplikasi beserta

dependensinya ke dalam satu unit yang dapat dijalankan secara konsisten di berbagai lingkungan (Gupta et al., 2021). *Container* juga mendukung siklus hidup aplikasi yang lebih cepat karena memungkinkan otomatisasi dalam proses pengembangan, pengujian, dan penerapan. Sebagai perbandingan, pendekatan *serverless* lebih cocok untuk tugas ringan yang berjalan dalam waktu singkat, sedangkan *container* lebih efektif untuk aplikasi yang memerlukan kontrol penuh atas lingkungan eksekusi. Dalam beberapa kasus, kombinasi keduanya digunakan untuk memanfaatkan keunggulan masing-masing, seperti fleksibilitas *serverless* untuk fungsi-fungsi kecil dan *container* untuk komponen inti aplikasi (Malawski et al., 2020). Berikut merupakan table perbandingan penggunaan *server*, *serverless*, dan *container*:

Tabel 2.3 Perbandingan Deployment di *Cloud Computing*

Aspek	Server	Serverless	Container
Efisiensi Biaya	Membutuhkan biaya tetap tinggi untuk infrastruktur, termasuk pemeliharaan.	Model <i>pay-as-you-go</i> , biaya berdasarkan penggunaan.	Biaya fleksibel sesuai jumlah instans container yang berjalan.
Skalabilitas	Memerlukan konfigurasi manual untuk meningkatkan skala.	Skalabilitas otomatis berdasarkan kebutuhan beban.	Skalabilitas lebih cepat melalui orkestrasi.
Kecepatan Deploy	Lambat karena pengaturan server secara manual.	Sangat cepat, fokus hanya pada fungsi.	Cepat dengan penggunaan gambar container yang disiapkan.
Manajemen Data	Memungkinkan manajemen data yang kompleks dan berkelanjutan.	Stateless, kurang cocok untuk data yang memerlukan pemrosesan berkelanjutan.	Cocok untuk pengelolaan data dan isolasi yang lebih baik.
Kompleksitas	Tinggi, memerlukan pemeliharaan server.	Rendah, dikelola sepenuhnya oleh penyedia layanan.	Sedang, membutuhkan manajemen gambar dan orkestrasi.

Pemilihan antara *server*, *serverless*, dan *container* bergantung pada kebutuhan aplikasi dan karakteristik teknologi. Server tradisional membutuhkan biaya tetap tinggi dan konfigurasi manual untuk skalabilitas, menjadikannya cocok untuk aplikasi dengan kebutuhan data yang kompleks dan stabil (Al-Said Ahmad & Andras, 2019). *Serverless*, dengan model *pay-as-you-go* dan skalabilitas otomatis, ideal untuk aplikasi dinamis tetapi kurang sesuai untuk pengelolaan data berkelanjutan karena sifatnya yang stateless (Malawski et al., 2020). Container menawarkan fleksibilitas biaya, deployment cepat melalui gambar aplikasi, dan

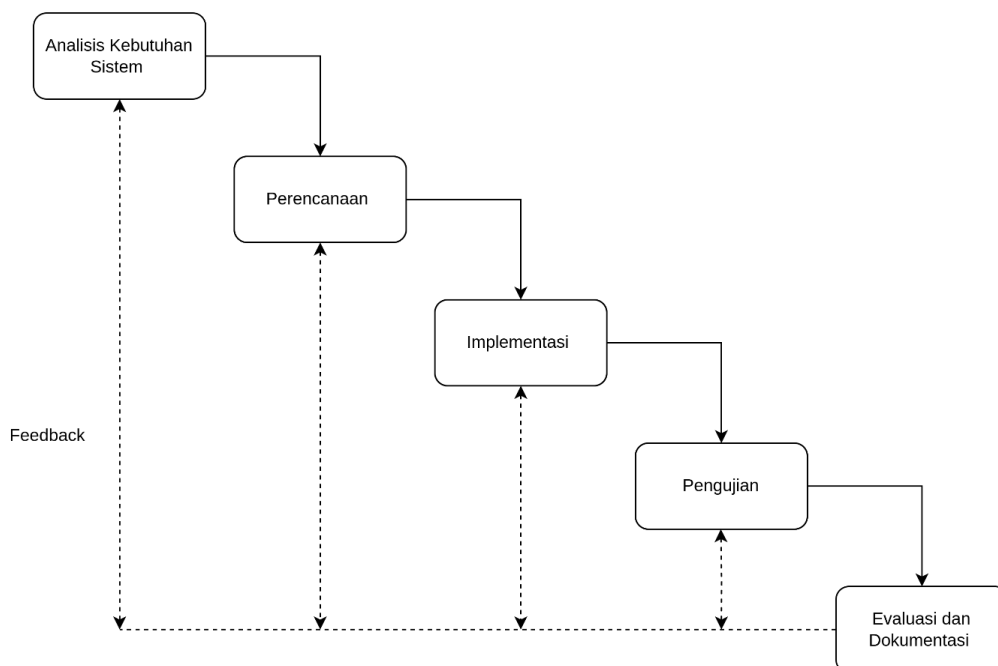
isolasi lingkungan, menjadikannya pilihan terbaik untuk aplikasi berbasis mikroservis dengan kebutuhan skalabilitas tinggi (Gupta et al., 2021)

Keuntungan utama dari *cloud computing* mencakup efisiensi biaya melalui model *pay-as-you-go* yang memungkinkan pengurangan pengeluaran operasional dengan menyesuaikan biaya berdasarkan penggunaan aktual (Al-Said Ahmad & Andras, 2019). Selain itu, kemampuan skalabilitas dinamis memungkinkan penyesuaian sumber daya secara otomatis untuk memenuhi kebutuhan beban kerja yang fluktuatif (Gupta et al., 2021). *Cloud computing* juga memberikan aksesibilitas global yang memungkinkan pengguna untuk mengakses aplikasi dari lokasi mana saja selama terhubung dengan internet (Al-Said Ahmad & Andras, 2019). Namun, teknologi ini menghadirkan tantangan seperti risiko kebocoran informasi karena data disimpan di *cloud*, yang memerlukan solusi keamanan yang kuat (Malawski et al., 2020). Selain itu, ketergantungan pada penyedia layanan dapat menimbulkan risiko vendor lock-in, sehingga organisasi harus berhati-hati dalam memilih penyedia *cloud* dan mempertimbangkan fleksibilitas serta portabilitas data mereka (Gupta et al., 2021).

### BAB III

## METODOLOGI

Bab ini membahas metodologi yang digunakan dalam pengembangan dasbor responsif berbasis *web* untuk Bank Air Kami. Penelitian ini menggunakan pendekatan penelitian pengembangan (*Research and Development/R&D*) dengan model pengembangan perangkat lunak *waterfall* yang dimodifikasi. Model ini terdiri atas lima tahap utama, yaitu analisis kebutuhan, perencanaan, implementasi, pengujian, serta evaluasi dan dokumentasi, seperti ditunjukkan pada Gambar 3.1. Meskipun digambarkan secara berurutan, dalam pelaksanaannya dimungkinkan adanya umpan balik antar tahap ketika pada proses implementasi dan pengujian ditemukan kebutuhan penyesuaian rancangan. Seluruh tahapan pengembangan dilakukan dengan memanfaatkan data sekunder dan data simulasi, tanpa pengumpulan data primer baru, sehingga fokus penelitian berada pada perancangan, implementasi, dan pengujian sistem berdasarkan proses bisnis usulan Bank Air Kami.



Gambar 3.1 Model Pengembangan *Waterfall* yang Dimodifikasi

Tahapan model *waterfall* yang dilalui dalam penelitian ini dimulai dari analisis kebutuhan, yaitu mengidentifikasi kebutuhan fungsional dan non-fungsional dasbor melalui telaah sistem dan penelitian Bank Air Kami sebelumnya, analisis kondisi terkini serta potensi

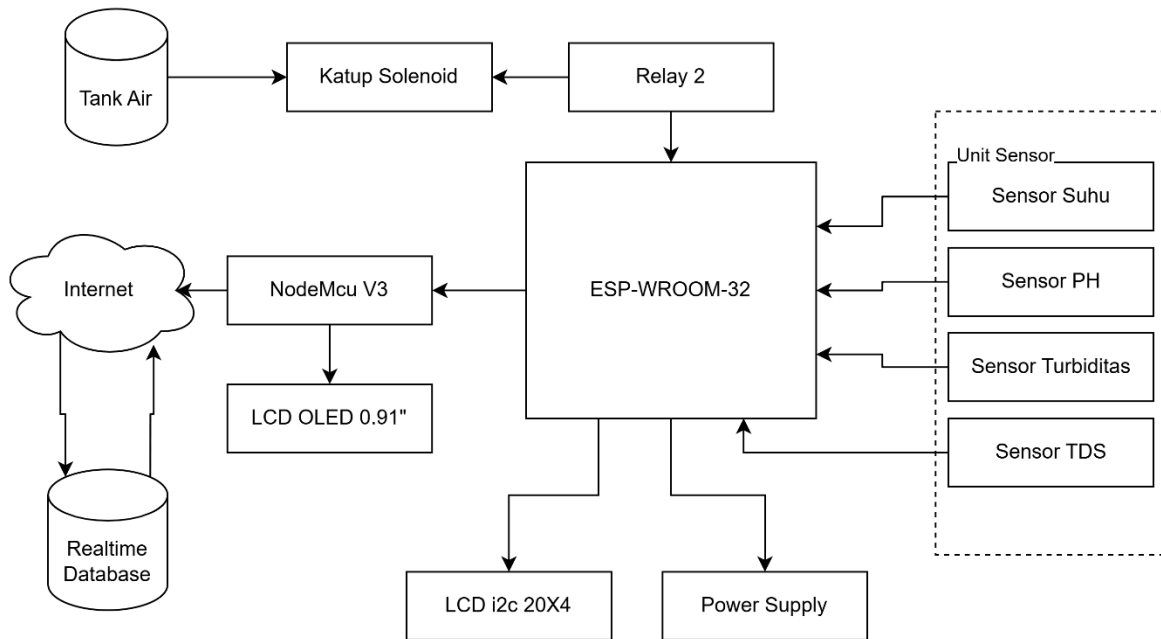
pengembangannya, dan kajian literatur. Tahap berikutnya adalah perencanaan, yang berfokus pada penentuan ruang lingkup pengembangan dalam bentuk *Minimum Viable Product* (MVP) serta penyusunan arsitektur aplikasi, desain basis data, desain infrastruktur, dan rancangan antarmuka dasbor. Tahap implementasi merealisasikan rancangan tersebut ke dalam bentuk aplikasi web, meliputi pengembangan *backend* dan *frontend* serta integrasi dengan data sekunder dan data simulasi yang digunakan sebagai dasar pengujian. Setelah modul-modul inti selesai dikembangkan, dilakukan pengujian fungsional dan pengujian integrasi untuk memastikan sistem berjalan sesuai kebutuhan yang telah dirumuskan. Tahap terakhir adalah evaluasi dan dokumentasi, di mana hasil pengujian dianalisis untuk menilai kesesuaian antara rancangan dan implementasi, mengidentifikasi keterbatasan sistem, serta menyusun dokumentasi teknis dan laporan penelitian sehingga seluruh proses pengembangan dapat dipertanggungjawabkan.

### **3.1 Kebutuhan sistem**

#### **3.1.1 Gambaran Sistem**

Secara umum, sistem yang dikembangkan dalam penelitian ini berupa dasbor *web* yang mendukung pemantauan kualitas air dan pengelolaan filter air pada Bank Air Kami berdasarkan proses bisnis usulan. Dasbor ini dirancang sebagai lapisan aplikasi yang berada di atas infrastruktur data yang telah dibangun pada penelitian sebelumnya, sehingga dapat memanfaatkan kembali konsep integrasi dengan perangkat IoT dan basis data berbasis cloud, serta tetap dapat berfungsi menggunakan data sekunder dan data simulasi ketika perangkat IoT tidak aktif secara kontinu.

Pada penelitian-penelitian terdahulu, Bank Air Kami telah mengimplementasikan perangkat IoT untuk mengukur parameter kualitas air seperti pH, suhu, kekeruhan, dan TDS, kemudian mengirimkan data ke layanan penyimpanan berbasis *cloud*. Penelitian ini mengadopsi arsitektur tersebut sebagai acuan, namun pada saat pelaksanaan penelitian, data kualitas air tidak lagi diperoleh secara waktu nyata dari perangkat IoT. Oleh karena itu, aliran data dalam sistem dimodelkan sebagai aliran dari sumber data sekunder dan data simulasi menuju basis data aplikasi dan layanan penyimpanan, yang kemudian diolah dan disajikan melalui dasbor *web*.

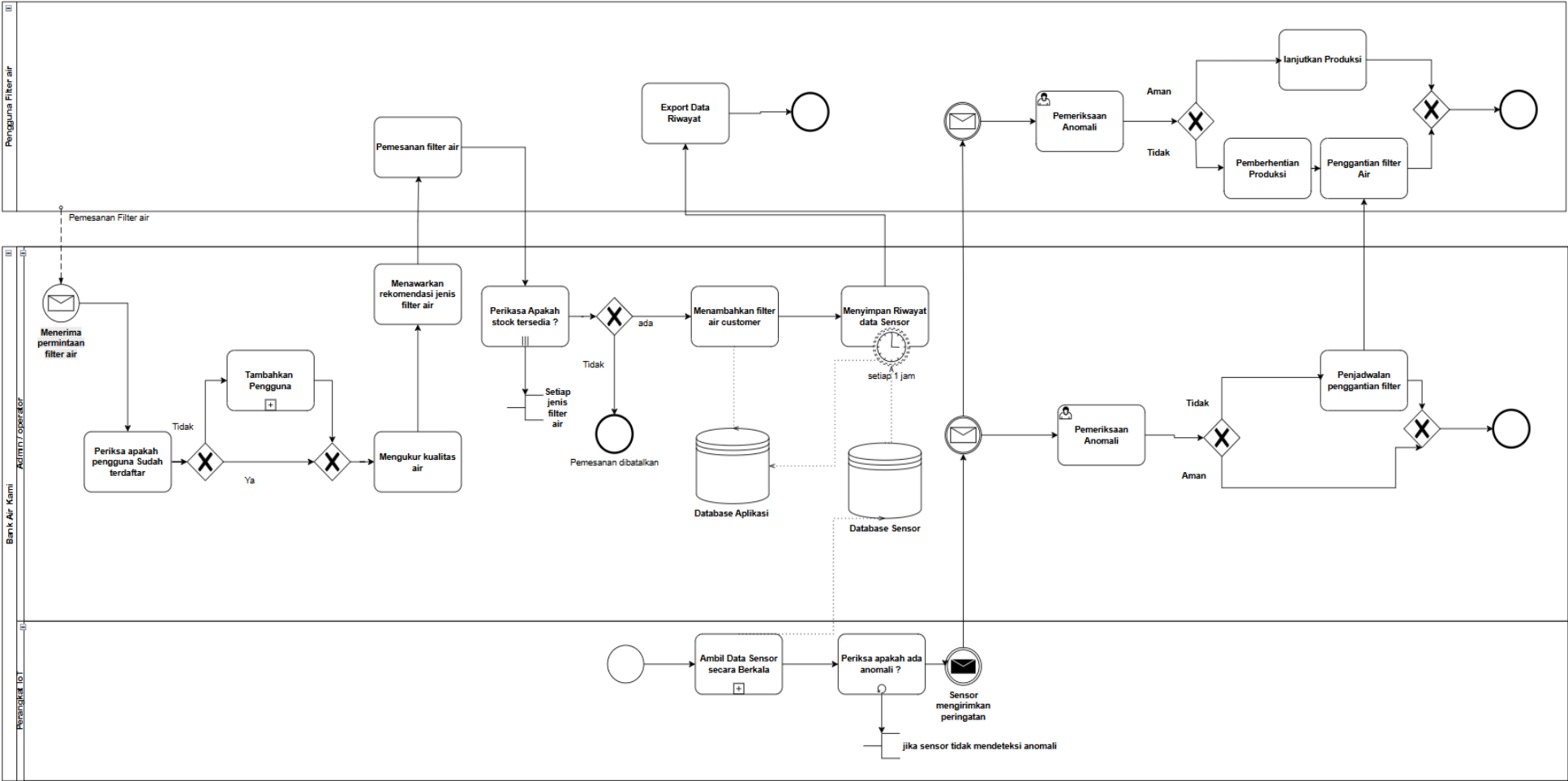


Gambar 3.2 Gambaran IoT Sistem dari Penelitian Sebelumnya

Dari sisi fungsional, sistem dibagi menjadi beberapa komponen utama, yaitu modul pemantauan kualitas air, modul pengelolaan filter air, dan modul manajemen pengguna. Modul pemantauan kualitas air bertugas menampilkan informasi parameter kualitas air dalam bentuk tabel dan visualisasi grafis yang memudahkan pengelola dalam membaca kecenderungan data. Modul pengelolaan filter air berfokus pada pencatatan dan pengelolaan stok serta penggunaan filter pada pelanggan, sehingga informasi terkait ketersediaan dan distribusi filter terdokumentasi dengan lebih terstruktur. Modul manajemen pengguna mengatur hak akses berdasarkan peran sehingga hanya pengguna yang berwenang yang dapat mengelola data tertentu. Hal ini diharapkan dapat memastikan bahwa skala operasi sistem dapat ditingkatkan tanpa perlu mengubah struktur dasar aplikasi. Untuk menilai aspek tersebut, digunakan beberapa parameter, yaitu: (1) kompatibilitas perangkat, (2) peningkatan skalabilitas tata kelola air, dan (3) efektivitas sistem dalam memberikan informasi kepada pengelola. Gambaran ini menjadi dasar bagi analisis kebutuhan dan perancangan yang dijelaskan pada subbab berikutnya.

### 3.1.2 Analisis Kebutuhan Sistem

Untuk memahami kebutuhan dari sistem maka perlu mengetahui proses bisnis usulan pada sistem Bank Air Kami. Berikut merupakan BPMN (*Business Process Model and Notation*) dari Bank Air Kami yang akan dibuat berdasarkan proses bisnis.



Gambar 3.3 Diagram BPMN Proses Bisnis Usulan Pengelolaan Filter Air pada Bank Air Kami

Berikut alur per poin untuk setiap role/aktor berdasarkan Gambar 3.3:

- a. Pengguna Filter Air:
  1. Mengajukan permintaan/pemesanan filter
  2. Menerima penawaran rekomendasi jenis filter
  3. Menerima pemasangan/penggantian filter dan memastikan filter aktif digunakan.
  4. Mengakses dasbor untuk memantau informasi kualitas air pada titik terkait serta melakukan ekspor data.
  5. Jika kualitas air terdeteksi menurun atau berada di luar ambang batas wajar (anomali), pengguna menerima peringatan/notifikasi dari sistem.
  6. Jika menerima peringatan, pengguna dapat memilih: (1) mematikan penggunaan filter sementara dan meminta penggantian oleh operator, atau (2) tetap mengoperasikan filter.
- b. Operator/Admin Bank Air Kami:
  1. Menerima dan memverifikasi permintaan/pemesanan filter dari pengguna filter air.
  2. Melakukan pengecekan kualitas air di lokasi pengguna.
  3. Menentukan/menyarankan jenis filter yang sesuai berdasarkan kebutuhan dan kualitas air.
  4. Mengecek ketersediaan stok filter pada sistem.
  5. Jika stok tersedia: melanjutkan proses penugasan/persiapan pemasangan.
  6. Jika stok tidak tersedia, mencatat status pemesanan sebagai tidak dapat dipenuhi dan menginformasikan pembatalan kepada pengguna.
  7. Melakukan pemasangan/penggantian filter.
  8. Memantau data kualitas air dan status filter melalui dasbor secara berkala untuk melihat tren dan indikasi anomali.
- c. Perangkat IoT:
  1. Pengumpulan dan Penyimpanan Data Sensor (IoT) kedalam *database* sensor.
  2. Mengirimkan data sensor dari *database* sensor ke sistem aplikasi agar data dapat dicatat sebagai riwayat kualitas air.
  3. Jika berdasarkan data yang dikirim terdeteksi kondisi anomali, sistem akan mengirimkan peringatan keada operator/admin dan pengguna.

### **Analisis Kebutuhan**

Berikut merupakan analisis kebutuhan fungsional dan Non-Fungsional berdasarkan Gambar 3.3:

- a. Analisis Kebutuhan Fungsional mencakup fitur inti yang ada pada dasbor, yaitu:
  1. Pemantauan, Pengumpulan dan Penyimpanan Data Sensor (IoT): Menampilkan parameter kualitas air seperti pH, suhu, turbiditas, TDS.
  2. Pembaharuan Visualisasi data: Menggunakan jenis visualisasi data yang sesuai untuk memberikan informasi yang tepat.
  3. Manajemen pengguna: sistem mendukung pengelolaan akun dengan memberikan sumber daya dan data yang tepat berdasarkan peran.
  4. Manajemen filter air dan sensor: sistem dapat mengelola filter air yang tersedia dan mengelola filter air
  5. Antarmuka Responsif: Dasbor harus dapat diakses melalui berbagai perangkat, seperti komputer, tablet, dan smartphone.
  6. Analisis dan Rekomendasi jenis filter: Menyediakan saran berbasis data untuk jenis filter air yang sesuai dengan kondisi kualitas air di lokasi tertentu.
  7. Pelaporan: Sistem bisa mengekspor data untuk kebutuhan laporan riwayat kualitas air.
  8. Notifikasi: mengirimkan notifikasi ketika adanya perubahan yang tidak wajar pada data sensor.
  9. Pencatatan riwayat pengelolaan filter: dapat menyimpan data histori penggantian filter air
- b. Analisis Kebutuhan Non-Fungsional adalah persyaratan yang terkait dengan kualitas sistem yang dikembangkan, seperti performa, keamanan, dan keandalan. Kebutuhan non-fungsional dalam sistem ini mencakup:
  1. Responsivitas Antarmuka: Dasbor harus dapat ditampilkan dan digunakan dengan baik di berbagai perangkat (komputer, tablet, ponsel).
  2. Skalabilitas: Sistem harus mampu berkembang untuk mengakomodasi penambahan titik sensor baru di seluruh Kampung Terban, serta penambahan pengguna dalam jumlah besar seiring waktu.
  3. Efektivitas Sistem dalam Memberikan Informasi: Antarmuka perlu jelas, mudah dipahami, dan menyajikan data dengan visualisasi yang membantu pengambilan keputusan.

## 3.2 Perencanaan

### 3.2.1 *Minimum Viable Product (MVP) Penelitian*

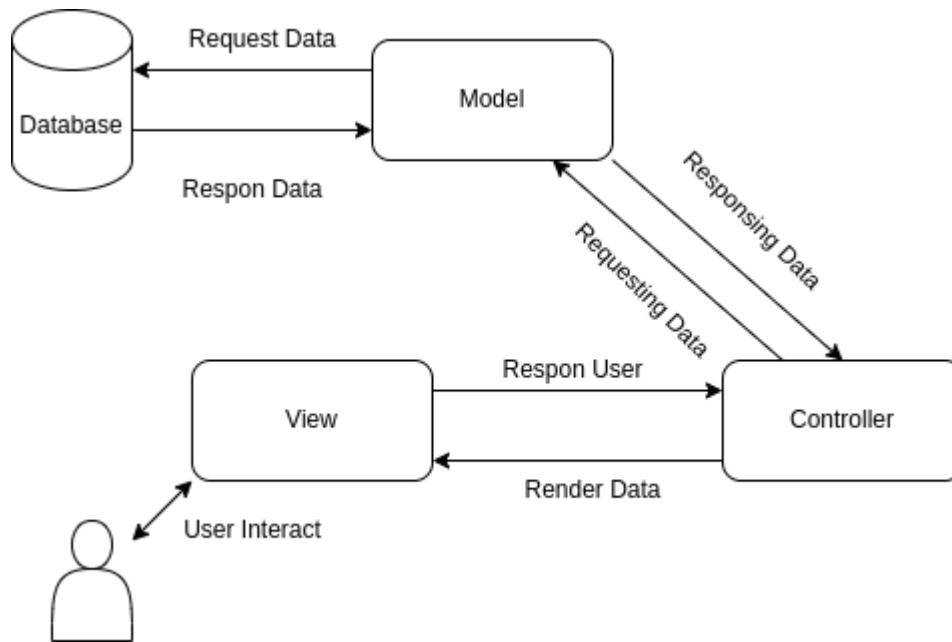
Pada tahap perencanaan, konsep *Minimum Viable Product (MVP)* diterapkan untuk mengarahkan pengembangan awal hanya pada fitur-fitur paling kritis yang secara langsung

mendukung tujuan penelitian, yaitu meningkatkan keandalan pemantauan kualitas air sekaligus menyiapkan fondasi skalabilitas proses bisnis Bank Air Kami. MVP ditentukan melalui matriks prioritas yang menimbang urgensi kebutuhan pengguna, nilai fungsional terhadap alur kerja operasional, serta beban implementasi.

Berdasarkan pertimbangan tersebut, enam fitur dianggap esensial untuk rilis awal, yaitu: (1) pemantauan, pengumpulan, dan penyimpanan data kualitas air dari sensor yang dirancang untuk mendukung pemrosesan waktu nyata ketika infrastruktur IoT aktif, dengan pengujian pada penelitian ini menggunakan data sekunder dan data simulasi; (2) pembaruan visualisasi data berbasis grafik dan peta interaktif; (3) manajemen pengguna dengan model peran (*role-based access control*/RBAC); (4) pelaporan melalui ekspor data; (5) antarmuka responsif lintas perangkat; dan (6) mekanisme skalabilitas horizontal untuk penambahan titik sensor. Fitur-fitur tersebut diharapkan memastikan dasbor mampu menampilkan kondisi kualitas air secara terkini berdasarkan data yang tersedia, mempermudah analisis cepat, serta menjaga keamanan akses meskipun jumlah pengguna dan sensor berkembang.

### **3.2.2 Arsitektur Perangkat Lunak**

Sistem ini memanfaatkan arsitektur *Model-View-Controller* (MVC) , yang memisahkan aplikasi menjadi tiga komponen utama: *Model*, *View*, dan *Controller* (Gambar 3.4). Pendekatan MVC digunakan untuk memfasilitasi pengembangan yang modular dan memungkinkan pengelolaan kode yang lebih mudah.

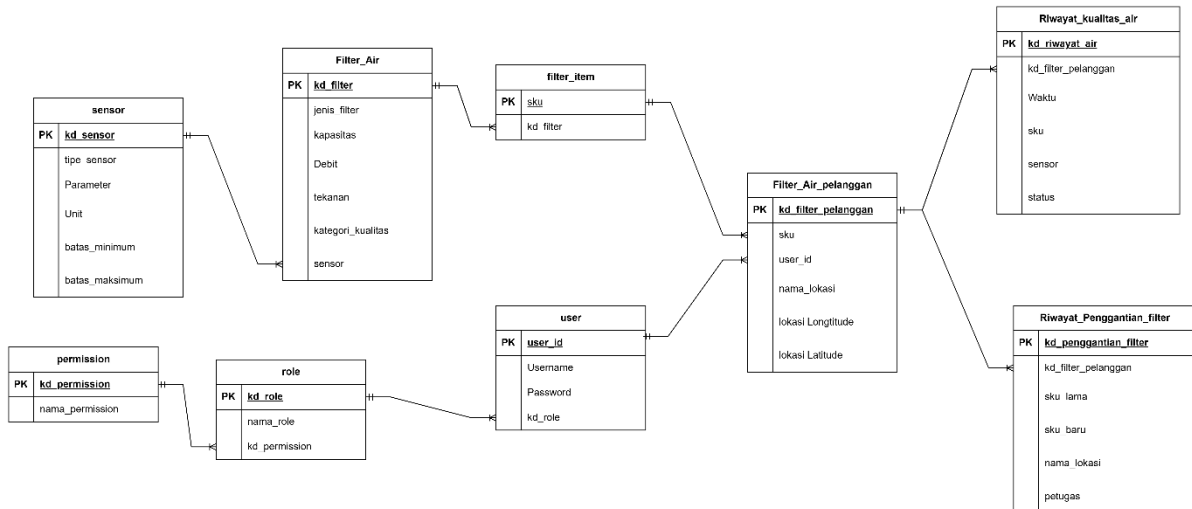
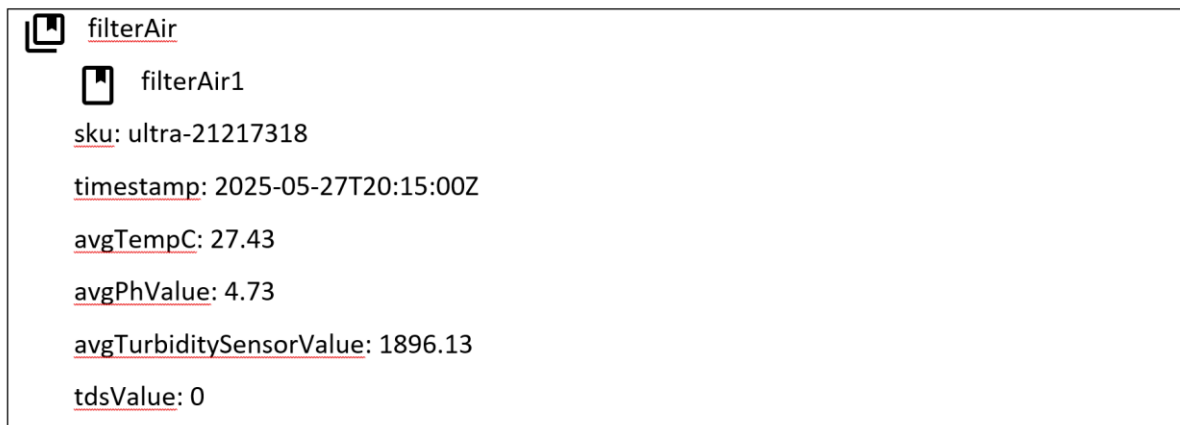


Gambar 3.4 Arsitektur MVC

Dalam kerangka MVC, Model berperan sebagai fondasi data sekaligus pusat logika bisnis: di dalamnya dirumuskan skema basis data, aturan validasi, relasi antar tabel, serta proses pengolahan data sensor sehingga informasi tersimpan konsisten dan sah. *Controller* menjadi jembatan antara pengguna dan sistem; setiap permintaan HTTPS melewati *middleware* autentikasi-RBAC untuk verifikasi, kemudian memanggil fungsi Model yang relevan sebelum menyiapkan respons. Adapun View menghadirkan lapisan presentasi yang memvisualisasikan data grafik, peta interaktif, dan tabel dinamis secara responsif pada berbagai perangkat, memungkinkan pengguna memantau kualitas air secara intuitif dan *realtime*.

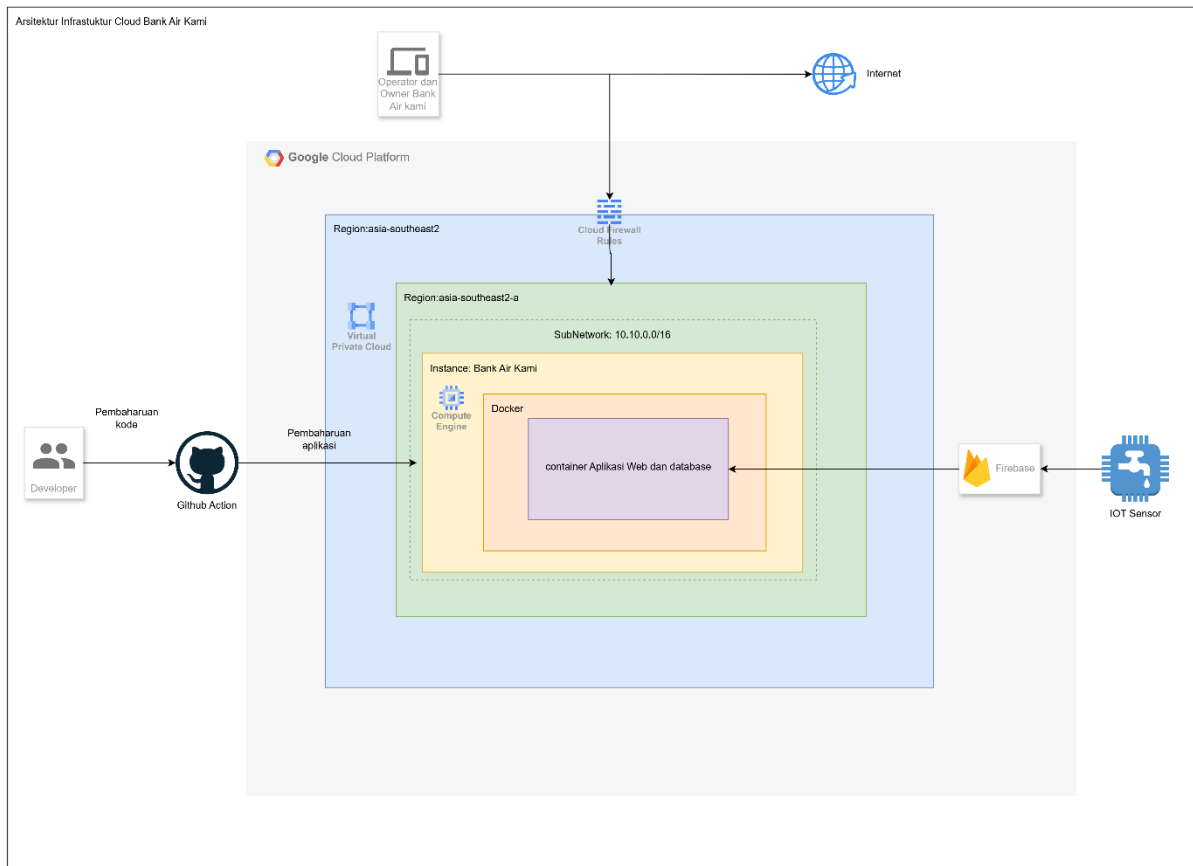
### 3.2.3 Arsitektur dan Desain *Database*

Untuk Dasbor Bank Air Kami, sistem menggunakan Firebase dengan arsitektur dokumen untuk mengambil data sensor filter secara streaming dan MySQL dengan arsitektur RDBMS (*Relational Database Management System*) menyimpan data dari aplikasi *web*. Pemisahan antara *database* sensor filter dan *web* aplikasi di dasari penelitian (Atalaric & Sujarwo, 2024) yang menggunakan Firebase untuk mengambil data dari sensor. Selain itu Firebase merupakan *database* NoSQL yang dapat melakukan data streaming secara *realtime* yang tidak bisa dilakukan oleh MySQL yang berbasis SQL. Berikut desain skema relasional digambarkan melalui *Entity-Relationship Diagram* (ERD) dan desain basis data NoSQL disajikan dalam bentuk model dokumen.

Gambar 3.5 Desain *Database* SQLGambar 3.6 Desain *Database* Dokumen

### 3.2.4 Desain Infrastruktur

Pada desain infrastruktur Dasbor Bank Air Kami, seluruh layanan ditempatkan di Google Cloud Platform (GCP) untuk memudahkan pengelolaan dan skalabilitas. Sensor-sensor IoT di lapangan mengukur kualitas air, kemudian mengirimkan data secara nirkabel ke Firebase Realtime Database. Sementara itu, data aplikasi *web* seperti akun pengguna dan konfigurasi disimpan di MySQL. Aplikasi *web* dijalankan dalam kontainer Docker pada layanan Compute Engine. Apabila terjadi pembaruan kode, GitHub Actions menjalankan *pipeline* pengujian, *build*, dan *deploy* aplikasi ke Compute Engine secara otomatis.

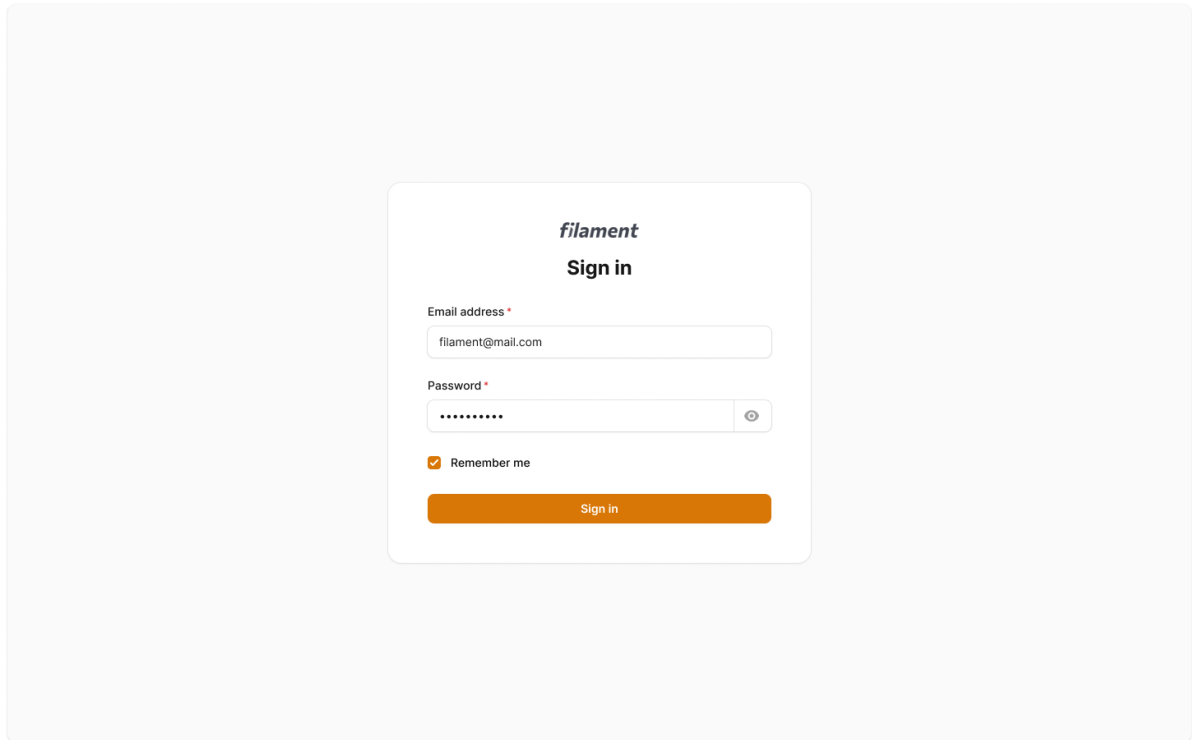


Gambar 3.7 Desain Infrastruktur Cloud

### 3.2.5 Rancangan Desain Antarmuka

#### Rancangan Halaman Login

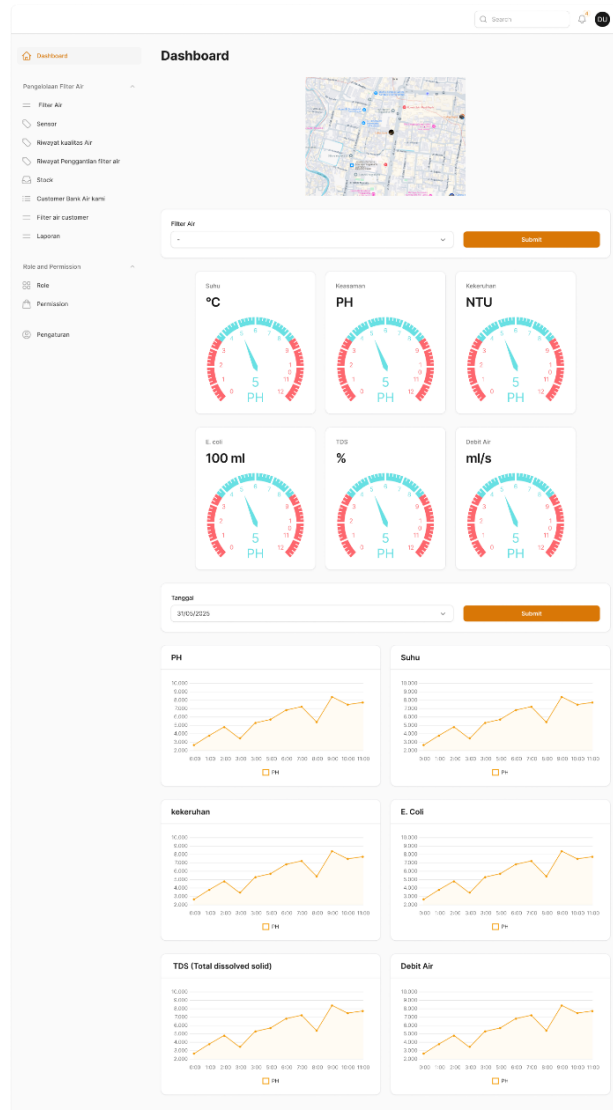
Gambar 3.8 merupakan rancangan antarmuka ketika pengguna aplikasi (admin/operator dan pengguna filter air) melakukan login. Untuk mengakses halaman selanjutnya pengguna harus melakukan login terlebih dahulu untuk mengidentifikasi identitas, dan sumber daya berdasarkan otoritas pengguna. Setelah melakukan login pengguna akan dialihkan ke halaman dasbor.



Gambar 3.8 Halaman Login

### Rancangan Halaman Dasbor

Gambar 3.9 merupakan rancangan antarmuka yang menyajikan rangkuman dari data sensor berdasarkan master data filter air yang telah dipilih. Pada halaman ini, pengguna dapat melihat ringkasan nilai sensor secara *realtime*, lokasi filter air dengan GIS, dan riwayat nilai sensor. Tampilan ini dibagi ke dalam tiga komponen utama agar informasi mudah dipahami dalam sekali pandang. Di bagian teratas, menampilkan lokasi filter air dan enam kartu ringkasan yang menampilkan sensor yang digunakan pada filter air. Untuk bagian bawah terdapat enam grafik riwayat dari sensor yang digunakan pada filter air selama 24 jam terakhir.



Gambar 3.9 Halaman Dasbor Utama

### Rancangan Halaman Filter Air Pengguna

Gambar 3.10 menggambarkan halaman manajemen filter air yang dikategorikan berdasarkan filter air yang dimiliki pengguna. Halaman ini akan menyimpan informasi mengenai nama pengguna, SKU filter air, jenis filter, *Longitude*, *Latitude* dan nama lokasi.

**Filter air pengguna**

no	Nama pengguna	SKU filter	Jenis Filter	Nama Lokasi	View	Edit	Delete
1	Budi	UL- 22055	Karbon aktif	lokasi A	View	Edit	Delete
2	Budi	UL- 22055	Pasir	lokasi A	View	Edit	Delete
3	Budi	UL- 22055	UltraFiltrasi	lokasi C	View	Edit	Delete
4	Budi	UL- 22055	NanoFiltrasi	lokasi D	View	Edit	Delete
5	Budi	UL- 22055	Pasir	lokasi E	View	Edit	Delete

Per page: 5 | Next

Gambar 3.10 Halaman Indeks Filter Air Pengguna

### Rancangan Halaman Informasi Filter Air

Gambar 3.11 merupakan rancangan halaman manajemen filter air yang di kategorikan berdasarkan jenis filter air. Halaman ini akan menyimpan informasi tentang sku filter air, Jenis filter air, kapasitas, debit, tekanan, kategori kualitas filter dan sensor yang digunakan.

The screenshot shows a web application interface for managing water filters. The main content area is titled "Filter Air" and contains a table with the following data:

SKU Filter	Jenis Filter	kapasitas	Kualitas	
OR319999	Karbon aktif	1000L	A	View Edit Delete
OR11531	UltraFiltrasi	1000L	B	View Edit Delete
OR983123	NanoFiltrasi	1000L	A	View Edit Delete
OR369459	Pasir	1000L	C	View Edit Delete
OR44823	NanoFiltrasi	1000L	B	View Edit Delete
OR44823	NanoFiltrasi	1000L	A	View Edit Delete

At the bottom of the table, there is a pagination control showing "Per page 5" and a "Next" button.

Gambar 3.11 Halaman Indeks filter Air

### Rancangan Halaman Riwayat Penggantian Filter Air

Gambar 3.12 merupakan rancangan halaman untuk menyimpan riwayat penggantian filter air yang telah dilakukan. Halaman ini akan menyimpan informasi tentang tanggal penggantian filter, lokasi sku filter lama, sku filter baru, dan operator yang melakukan penggantian filter.

The screenshot displays a web interface for managing filter replacement records. The main content area is titled "Riwayat Penggantian filter" and contains a table with the following data:

Tanggal Penggantian	Lokasi	SKU Filter Lama	Filter Baru	Operator	View	Edit	Delete
2025-01-15	Lokasi A	A1-20240910	A1-20250115	Budi Santoso	View	Edit	Delete
2025-02-20	Lokasi B	B2-20241005	B2-20250220	Budi Santoso	View	Edit	Delete
2025-03-10	Lokasi B	C1-20241112	C1-20250310	Budi Santoso	View	Edit	Delete
2025-03-25	Lokasi C	D3-20241201	D3-20250325	Budi Santoso	View	Edit	Delete
2025-04-30	Lokasi D	D3-20241201	B2-20250430	Toni Wijaya	View	Edit	Delete
2025-04-30	Lokasi B	B2-20250220	B2-20250430	Toni Wijaya	View	Edit	Delete

The interface also includes a search bar, a "Tambah" (Add) button, and pagination controls showing "Per page 5" and a "Next" button.

Gambar 3.12 Halaman Riwayat Penggantian Filter

### Rancangan Halaman Riwayat Kualitas Air

Gambar 3.13 merupakan rancangan halaman untuk menyimpan riwayat kualitas air dari setiap filter air yang berjalan. Halaman ini akan menyimpan informasi tentang waktu pengambilan data sensor, lokasi, sku filter, sensor yang digunakan dan status kualitasnya.

**Riwayat kualitas Air**

Waktu	Lokasi	Filter	pH	E. Coli (/100ml)	kekeruhan (NTU)	TDS (mg/L)	Status Kualitas
10/05/2025 8:00:00	Lokasi A	Filter A1	7,2	3	0,6	386	Tidak Memenuhi
2025-05-10 8:00:00	Lokasi A	Filter B1	7	0	1,4	386	Memenuhi
2025-05-10 9:00:00	Lokasi A	Filter A1	6,9	0	1,7	278	Memenuhi
2025-05-10 9:00:00	Lokasi A	Filter C1	6,9	0	1,7	461	Tidak Memenuhi
2025-05-10 9:00:00	Lokasi A	Filter B2	7,1	0	06	461	Tidak Memenuhi

Per page 5 Next

Gambar 3.13 Halaman Riwayat Kualitas Air

### Rancangan Halaman Informasi Sensor

Gambar 3.14 merupakan rancangan halaman untuk informasi alat sensor sebagai indikator kualitas air yang digunakan untuk filter air. Halaman ini akan berisikan informasi tentang kode sensor, parameter sensor, satuan, batas minimum, dan batas maksimum.

**Sensor**

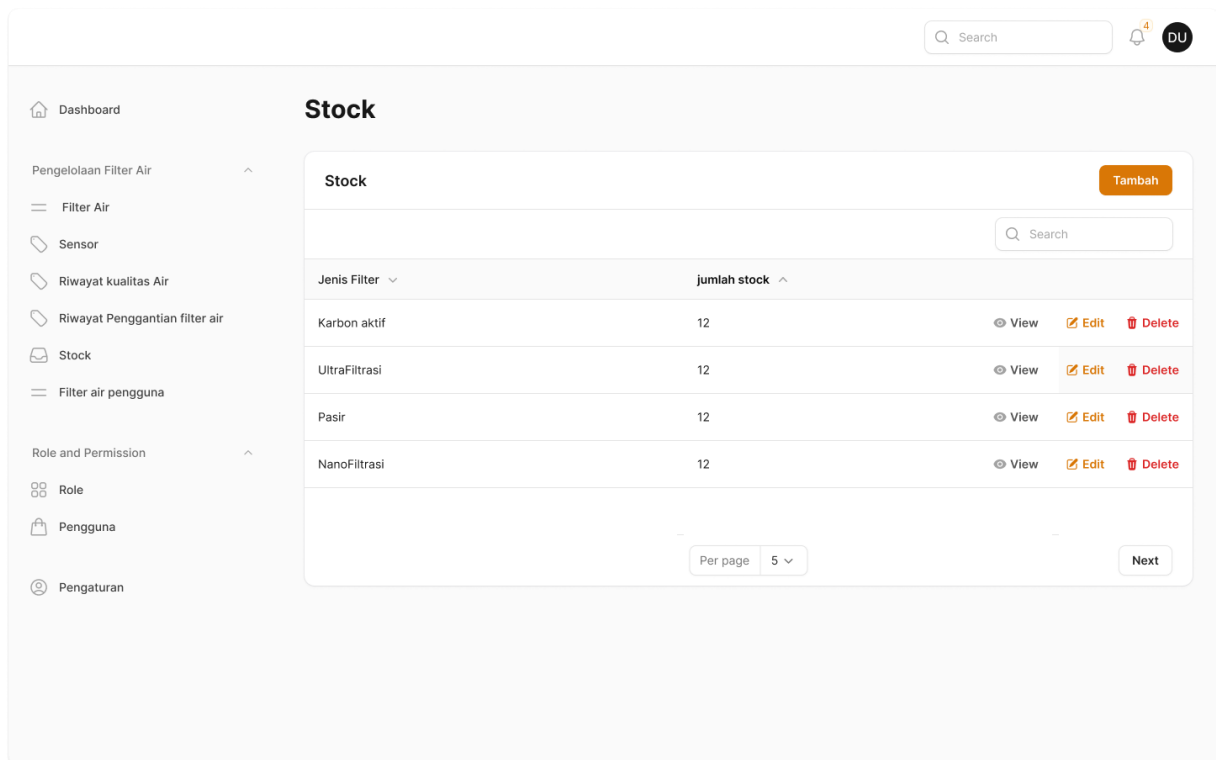
Kode Sensor	Parameter	Satuan	batas minimum	batas maksimum	
OR319999	Suhu	C	25	42	View Edit Delete
OR319999	E. coli	100/ml	0	0	View Edit Delete
OR983123	Debit	m <sup>3</sup> /s	10 m <sup>3</sup> /s	15 m <sup>3</sup> /s	View Edit Delete
OR983123	TDS	mg/l	0 mg/l	500 mg/l	View Edit Delete
OR983123	kekeruhan	NTU	1 NTU	25 NTU	View Edit Delete

Per page 5 Next

Gambar 3.14 Halaman Indeks Sensor

### Rancangan Halaman Pengelolaan Stok

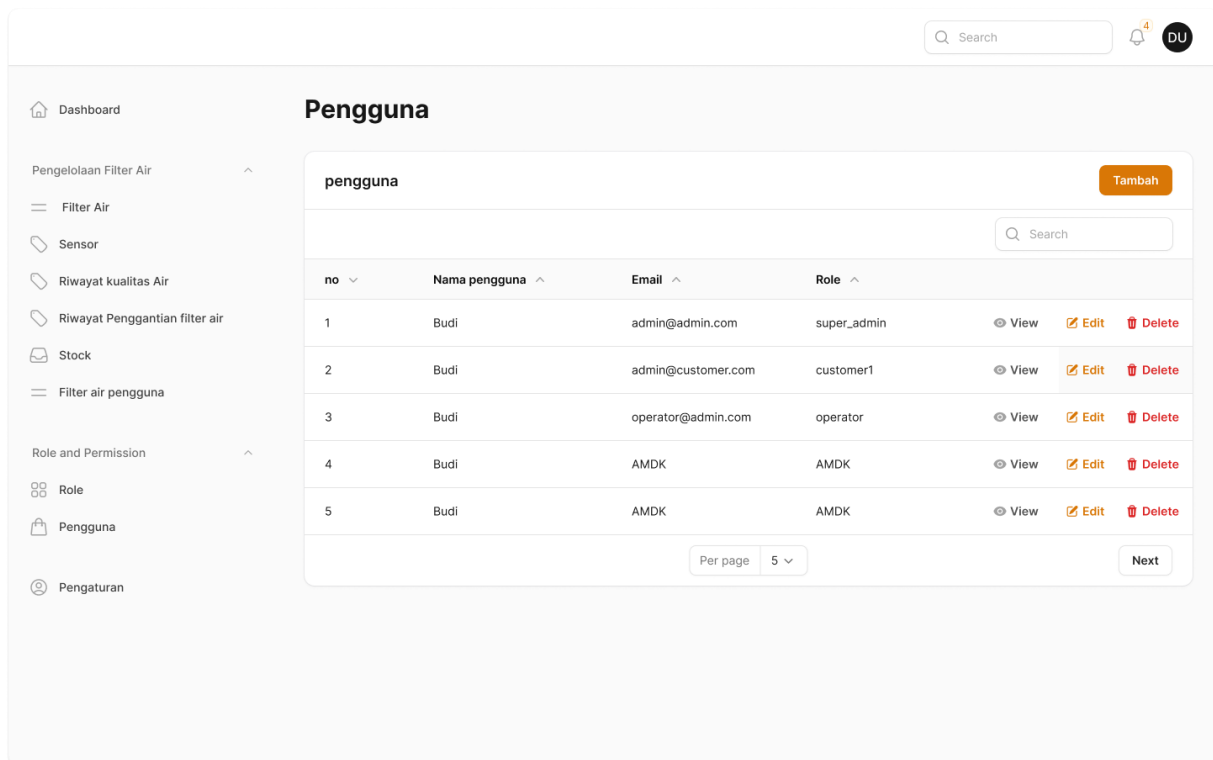
Gambar 3.15 merupakan rancangan halaman untuk menyimpan informasi stok filter air yang dimiliki oleh admin berdasarkan jenis filter air. Halaman ini akan berisikan informasi tentang SKU filter, jenis filter air, dan jumlah stok saat ini.



Gambar 3.15 Halaman Indeks Stok

### Rancangan Halaman Pengelolaan Pengguna Website

Gambar 3.16 merupakan rancangan halaman untuk mengelola akun pengguna berdasarkan role yang dimiliki. Halaman ini akan berisikan informasi nama pengguna, password, email, konfirmasi password dan role.



Gambar 3.16 Halaman Indeks User

### 3.3 Rencana Implementasi

#### 3.3.1 Lingkungan Pengembangan

Penelitian ini mengembangkan aplikasi *web* menggunakan Laravel Filament sebagai pondasi utama *backend* dan *frontend*. Kerangka ini dipilih karena menyediakan arsitektur MVC yang terintegrasi dengan komponen antarmuka yang siap pakai, sehingga dapat mempercepat proses *prototyping* aplikasi. Pada bagian *backend*, filament akan memanfaatkan kemampuan laravel ORM untuk memodelkan data yang ingin dibangun selaras dengan *database*. Fitur Resource & Relation Manager-nya menyederhanakan pembuatan antarmuka administrasi sehingga dapat mengurangi penulisan kode yang berulang. Disisi *frontend*, integrasi Filament dengan Livewire memungkinkan komponen antarmuka bereaksi secara *realtime* tanpa memuat ulang halaman, sementara dukungannya terhadap Tailwind CSS mempercepat penyusunan tata letak responsif dengan konsistensi visual yang terjaga. Untuk menjaga konsistensi konfigurasi serta meminimalkan *dependency drift*, seluruh stack dijalankan di dalam kontainer Docker. Berikut merupakan rincian versi dan fungsi setiap komponen yang akan digunakan:

Tabel 3.1 Komponen Aplikasi yang Digunakan

No	Komponen	Versi	Catatan singkat
1	PHP	8.4.x	Rilis stabil untuk pengembangan
2	Composer	2.8	Pengelola dependensi PHP
3	Laravel Framework	12.x	Fondasi MVC utama aplikasi
4	Laravel Filament	3.x	Generator UI admin + komponen siap pakai.
5	Node.js	22.04 LTS	Kompilasi aset (Vite/Tailwind)
6	NPM	1.8.x	Manajer paket bawaan Node 22
7	Docker engine	20.10.12	Pengelolaan kontainer di lingkungan pengembangan
8	Fedora	Fedora 42	Sistem operasi yang digunakan selama pengembangan
9	MySQL	8.0.0	RDBMS untuk data aplikasi <i>web</i>
10	PhpStorm	2025.1	IDE utama yang digunakan
11	Github action		Untuk mengelola CI/CD testing dan deployment

### 3.3.2 Manajemen Versi & CI/CD

Untuk *Version control system* (VCS), penelitian ini memilih GitHub karena memiliki ekosistem terpadu yang mendukung alur kerja pengembangan perangkat lunak modern. Repositori GitHub memungkinkan pelacakan perubahan kode, penyusunan *branch*, dan keberlanjutan proyek. Fitur GitHub Actions pada GitHub sangat berguna untuk melakukan otomisasi pipeline CI/CD yang melakukan pengujian aplikasi dan pembaharuan aplikasi. penggunaan github juga akan dipakai untuk melakukan dokumentasi dan pedoman untuk kelanjutan dari penelitian ini.

Tahapan pipeline dirancang dalam 2 stage terpisah yang saling bergantung. Stage testing menjalankan seluruh laravel test suite (Pest/PHPUnit) di GitHub Actions dan mengukur *code coverage*. Build hanya akan dapat dilanjutkan bila cakupan melebihi 80%; apabila tahapan itu tidak terpenuhi, workflow akan secara otomatis berhenti. Stage development akan terpicu ketika stage testing berhasil dijalankan. Pada fase ini, action akan melakukan remote ke vm tujuan, menyalin berkas proyek, membuat image laravel dan kemudian mengeksekusi kontainernya.

### 3.3.3 Setup Infrastruktur

Bagian ini akan menguraikan prosedur dalam penyediaan sumber daya yang diperlukan aplikasi untuk dapat berjalan yang stabil, aman dan skalabilitas yang fleksibel. Seluruh proses dijalankan sekali di akhir proyek. Seperti desain infrastruktur pada Gambar 3.7 memperlihatkan hubungan antar-komponen, sedangkan Tabel 3.2 di bawah merangkum spesifikasi inti yang diterapkan.

Tabel 3.2 Spesifikasi Infrastruktur

No.	Komponen	Konfigurasi Utama
1	Region	Asia-southeast2
2	VPC Network	Bank-air-kami
3	Cloud Firewall Rules	Allow-http (TCP 80) Allow-https (TCP 443) Allow-ssh (TCP 22)
4	Compute Engine VM	E2-medium (2 vcpus, 4 GB Memory)
5	Firebase/Firestore	Realtime database

Konfigurasi minimalis tersebut telah mencukupi kebutuhan penelitian: VM menjalankan kontainer aplikasi dan *database* MySQL secara lokal, Firebase menangani aliran data sensor IoT, sementara VPC dan aturan firewall memastikan lalu-lintas hanya melalui port yang diperlukan.

## 3.4 Pengujian

Pada bagian ini akan melakukan pengujian berdasarkan ketentuan yang telah dipaparkan pada Subbab 3.2 (Perancangan Sistem) dan diimplementasikan melalui prosedur pada Subbab 3.3 (Rencana Implementasi). Pengujian akan dibagi menjadi dua kategori, fungsional dan non-fungsional.

### 3.4.1 Pengujian Fungsional

Pengujian fungsional memastikan setiap modul aplikasi dapat berjalan sesuai dengan kebutuhan proses bisnis Bank Air Kami. Berikut merupakan matriks *test-case* yang telah dibuat:

Tabel 3.3 Daftar *Test-case* untuk Pengujian Fungsional

ID	Modul / Fitur	Aksi yang Diuji	Langkah Uji	Hasil Diharapkan
TC-US-01	Pengguna	Tambah pengguna	Buka menu Pengguna → Klik “Tambah” → Isi form valid → Simpan	Baris baru muncul di tabel Pengguna; toast “Berhasil disimpan”
TC-US-02	Pengguna	Hapus pengguna	Pilih baris → Klik ikon hapus → Konfirmasi	Baris hilang; toast “Berhasil dihapus”
TC-US-03	Pengguna	Ubah pengguna	Pilih baris → Klik edit → Perbarui data → Simpan	Data ter-update; toast “Berhasil diperbarui”
TC-US-04	Pengguna	Tampil daftar	Buka menu Pengguna	Tabel menampilkan semua pengguna aktif
TC-FA-01	Filter Air	Tambah filter	Menu Filter Air → Tambah → Isi form → Simpan	Filter tercatat & tampil di tabel
TC-FA-02	Filter Air	Hapus filter	Pilih baris → Hapus → Konfirmasi	Baris filter hilang
TC-FA-03	Filter Air	Ubah filter	Pilih → Edit → Simpan	Kolom berubah sesuai input
TC-FA-04	Filter Air	Tampil daftar	Buka menu Filter Air	Seluruh filter ter-render
TC-SEN-01	Sensor	Tambah sensor	Menu Sensor → Tambah → Isi form → Simpan	Sensor aktif & muncul di tabel
TC-SEN-02	Sensor	Hapus sensor	Pilih baris → Hapus → Konfirmasi	Sensor tidak tercantum lagi
TC-SEN-03	Sensor	Ubah sensor	Pilih → Edit → Simpan	Data sensor ter-update
TC-SEN-04	Sensor	Tampil daftar	Buka menu Sensor	Semua sensor ter-render
TC-FPU-01	Filter Air Pengguna	Tambah	Menu Filter Pengguna → Tambah → Simpan	Relasi filter-user tersimpan
TC-FPU-02	Filter Air Pengguna	Hapus	Pilih → Hapus → Konfirmasi	Relasi hilang dari tabel
TC-FPU-03	Filter Air Pengguna	Ubah	Edit → Simpan	Relasi diperbarui
TC-FPU-04	Filter Air Pengguna	Tampil	Buka menu	Daftar relasi tampil lengkap
TC-STK-01	Stok/Persediaan	Tambah stok	Menu Stok → Tambah → Simpan	Item stok baru tercatat
TC-STK-02	Stok/Persediaan	Hapus stok	Pilih → Hapus → Konfirmasi	Item hilang
TC-STK-03	Stok/Persediaan	Ubah stok	Edit → Simpan	Jumlah stok ter-update
TC-STK-04	Stok/Persediaan	Tampil stok	Buka menu Stok	Tabel menampilkan stok terkini

TC-QUAL-01	Riwayat Kualitas Air	Tambah rekaman	Sistem menerima paket sensor → Simpan otomatis	Rekaman muncul di tabel riwayat
TC-QUAL-02	Riwayat Kualitas Air	Tampil riwayat	Buka menu Riwayat Kualitas	Riwayat 24 jam ter-render
TC-RF-01	Riwayat Penggantian Filter	Tambah	Menu Riwayat Filter → Tambah → Simpan	Catatan penggantian tercatat
TC-RF-02	Riwayat Penggantian Filter	Hapus	Pilih catatan → Hapus	Catatan hilang
TC-RF-03	Riwayat Penggantian Filter	Ubah	Edit → Simpan	Catatan ter-update
TC-RF-04	Riwayat Penggantian Filter	Tampil	Buka menu	Seluruh catatan tampil
TC-DH-04	Dasbor	Tampil	Buka menu	Seluruh grafik dan GIS tampil

Pengujian ini dinyatakan lulus bila *code coverage* melebihi 80 %. Seluruh skenario dieksekusi sebagai testing di pipeline GitHub Actions.

### 3.4.2 Pengujian Non-Fungsional

Pengujian non-fungsional difokuskan pada antarmuka responsif dan manajemen peran (RBAC), dua aspek yang paling memengaruhi pengalaman dan keamanan pengguna.

Tabel 3.4 Daftar Pengujian Non-Fungsional

Aspek Pengujian	Aspek	Metode dan Instrumen	Indikator Keberhasilan
Responsivitas Antarmuka	Tampilan mobile, tablet, dan desktop	Pengujian dilakukan menggunakan BrowserStack dan DevTools Responsive Mode pada lebar layar $\leq 480$ px, 481–768 px, dan $\geq 1024$ px.	Antarmuka dinyatakan lulus apabila tata letak tetap proporsional, tidak terpotong, tidak saling tumpang tindih, serta seluruh fungsi tetap dapat diakses pada setiap ukuran tampilan.
Manajemen Hak Akses Berbasis Peran (RBAC)	Admin	Pengujian dilakukan menggunakan Laravel <i>Feature Test</i> (Pest/PHPUnit) dengan mensimulasikan peran	Sistem dinyatakan lulus apabila Admin memperoleh respons HTTP 200 dan dapat mengakses seluruh menu serta fitur sesuai

		Admin.	kewenangannya.
	Pengguna	Pengujian dilakukan menggunakan Laravel menggunakan <i>Feature Test</i> (Pest/PHPUnit) dengan mensimulasikan peran Pengguna.	Sistem dinyatakan lulus apabila Pengguna hanya dapat mengakses data miliknya sendiri pada dasbor, riwayat penggantian filter, riwayat kualitas air, dan data filter air. Ekspor CSV/PDF hanya diperbolehkan untuk data milik sendiri, sedangkan akses ke data atau endpoint di luar kewenangannya harus menghasilkan respons HTTP 403.

### 3.5 Evaluasi dan Dokumentasi

Evaluasi dan dokumentasi pada penelitian ini disusun sebagai artefak yang dapat digunakan sebagai acuan bagi penelitian atau pengembangan berikutnya guna mendukung keberlanjutan proyek. Dokumentasi yang disediakan meliputi ERD, tautan source code, daftar fitur yang diimplementasikan, serta ringkasan hasil unit testing.

## BAB IV

### HASIL DAN PEMBAHASAN

Bab ini akan menyajikan hasil dan pembahasan dari pengembangan sistem Bank Air Kami yang berupa dasbor pemantauan kualitas air dan manajemen filter air yang responsif. Fokus utama adalah menunjukkan capaian implementasi berdasarkan rancangan pada Bab III, sekaligus menilai sejauh mana sistem memenuhi tujuan penelitian. Bab IV menyajikan hasil implementasi modul pemantauan kualitas air, GIS, manajemen filter, dan RBAC, serta ringkasan pengujian fungsional dan non-fungsional. Rincian implementasi dan hasil uji dirangkum.

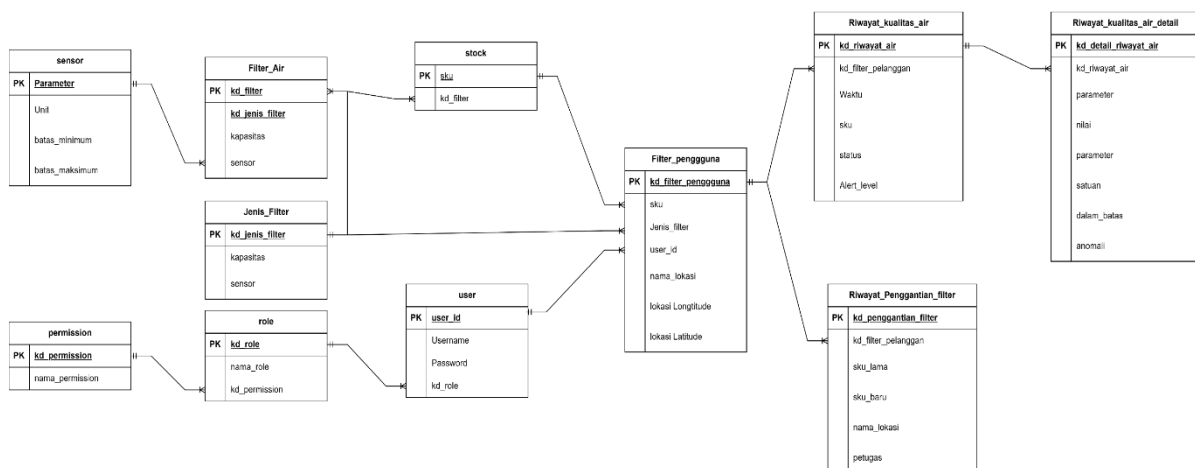
#### 4.1 Hasil Implementasi Sistem

Subbab ini memaparkan hasil implementasi dasbor pemantauan kualitas air dan manajemen filter air yang responsif. Implementasi difokuskan management dan monitoring kualitas air serta menggunakan simulasi IoT.

##### 4.1.1 Implementasi Database

Implementasi basis data pada sistem Bank Air Kami dibagi menjadi dua lapisan: (1) MySQL sebagai basis data operasional aplikasi *web*; (2) Firebase Realtime Database sebagai penyimpan data sensor IoT secara *realtime*.

#### Basis Data Operasional



### Gambar 4.1 ERD Implementasi

Struktur basis data MySQL digambarkan pada ERD di Gambar 4.1. Secara garis besar, entitas yang digunakan dan perannya adalah:

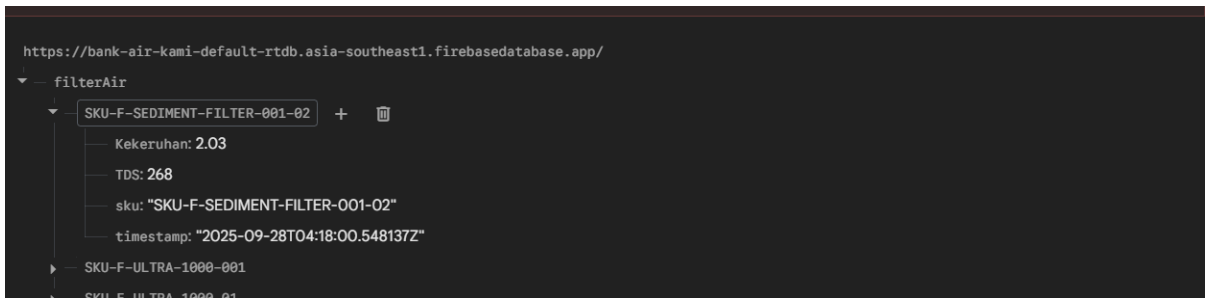
- a. user: menyimpan akun admin dan Pengguna yang mengakses dasbor.
- b. role dan permission: mendukung mekanisme Role Based Access Control (RBAC).
- c. jenis\_filters: daftar jenis filter air (misal: sediment, karbon aktif, ultra).
- d. filters: master produk filter berdasarkan kode/sku dan kapasitas.
- e. stocks: stok/persediaan filter di sisi admin berdasarkan jenis dan kode filter.
- f. filterPengguna: relasi antara pengguna dan filter yang terpasang, termasuk lokasi, koordinat, status perangkat, dan jadwal pemeliharaan.
- g. sensors: konfigurasi setiap parameter sensor (pH, kekeruhan, TDS, suhu), satuan, dan batas minimum–maksimum.
- h. Riwayat\_kualitas\_air: menyimpan ringkasan nilai sensor untuk satu filter pada waktu tertentu beserta klasifikasi level kualitas air
- i. Riwayat\_Penggantian\_filter: mencatat setiap peristiwa penggantian filter (filter lama, filter pengganti, dan waktu penggantian).

Seluruh tabel menggunakan kunci primer bertipe bilangan bulat (*auto increment*) dan relasi antar entitas diatur dengan *foreign key*. Skema ini diimplementasikan melalui berkas *migration* Laravel sehingga dapat direplikasi secara konsisten di lingkungan pengembangan, pengujian, dan produksi.

Perubahan terhadap ERD awal terutama berupa penyederhanaan entitas dan penyesuaian penamaan. Entitas Jenis\_Filter pada rancangan awal digabung ke dalam Filter\_Air sehingga informasi jenis dan spesifikasi filter cukup dikelola pada satu tabel. Tabel stocks diganti menjadi filter\_item untuk menegaskan bahwa tabel ini merepresentasikan item fisik filter di gudang, sedangkan Filter\_pengguna disesuaikan menjadi Filter\_Air\_pelanggan agar lebih jelas menggambarkan pemasangan filter pada pelanggan. Selain itu, tabel Riwayat\_kualitas\_air\_detail yang sebelumnya dirancang dihapus karena kebutuhan pelaporan sudah dapat dipenuhi melalui ringkasan pada Riwayat\_kualitas\_air. Beberapa atribut non-esensial juga disederhanakan agar formulir input lebih ringkas dan lebih sesuai dengan kebutuhan MVP.

### Basis Data Sensor Realtime

Firestore Realtime Database digunakan untuk menyimpan data mentah sensor secara *streaming* dari perangkat IoT. Struktur data utama ditunjukkan pada Gambar 4.2 dengan pola:



Gambar 4.2 Struktur *Database* Dokumen di Firebase

Setiap node SKU filter air menyimpan nilai terakhir untuk parameter Kekeruhan, Suhu, TDS, pH, kode sku, serta timestamp. Perangkat IoT melakukan *push* data ke node sesuai SKU filter, sedangkan aplikasi *web* mengambil data tersebut untuk ditampilkan dalam bentuk grafik dan kemudian merangkum sebagian nilainya menjadi rekaman di tabel Riwayat\_kualitas\_air pada MySQL.

#### 4.1.2 Implementasi Aplikasi Website

Dari sisi aplikasi *web*, implementasi masih sesuai dengan perencanaan. Aplikasi dibangun menggunakan Laravel dengan Laravel Filament sebagai fondasi utama baik untuk *backend* maupun *frontend*. Pemilihan *stack* ini mempermudah pembuatan *admin panel*, *form* data, serta komponen tabel yang konsisten dengan rancangan pada Bab III.

Secara umum, hampir seluruh fitur utama yang didefinisikan MVP telah terealisasi. Beberapa penyesuaian dilakukan dalam bentuk perubahan tampilan, penyederhanaan alur, serta penambahan atau pengurangan fitur pendukung agar sistem lebih fokus pada kebutuhan operasional Bank Air Kami. Ringkasan perbandingan antara perencanaan dan hasil implementasi ditunjukkan pada Tabel 4.1.

Tabel 4.1 Daftar Fitur yang Terimplementasi

Fitur	Perencanaan	Implementasi	Perubahan
Dasbor pemantauan kualitas air	Dasbor menampilkan ringkasan kualitas air dan status perangkat per filter.	Dasbor menampilkan kartu ringkasan (jumlah filter aktif/offline, kategori kualitas air), grafik tren	Beberapa grafik detail dihilangkan agar tampilan lebih ringkas

Manajemen master filter & stok	Dasbor menampilkan ringkasan kualitas air dan status perangkat per filter.	Modul Filter_Air dan filter_item terpisah	Struktur dipisah agar lebih jelas membedakan master data dan item fisik.
Manajemen pengguna	Modul untuk mengelola jenis filter, master filter, dan stok dalam satu halaman.	Modul user (pengguna) dan Filter Air pengguna untuk pemasangan filter, lengkap dengan alamat, koordinat, dan status filter.	Fitur detail kunjungan teknisi belum diimplementasikan; diganti catatan singkat dan status pemeliharaan.
Manajemen sensor dan parameter	Modul pelanggan dan relasi filter-pengguna (satu pelanggan bisa memiliki beberapa filter).	Resource sensor dan pengaitan ke filter melalui relasi di backend .	Fitur kalibrasi otomatis belum diimplementasi, digantikan dengan pengaturan parameter secara manual.
Riwayat kualitas air	CRUD sensor, ambang batas, dan pengaitan sensor ke tiap filter.	Halaman Riwayat kualitas air menampilkan data ringkasan per filter dan rentang waktu, dengan grafik sederhana per filter.	Secara tampilan mengalami perubahan. Secara fitur tidak memiliki perubahan yang signifikan.
Peta/GIS lokasi pemasangan filter	Peta interaktif menampilkan posisi seluruh filter pengguna.	Halaman peta yang menampilkan marker lokasi pemasangan filter berdasarkan koordinat dari Filter_Air_pelanggan	Peta masih pada level marker sederhana.
Integrasi/ simulasi IoT	Menggunakan perangkat IoT untuk mengirim data sensor langsung ke server.	Integrasi dilakukan melalui Firebase Realtime Database. Untuk pengujian, digunakan simulasi pengiriman data dummy dari skrip ke node SKU.	Penggunaan perangkat IoT fisik masih terbatas pada tahap prototipe; simulasi diprioritaskan untuk memastikan alur data dan dasbor.

Perubahan-perubahan tersebut terjadi terutama karena penyesuaian terhadap waktu pengembangan, kemudahan penggunaan bagi admin, serta kebutuhan untuk menjaga sistem tetap stabil pada rilis awal. Fitur-fitur yang belum terimplementasi sepenuhnya tetap dicatat sebagai peluang pengembangan lanjutan, tanpa mengganggu fungsi utama sistem dalam memantau kualitas air dan mengelola filter.

Untuk menguji alur data dari perangkat ke dasbor tanpa harus selalu bergantung pada perangkat IoT fisik, dilakukan simulasi IoT. Simulasi ini memanfaatkan laravel schedule untuk

mengirimkan data sensor dummy (kekeruhan, suhu, TDS, pH, timestamp) ke Firebase Realtime Database pada node filterAir/{SKU}. Melalui pendekatan ini, penulis dapat:

- Memastikan struktur data Firebase sudah sesuai dengan kebutuhan aplikasi *web*.
- Menguji tampilan grafik dan tabel riwayat kualitas air di dasbor.
- Melakukan pengujian fungsional tanpa harus menunggu data dari sensor fisik secara terus-menerus.

Simulasi ini menjadi jembatan antara rancangan integrasi IoT dengan kondisi perangkat lapangan yang masih dalam tahap iterasi.

Untuk membantu pembaca memahami alur penggunaan aplikasi berdasarkan tampilan antarmuka, disusun *Visual Table of Content* (VTOC) yang ditunjukkan pada Gambar 4.3. VTOC ini berupa diagram yang merangkum halaman-halaman utama aplikasi (misalnya Dasbor, Manajemen Filter, Manajemen Pelanggan, Riwayat Kualitas Air, Riwayat Penggantian Filter, dan Peta Lokasi) beserta hubungan navigasinya.



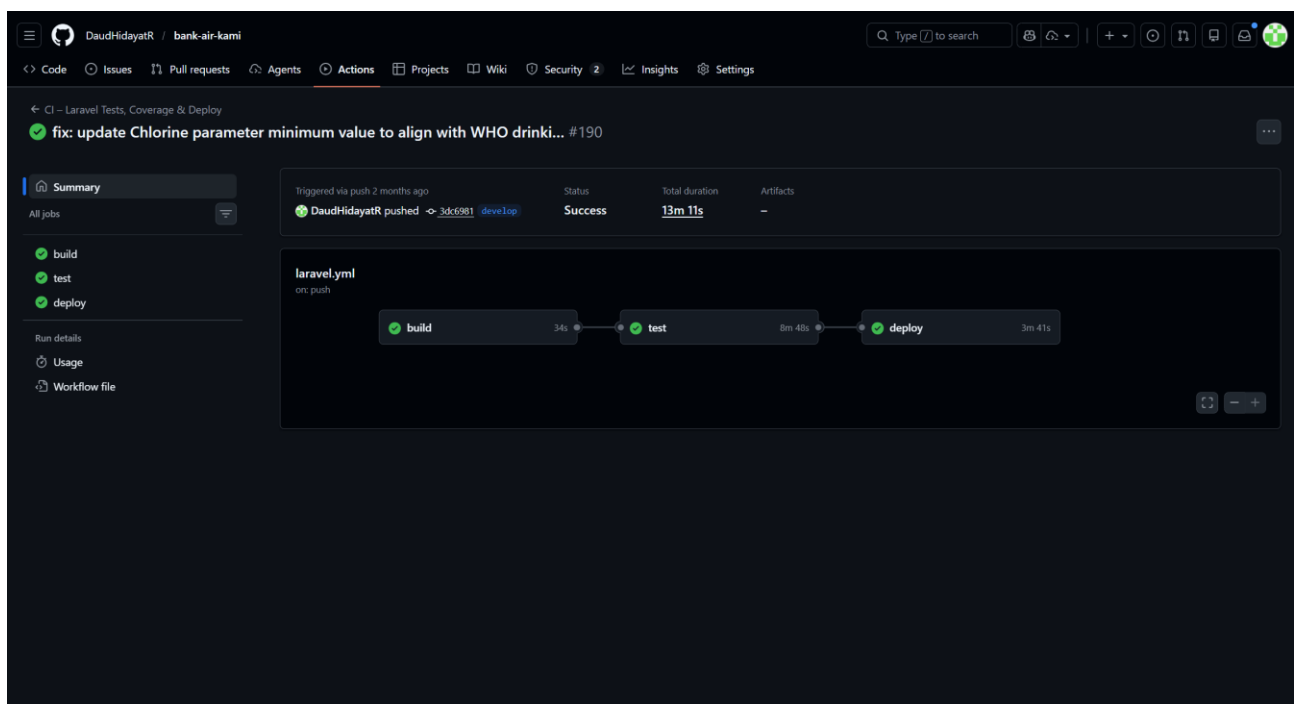
Gambar 4.3 *Visual Table of Content* (VTOC) Aplikasi Hasil Implementasi

Dengan adanya VTOC, pembaca non-teknis dapat melihat “peta” aplikasi secara menyeluruh sebelum melihat contoh tampilan yang lebih detail. Hal ini juga memudahkan penguji untuk menghubungkan setiap fitur yang dijelaskan dalam teks dengan halaman dan komponen nyata di aplikasi.

### 4.1.3 Implementasi Infrastruktur & Automasi

Secara umum, *tech stack* yang digunakan pada tahap implementasi tetap sama dengan perencanaan pada Bab III. Aplikasi dibangun dengan Laravel dan Laravel Sail/Filament, dijalankan di atas kontainer Docker, dengan GitHub Actions sebagai alat utama untuk automasi *testing* dan *deployment*. Google Cloud digunakan sebagai infrastruktur server produksi, sedangkan Cloudflare ditambahkan sebagai pengelola DNS dan gerbang akses aplikasi sehingga pengelolaan nama domain dan routing trafik menjadi lebih terpusat.

Pipeline kerja sistem secara garis besar mengikuti tiga tahap: instalasi lingkungan, pengujian otomatis, dan deployment ke server produksi.



Gambar 4.4 Hasil Implementasi Pipeline Github Action

### Tahap Instalasi Lingkungan

Sebelum aplikasi dijalankan, dilakukan beberapa langkah persiapan pada server berbasis Ubuntu/Debian:

- a. Instalasi tool pendukung server terlebih dahulu dipasang runtime dan tool yang dibutuhkan, yaitu Node.js, npm, Docker Engine, PHP, MySQL client, dan Composer. Untuk mempermudah, disediakan shell script sehingga proses instalasi dapat dijalankan secara otomatis.

- b. Konfigurasi environment Berkas `.env` dibuat untuk menentukan parameter lingkungan (*database*, *Firebase*, dan sebagainya) yang disesuaikan dengan kebutuhan *development*, *testing*, atau *production*.
- c. Instalasi dependensi aplikasi dependensi backend dipasang menggunakan `composer install`, sedangkan dependensi frontend menggunakan perintah `npm`.
- d. Menjalankan container Karena proses pengembangan, pengujian, dan deployment menggunakan kontainer, aplikasi dijalankan dengan membangun image dan menyalakan kontainer menggunakan `docker compose up -d` atau `sail up -d`.
- e. Migrasi basis data Setelah kontainer aktif, skema basis data dimigrasikan menggunakan *Laravel ORM*, misalnya melalui perintah `sail artisan migrate:fresh --seed`. Setelah langkah ini, aplikasi siap digunakan.

### **Tahap Pengujian Otomatis**

Tahap pengujian otomatis diatur melalui *GitHub Actions*. Setiap ada perubahan kode yang dikirim (*push*) ke repositori, pipeline akan:

- a. Menjalankan seluruh berkas pengujian yang berada di folder `tests` (*unit test* dan *functional/feature test*).
- b. Menjalankan perintah pengujian *Laravel* dengan target coverage minimal 80% dan mode paralel untuk mempercepat eksekusi, misalnya melalui `sail artisan test --coverage`
- c. `--parallel --processes=8`.
- d. Jika salah satu pengujian gagal atau coverage tidak memenuhi batas minimal, pipeline dinyatakan gagal dan proses berhenti pada tahap ini sehingga deployment tidak akan dijalankan.

### **Tahap Deployment ke Produksi**

Deployment hanya dijalankan ketika seluruh pengujian pada *GitHub Actions* berhasil. Alur umumnya adalah:

- a. Trigger dari branch tertentu Kode yang sudah lulus pengujian dan dikirim ke branch *staging* atau branch produksi akan memicu job deployment.
- b. *GitHub Actions* melakukan remote ke server *Google Cloud* menggunakan *SSH* dengan kredensial khusus.
- c. Sinkronisasi kode dan kontainer Di server, repositori aplikasi diperbarui dengan menjalankan `git pull` dari branch yang bersangkutan. Setelah itu kontainer

diperbarui/dinyalakan ulang dengan docker compose up -d atau sail up -d sehingga versi aplikasi di produksi mengikuti kode terbaru.

- d. Pengelolaan DNS melalui Cloudflare Aplikasi yang sudah terdeploy diakses menggunakan domain yang dikelola oleh Cloudflare. Layanan ini menangani manajemen DNS dan pengaturan rute trafik ke server Google Cloud, sehingga perubahan alamat IP atau penambahan subdomain dapat dilakukan tanpa mengubah konfigurasi di sisi server aplikasi.

## 4.2 Hasil Pengujian Aplikasi

Bagian ini menjelaskan hasil pengujian yang dilakukan terhadap aplikasi Bank Air Kami. Pengujian dilakukan sesuai rencana yang telah disusun pada Bab III, khususnya Subbab 3.4 tentang pengujian fungsional dan non-fungsional. Seluruh pengujian dijalankan pada environment yang sama dengan lingkungan implementasi serta terintegrasi dengan pipeline GitHub Actions. Kriteria keberhasilan pengujian ditetapkan sebagai berikut:

- a. Seluruh *test case* penting pada matriks pengujian fungsional berstatus lulus (Pass).
- b. Seluruh unit/*feature* test otomatis berstatus lulus tanpa ada kegagalan.
- c. Code coverage baris kode aplikasi minimal 80%.

Pada pengujian ini diperoleh 772 test (unit + feature) dieksekusi dan seluruhnya lulus dan Code coverage baris sebesar  $\pm 88,54\%$ . Ringkasan jumlah dan jenis pengujian ditunjukkan pada Tabel 4.2.

Tabel 4.2 Daftar Ringkasan Test Secara Umum

Kategori	Pendekatan	Jumlah / Cakupan	Status
Unit & <i>Feature</i> Test	Pest/PHPUnit	772 <i>test</i> , 3034 <i>assertion</i>	Lulus
Functional Test per modul	Black-box sesuai matriks TC	54 <i>test case</i> (52 otomatis, 2 manual)	Lulus
Responsif	inspeksi tampilan	Mobile, tablet, desktop	Lulus
RBAC	<i>Feature</i> test dan simulasi role	Skenario admin vs Pelanggan	Lulus

### 4.2.1 Pengujian Fungsional

Pada tahap implementasi, matriks pengujian yang telah disusun sebelumnya diperdalam dan diperluas menjadi 54 *test case* fungsional yang lebih rinci untuk merepresentasikan kebutuhan aktual sistem. Dalam proses pematangan tersebut, sebagian *test case* mengalami penyesuaian nama modul (misalnya “Filter Air Pengguna” menjadi “Filter Pengguna”),

penggabungan fitur (misalnya riwayat penggantian filter diintegrasikan sebagai bagian dari monitoring perangkat dan riwayat kualitas air), serta penambahan modul baru yang muncul sebagai kebutuhan implementasi, seperti modul Role/Hak Akses untuk mendukung penerapan RBAC dan modul Simulasi & Commands berupa perintah Artisan untuk simulasi data sensor serta sinkronisasi Firebase–MySQL.

Pelaksanaan pengujian fungsional dilakukan melalui dua mekanisme, yaitu pengujian otomatis yang diimplementasikan sebagai *feature test* menggunakan Pest/PHPUnit, dijalankan dengan perintah `sail test --parallel` dan terintegrasi ke GitHub Actions, mencakup skenario CRUD, filtrasi data, integrasi antar modul, hingga skenario anomali kualitas air; serta pengujian manual yang digunakan untuk skenario UI/UX dan integrasi visual, seperti integrasi Google Maps pada form Filter Pengguna dan visualisasi widget sensor pada dasbor, dengan hasil dicatat pada tabel *test case* berstatus “Lulus”. Ringkasan jumlah *test case* fungsional per modul disajikan pada Tabel 4.3.

Tabel 4.3 Daftar Ringkasan *Test* Secara Fungsional

Modul / Fitur	Kode TC (Rentang)	Jumlah TC	Metode Eksekusi	Status
Pengguna	TC-US-01 – 06	6	Otomatis	Lulus
Jenis Filter	TC-JF-01 – 04	4	Otomatis	Lulus
Filter Air	TC-FA-01 – 05	5	Otomatis	Lulus
Sensor	TC-SEN-01 – 06	6	Otomatis	Lulus
Stok / Persediaan	TC-STK-01 – 04	4	Otomatis	Lulus
Filter Pengguna	TC-FPU-01 – 08	8	7 Otomatis dan 1 manual	Lulus
Riwayat Kualitas Air	TC-QUAL-01 – 06	6	Otomatis	Lulus
Dasbor	TC-DH-01 – 09	9	8 Otomatis dan 1 manual	Lulus
Simulasi & Commands	TC-SIM-01 – 10	10	Otomatis	Lulus
Role / Hak Akses	TC-RL-01 – 07	7	Otomatis	Lulus
Total		54	52 Otomatis dan 2 manual	Lulus

Seluruh operasi CRUD utama (tambah, ubah, hapus, tampil) pada modul Pengguna, Jenis Filter, Filter Air, Sensor, Stok, Filter Pengguna, Riwayat Kualitas Air, Role berjalan sesuai harapan. Rincian langkah uji per *test case* disajikan pada Lampiran.

### 4.2.2 Pengujian Non-Fungsional

Dari pelaksanaan pengujian non-fungsional diperoleh hasil bahwa komponen utama Filament, seperti tabel data, formulir, dan modal, mampu menyesuaikan lebar layar secara responsif tanpa mengalami pemotongan tampilan. Pada perangkat mobile, kolom tabel yang dinilai kurang penting secara otomatis disembunyikan atau ditampilkan melalui mekanisme horizontal scroll oleh Filament, sehingga informasi utama tetap terbaca dengan jelas. Selain itu, formulir input pada berbagai modul seperti Pengguna, Filter Air, Filter Pengguna, Sensor, dan modul lainnya tetap dapat diisi dengan nyaman pada lebar layar kecil. Pengujian juga menunjukkan bahwa peta Google Maps pada form Filter Pengguna masih dapat dioperasikan dengan baik di layar mobile, termasuk fungsi drag, zoom, dan pemilihan lokasi, tanpa kendala fungsional. Ringkasan hasil pengujian non-fungsional tersebut disajikan pada Tabel 4.4.

Tabel 4.4 Daftar Ringkasan *Test* Secara Non-Fungsional

Aspek	Metode / Alat	Skenario Utama	Hasil Pengujian
Responsif UI	<i>DevTools Responsive Mode</i> , inspeksi manual	Mobile, tablet, desktop	Lulus, tidak ada layout terpotong / overlap
RBAC super_admin	<i>Feature test</i>	Akses semua modul admin	Lulus, seluruh menu yang diharapkan dapat diakses
RBAC pengguna	<i>Feature test</i>	Akses dasbor dan data milik sendiri	Lulus, akses tidak sah mendapat 403 dan menu disembunyikan

## 4.3 Pembahasan

Bab ini membahas keterkaitan antara hasil implementasi dan pengujian dasbor dengan tujuan penelitian yang telah dirumuskan. Pembahasan juga memposisikan sistem yang dikembangkan terhadap penelitian-penelitian sebelumnya pada Bank Air Kami. Fokus pembahasan meliputi capaian fungsional sistem, evaluasi keterbatasan selama penelitian, serta bentuk pengembangan proses bisnis yang diusulkan, khususnya pada pengelolaan filter air.

### 4.3.1 Dasbor Responsif Untuk Pemantauan Kualitas Air

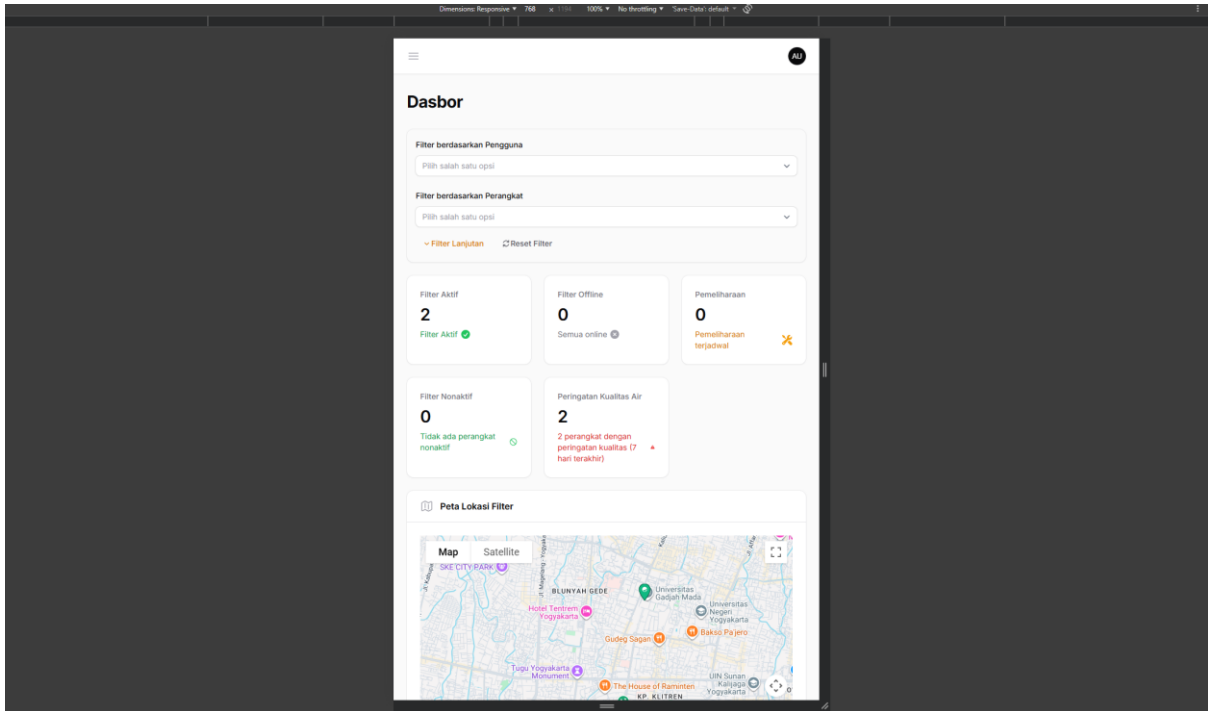
Dasbor monitoring dirancang untuk mendukung pemantauan kualitas air secara cepat pada berbagai perangkat. Pada tampilan dasbor (Gambar 4.2.1), kontrol penyaringan data ditempatkan pada bagian atas halaman untuk memudahkan pengguna menentukan konteks pemantauan, seperti pengguna, perangkat, status, rentang tanggal, serta periode cepat (misalnya 1 jam terakhir hingga 30 hari terakhir). Pendekatan ini selaras dengan pedoman desain

dashboard yang menekankan pentingnya kemampuan bernavigasi secara cepat antar representasi data agar pengguna dapat memahami kondisi terkini dalam konteks tren historis maupun statistik.

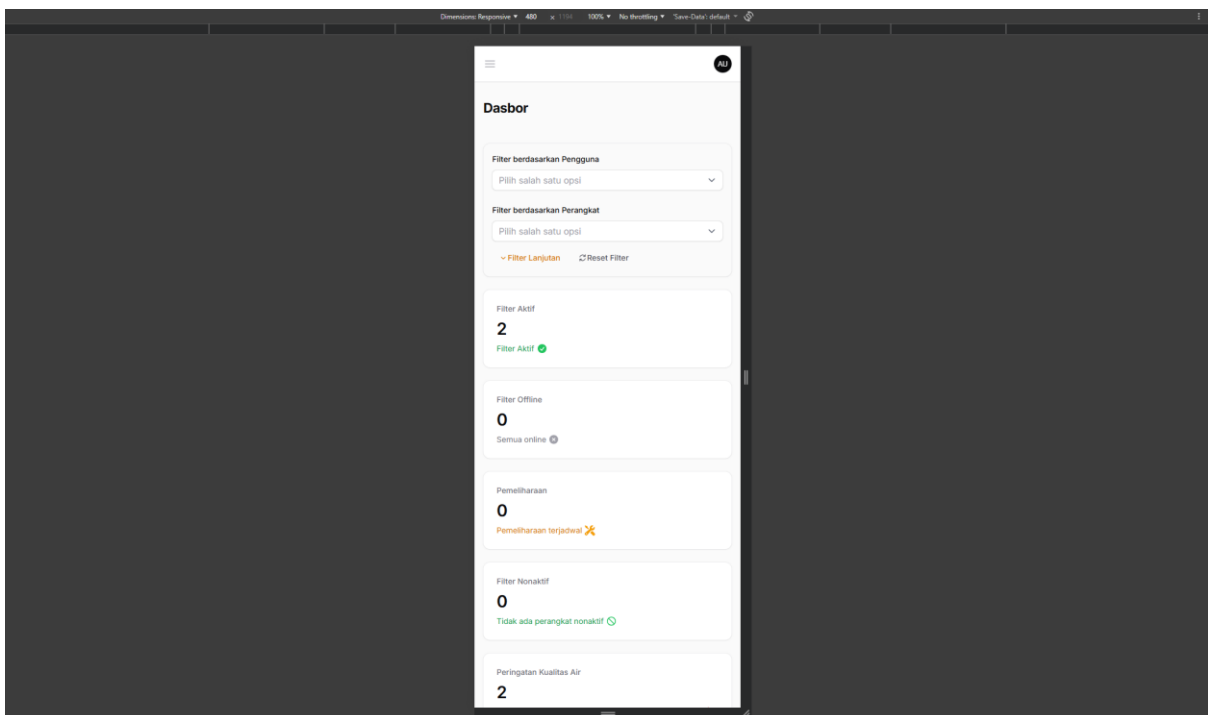
Setelah pengguna menentukan konteks data, sistem menampilkan ringkasan kondisi melalui kartu informasi, seperti jumlah filter aktif, filter *offline*, pemeliharaan, filter nonaktif, dan peringatan kualitas air. Ringkasan ini memberikan gambaran awal (*overview*) sebelum pengguna meninjau detail data, sehingga dasbor mampu menyajikan kondisi sistem secara komprehensif untuk mendukung aktivitas monitoring dan pengambilan keputusan. Selanjutnya, peta lokasi filter digunakan untuk menampilkan sebaran titik perangkat/filter yang dipantau, sehingga pengguna dapat memahami konteks spasial ketika terjadi peringatan atau ketidakwajaran data. EPA menjelaskan bahwa penyajian geospasial membantu pengguna melihat informasi dalam konteks ruang (GIS) serta menggunakan konsep layer (basemap dan layer informasi) untuk menampilkan data tambahan di atas peta dasar.

Detail perubahan parameter kualitas air ditampilkan menggunakan grafik garis (*time-series*) untuk setiap parameter (klorin, laju air, kekeruhan, pH, suhu, dan TDS). Pemilihan *time-series* plot dilakukan karena data sensor bersifat deret waktu dan visualisasi ini efektif untuk memahami tren pada berbagai skala waktu. Skala waktu dapat divariasikan untuk mengamati perubahan dramatis pada periode pendek maupun tren gradual pada periode yang lebih panjang (hari/minggu/bulan). Hal ini konsisten dengan fitur periode cepat pada sistem (1 jam, 1 hari, 7 hari, 30 hari) yang mendukung pemantauan rutin maupun investigasi anomali. Selain itu, penanda batas minimum–maksimum dan rentang aman membantu pengguna menilai apakah kondisi masih berada pada ambang yang ditetapkan.

Implementasi dasbor dibangun menggunakan Laravel Filament yang menyediakan komponen antarmuka berbasis widget serta layout grid responsif, sehingga tata letak kartu, peta, dan grafik dapat menyesuaikan ukuran layar. Berdasarkan skenario pengujian responsivitas yang telah ditetapkan pada Bab III, implementasi antarmuka diuji pada tiga rentang ukuran layar, yaitu  $\leq 480$  px (*mobile*), 481–768 px (*tablet*), dan  $\geq 1024$  px (*desktop*). Hasil pengujian menunjukkan bahwa tampilan dasbor dapat menyesuaikan ukuran layar dengan baik tanpa ditemukan permasalahan tata letak yang mengganggu. Bukti pengujian responsivitas ditunjukkan pada Gambar 4.5 dan 4.6.

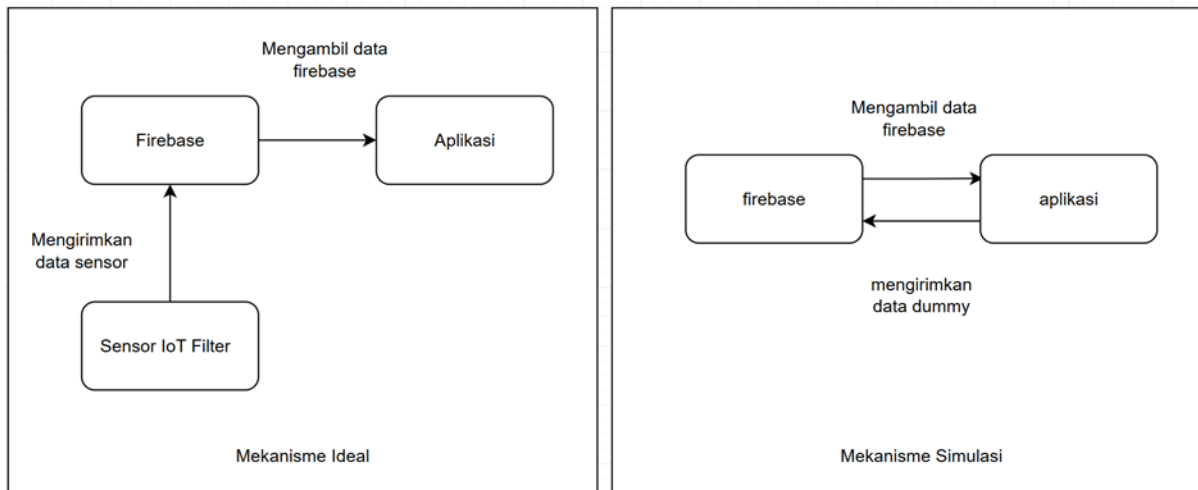


Gambar 4.5 Dasbor ukuran 768 px (*tablet*)



Gambar 4.6 Dasbor ukuran 480 px (*mobile*)

### 4.3.2 Mekanisme Simulasi Data IoT



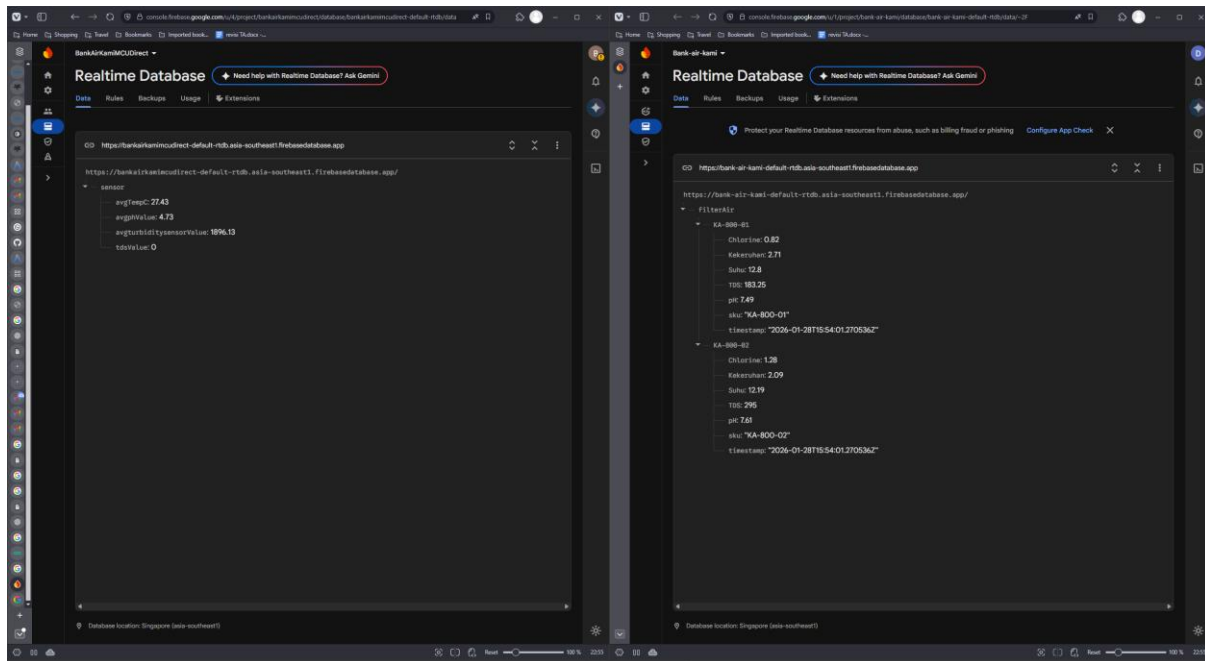
Gambar 4.7 Perbandingan Mekanisme Ideal (kiri) dan Mekanisme Simulasi (kanan)

Simulasi data sensor IoT pada sistem dilakukan melalui mekanisme data ingestion dengan perantara Firebase sebagai penyimpanan data sementara. Pada mekanisme ideal, perangkat IoT melakukan pengukuran parameter kualitas air secara berkala, kemudian mengirimkan data hasil pengukuran ke Firebase. Pada penelitian ini, peran perangkat IoT tersebut digantikan oleh modul simulasi yang menghasilkan data parameter kualitas air secara periodik dan mengirimkannya ke Firebase. Selanjutnya, aplikasi website mengambil data yang tersimpan di Firebase secara berkala menggunakan scheduler pada Laravel untuk memastikan data dapat direplikasi ke basis data aplikasi. Data yang berhasil diambil kemudian disimpan ke dalam *database* aplikasi dalam bentuk riwayat kualitas air dan detail riwayat kualitas air, sehingga dapat diolah lebih lanjut untuk kebutuhan visualisasi grafik, rekap historis, serta pelacakan kondisi kualitas air dari waktu ke waktu.

Pada penelitian ini, mekanisme simulasi tetap mengikuti alur tersebut, dengan penekanan bahwa proses pengambilan data dilakukan secara periodik melalui Laravel Scheduler. Proses ini mencakup pembacaan data terbaru dari Firebase, pemetaan atribut data (misalnya waktu pencatatan, identitas perangkat/filter, dan nilai parameter), serta penyimpanan ke skema relasional pada *database* aplikasi. Dengan pemindahan data ke *database* aplikasi, sistem dapat menyediakan fitur pencarian, penyaringan berdasarkan periode, serta penyusunan riwayat yang lebih terstruktur untuk ditampilkan pada dasbor.

Karena keterbatasan ketersediaan perangkat sensor secara penuh selama penelitian, data pada Firebase tidak berasal langsung dari perangkat fisik, melainkan menggunakan data simulasi (dummy) yang dibuat secara berkala dan dikirim ke Firebase. Perbedaan utama dibanding mekanisme ideal terletak pada sumber data: pada kondisi ideal data berasal dari

perangkat IoT, sedangkan pada penelitian ini data berasal dari modul simulasi. Meskipun demikian, alur setelah data berada di Firebase tetap sama, sehingga mekanisme pengambilan data berkala oleh website hingga penyimpanan dan visualisasi pada dasbor tetap dapat divalidasi.



Gambar 4.8 Perbandingan Struktur *Database* Penelitian Sebelumnya (kiri) dan Sekarang (kanan)

Pada penelitian sebelumnya, integrasi data dirancang untuk mengakomodasi satu set data filter, sehingga struktur penyimpanan pada Firebase dan pengolahan data di sisi aplikasi berfokus pada satu titik pemantauan. Pada penelitian ini, struktur integrasi dikembangkan agar dapat mendukung lebih dari satu filter. Perubahan ini dilakukan dengan menambahkan identitas filter pada struktur data Firebase, sehingga data riwayat kualitas air dapat dikelompokkan per filter dan dikelola secara paralel. Dengan demikian, sistem yang dikembangkan lebih adaptif untuk kondisi operasional yang melibatkan beberapa titik filter dan memungkinkan perluasan proses bisnis pengelolaan filter.

### 4.3.3 Perluasan Sistem Untuk Proses Bisnis Pengelolaan Filter

Subbab ini membahas perluasan sistem yang dikembangkan agar tidak hanya berfungsi sebagai media pemantauan kualitas air, tetapi juga mampu mendukung proses bisnis usulan

pengelolaan filter air. Pada penelitian ini, proses bisnis usulan diposisikan sebagai rancangan alur operasional target yang dijadikan acuan dalam pengembangan sistem, sedangkan Bank Air

Kami digunakan sebagai konteks/studi kasus penerapan agar kebutuhan monitoring kualitas air dan pengelolaan operasional tetap relevan. Dalam konteks BPMN yang digunakan, aktor utama yang terlibat pada proses bisnis usulan meliputi operator/admin, pengguna filter air, dan perangkat IoT sebagai sumber data pengukuran.

Perluasan yang dimaksud pada penelitian ini adalah perubahan cakupan sistem dari yang sebelumnya berfokus pada keterbukaan informasi kualitas air (*monitoring*) menjadi sistem yang juga mendukung aktivitas operasional pengelolaan filter (*management*). Pada sistem sebelumnya, fokus utama berada pada penyajian informasi kualitas air secara lebih terbuka sebagai bagian dari konteks pengabdian masyarakat, sehingga data pemantauan ditujukan untuk dapat diakses secara luas. Sementara itu, pada proses bisnis usulan, fokus sistem diarahkan untuk mendukung manajemen filter air sebagai layanan operasional, sehingga informasi kualitas air tidak lagi bersifat sepenuhnya terbuka. Akses terhadap informasi dan data riwayat disesuaikan berdasarkan peran dan kepemilikan agar hanya pihak terkait yang dapat mengakses data sesuai kewenangannya.

Ringkasan perluasan dari sisi proses bisnis disajikan pada Tabel 4.5. Tabel ini merangkum aktivitas/proses yang ditambah dan diubah.

Tabel 4.5 Ringkuman Perubahan Proses Bisnis

<b>Aktivitas/Proses</b>	<b>Status</b>	<b>Ringkasan</b>	<b>Aktor Utama</b>
Manajemen kategori/jenis filter	Ditambah	Mengategorikan filter berdasarkan fungsi dan kegunaan	Operator/Admin
Konfigurasi spesifikasi filter (sensor dan kapasitas)	Ditambah	Spesifikasi dibuat terpisah dari pemasangan filter agar dapat digunakan ulang	Operator/Admin
Manajemen sensor	Diubah	Penambahan batas atas dan bawah untuk setiap sensor	Operator/Admin
Manajemen stok dan identitas unik filter	Ditambah	Dapat memberikan identitas setiap filter berdasarkan spesifikasi	Operator/Admin
Pemasangan filter	Diubah	Pemasangan hanya dapat dilakukan ketika stok tersedia	Operator/Admin
Pemantauan kualitas air berbasis filter terpasang	Diubah	Pemantauan dikaitkan ke kepemilikan filter	Operator/Admin, pengguna filter air
Pengaturan hak akses	Ditambah	Membatasi akses fitur sesuai peran.	Operator/Admin
Keterbukaan Informasi	Diubah	Membatasi akses data dan hanya dapat diakses pengguna filter air	Operator/Admin, Pengguna filter air

Perubahan fitur sistem untuk mendukung proses bisnis usulan dirangkum pada Tabel 4.6. Tabel ini menunjukkan fitur yang ditambah dan diubah ada system yang dikembangkan dibanding sistem sebelumnya.

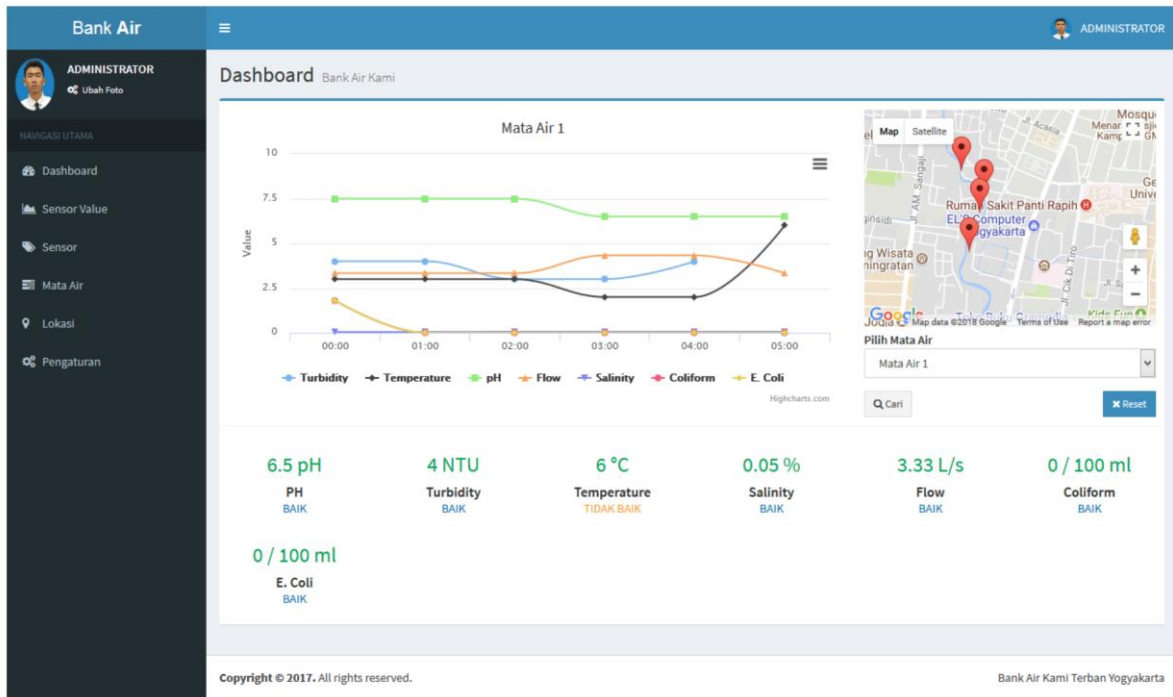
Tabel 4.6 Rangkuman Perbandingan Sistem Berdasarkan Fitur

Fitur/Komponen	Sistem Sebelumnya	Sistem yang Dikembangkan	Status	Keterangan Singkat
Cakupan pemantauan	Fokus 1 filter	Memantau banyak filter	Diubah	Monitoring semua filter Sekaligus
Struktur pemantauan sensor (chart sensor pada dashboard)	Sensor digabung dalam satu tampilan utama (chart)	Lebih terstruktur untuk multi-filter dan multi-sensor	Diubah	Menyesuaikan kebutuhan pemantauan banyak titik/filter berdasarkan masing-masing sensor
Keterbukaan data kualitas air	Diakses secara bebas	Terbatas sesuai peran dan kepemilikan	Diubah	Mendukung konteks layanan/operasional
Template/kategori filter	Tidak ada	Ada	Ditambah	Standardisasi jenis filter
Konfigurasi spesifikasi (sensor dan kapasitas)	Menyatu dengan pemasangan	Terpisah dan dapat digunakan kembali	Diubah	Mengurangi duplikasi input
Stok & kode unik filter	Tidak ada	Ada	Ditambah	Kontrol ketersediaan unit filter berdasarkan konfigurasi/spesifikasi
Validasi pemasangan berbasis stok	Tidak ada	Ada	Ditambah	Pemasangan hanya jika stok tersedia
Manajemen pengguna	Terbatas	Ada	Ditambah	Mendukung pengguna selain operator
RBAC	Tidak ada	Ada	Ditambah	Pembatasan fitur dan data sesuai peran

Untuk memperjelas perubahan fitur yang telah dirangkum pada Tabel 4.6, bagian berikut menampilkan contoh perbandingan berdasarkan tampilan antarmuka. Perbandingan ini bertujuan menunjukkan perubahan secara visual pada fitur inti. Berikut beberapa contoh perbandingan fitur berdasarkan tampilan:

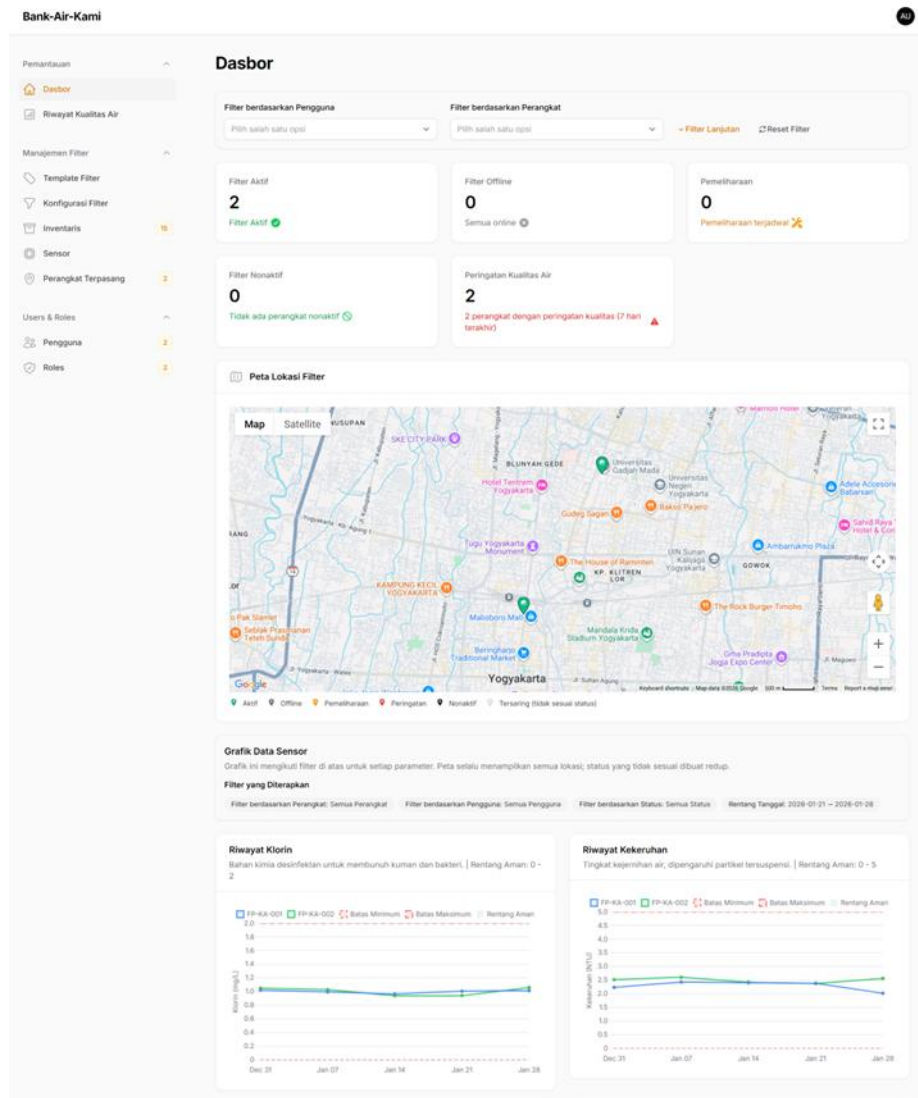
### Pemantauan Kualitas Air

Tampilan pemantauan pada sistem sebelumnya (single filter) dibandingkan dengan sistem yang dikembangkan (multi-filter dan informasi lebih terstruktur) ditunjukkan pada Gambar 4.9 dan Gambar 4.10.



Gambar 4.9 Pemantauan Kualitas Air pada Sistem Sebelumnya

Sumber: Syahputra B (2018)



Gambar 4.10 Pemantauan Kualitas Air pada Sistem yang Dikembangkan

## Riwayat Kualitas Air

Tampilan riwayat kualitas air pada sistem sebelumnya dibandingkan dengan sistem yang dikembangkan ditunjukkan pada Gambar 4.11 dan Gambar 4.12. Pada sistem sebelumnya, riwayat disajikan berbasis mata air dan elemen sensor, sedangkan pada sistem yang dikembangkan riwayat ditampilkan lebih terstruktur berbasis waktu, lokasi, kode perangkat/filter, parameter, dan status, serta didukung fitur pencarian, ekspor, dan detail.

Bank Air

ADMINISTRATOR

Informasi Air Bank Air Kami

Water Value Filtering

5/Nov/2017 - 4/Dec/2017

Water Station: Mata Air

Water Element: All Sensors

No	Date	Mata Air	Sensor	Value	Satuan	Status
1	2017-12-03 00:00:05	Mata Air 1	Flow	3.35	L/s	FL good
2	2017-12-03 00:00:04	Mata Air 1	Salinity	0.05	ppm	SA good
3	2017-12-03 00:00:03	Mata Air 1	Temperature	11	Å°C	TE good
4	2017-12-03 00:00:02	Mata Air 1	Turbidity	10	NTU	TU good
5	2017-12-03 00:00:01	Mata Air 1	PH	4.5	pH	Not good
6	2017-11-30 00:01:00	Mata Air 1	Temperature	11	Å°C	TE good
7	2017-11-30 00:00:05	Mata Air 1	Salinity	0.05	ppm	SA good
8	2017-11-30 00:00:01	Mata Air 2	Turbidity	9	NTU	TU good
9	2017-11-29 09:10:00	Mata Air 1	Turbidity	2	NTU	Not good
10	2017-11-29 09:09:10	Mata Air 1	PH	7.5	pH	PH good

Showing 1 to 10 of 63 entries (filtered from 88 total entries)

Copyright © 2017. All rights reserved. Bank Air Kami Terban Yogyakarta

Gambar 4.11 Riwayat Kualitas Air pada Sistem Sebelumnya

Sumber: Syahputra B (2018)

Bank-Air-Kami

Riwayat Kualitas Air > Daftar

Riwayat Kualitas Air

Semua Data: 1551 OK: 1335 Peringatan: 216

Ekspor

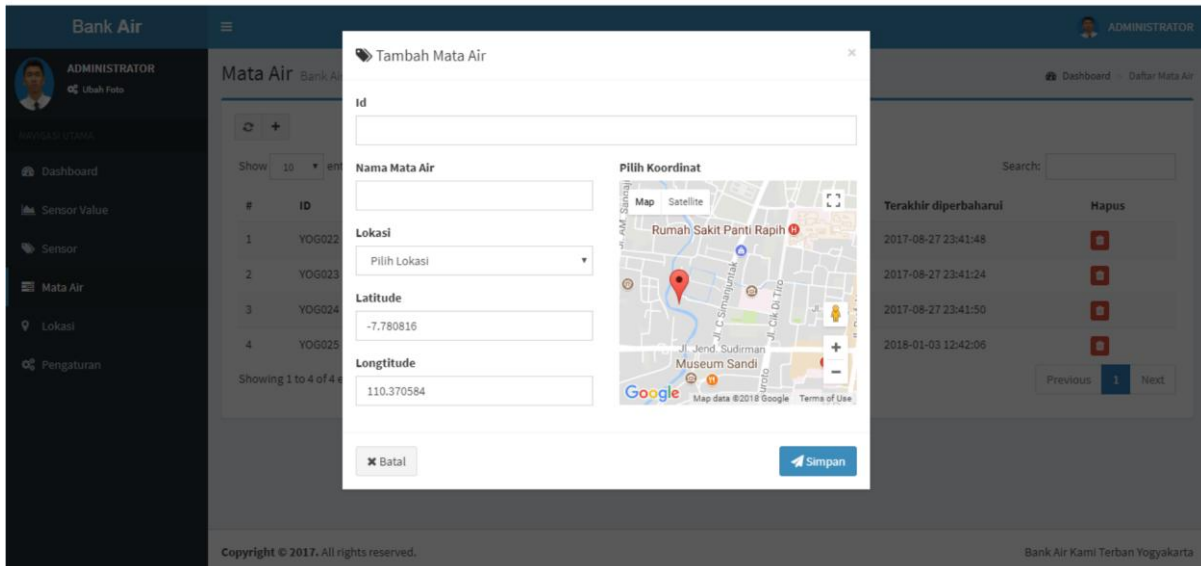
Waktu	Lokasi	Kode Perangkat	Parameter	Status	Sensor Bermasalah
Jan 29, 2026 00:04:02	UGM Campus, Sleman, Yogyakarta	KA-800-02	Chlorine, Kekeruhan, pH, Suhu, TDS	OK	-
Jan 29, 2026 00:04:02	Malioboro, Yogyakarta	KA-800-01	Chlorine, Kekeruhan, pH, Suhu, TDS	OK	-
Jan 29, 2026 00:02:03	Malioboro, Yogyakarta	KA-800-01	Chlorine, Kekeruhan, pH, Suhu, TDS	OK	-
Jan 29, 2026 00:02:03	UGM Campus, Sleman, Yogyakarta	KA-800-02	Chlorine, Kekeruhan, pH, Suhu, TDS	OK	-
Jan 29, 2026 00:00:04	UGM Campus, Sleman, Yogyakarta	KA-800-02	Chlorine, Kekeruhan, pH, Suhu, TDS	OK	-

Gambar 4.12 Riwayat Kualitas Air pada Sistem yang Dikembangkan

## Pemasangan Filter Air

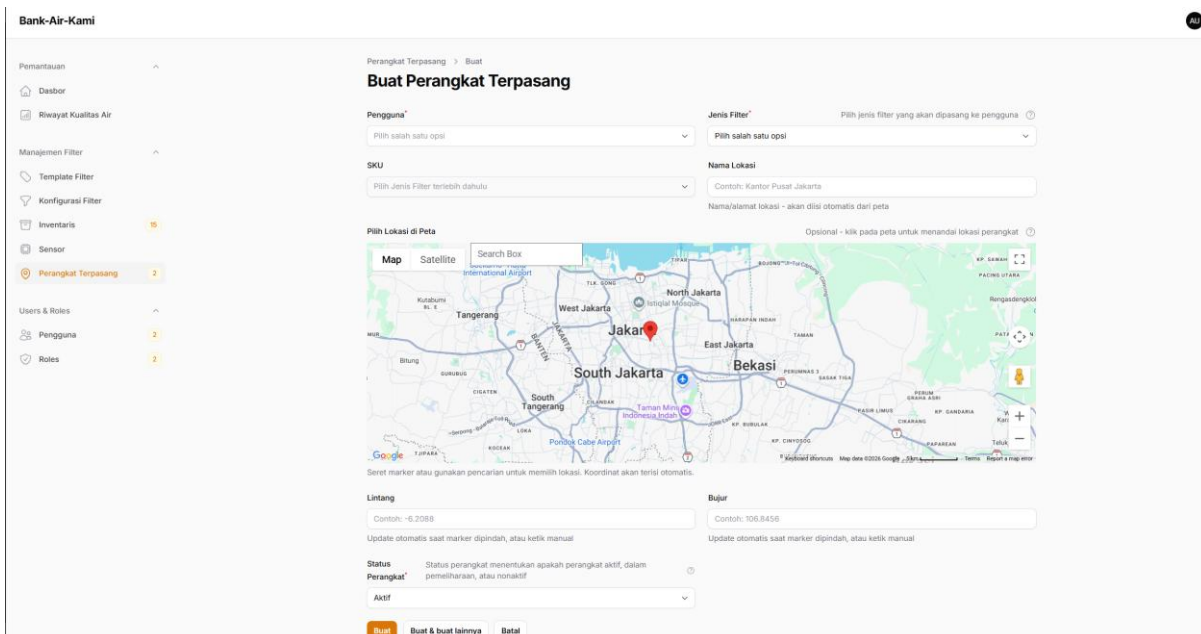
Tampilan pemasangan filter air pada sistem sebelumnya dibandingkan dengan sistem yang dikembangkan ditunjukkan pada Gambar 4.13 dan Gambar 4.14. Pada sistem sebelumnya, pemasangan/pendaftaran titik dilakukan melalui penambahan mata air yang

berfokus pada pengisian identitas lokasi dan koordinat. Pada sistem yang dikembangkan, pemasangan dilakukan melalui fitur Perangkat Terpasang yang mengaitkan pengguna, jenis filter, serta lokasi sehingga proses pemasangan lebih terstruktur untuk kebutuhan manajemen filter.



Gambar 4.13 Pemasangan Filter Air pada Sistem Sebelumnya

Sumber: Syahputra B (2018)



Gambar 4.14 Pemasangan Filter Air pada Sistem yang Dikembangkan

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian, dapat disimpulkan bahwa penelitian ini berhasil menyusun usulan proses bisnis Bank Air Kami. Proses tersebut memperluas layanan dari penyediaan air minum isi ulang menjadi pengelolaan filter air, yang diwujudkan dalam bentuk prototipe dasbor *web* responsif. Dasbor yang dikembangkan mendukung pemantauan parameter kualitas air, pengelolaan stok dan penggunaan filter, serta manajemen pengguna berbasis peran dengan pemisahan yang jelas antara antarmuka, logika bisnis, dan basis data. Pengujian fungsional dan non-fungsional menunjukkan bahwa sistem berperilaku sesuai dengan kebutuhan yang telah dirumuskan dan siap untuk dikembangkan lebih lanjut. Namun, terdapat keterbatasan berupa penggunaan data sekunder, ketidakaktifan infrastruktur IoT secara *realtime*, serta ketiadaan pengujian pengguna. Hal ini menyebabkan kesesuaian sistem dan proses bisnis hanya dapat dinilai pada tingkat rancangan, belum menyentuh konteks operasional nyata di Bank Air Kami.

#### 5.2 Saran

Berdasarkan hasil penelitian dan keterbatasan yang ada, beberapa saran yang dapat diajukan adalah sebagai berikut. Langkah pertama yang diperlukan adalah peninjauan dan penataan ulang infrastruktur IoT Bank Air Kami guna memastikan alur pengukuran dan transmisi data kualitas air berfungsi optimal. Dengan demikian, integrasi sistem dengan dasbor dapat diuji secara langsung dalam kondisi operasional yang sebenarnya. Kedua, usulan proses bisnis pengelolaan filter air harus divalidasi bersama pengelola untuk memastikan kesesuaian alur dengan praktik lapangan dan mengidentifikasi penyesuaian yang diperlukan sebelum implementasi penuh. Ketiga, dokumentasi dan artefak dari berbagai penelitian sebelumnya sebaiknya dihimpun dalam suatu repositori yang terkelola dengan baik. Hal ini diperlukan agar data tersebut dapat dimanfaatkan secara lebih sistematis sebagai dasar pengembangan serta pemeliharaan sistem Bank Air Kami di masa mendatang.

## DAFTAR PUSTAKA

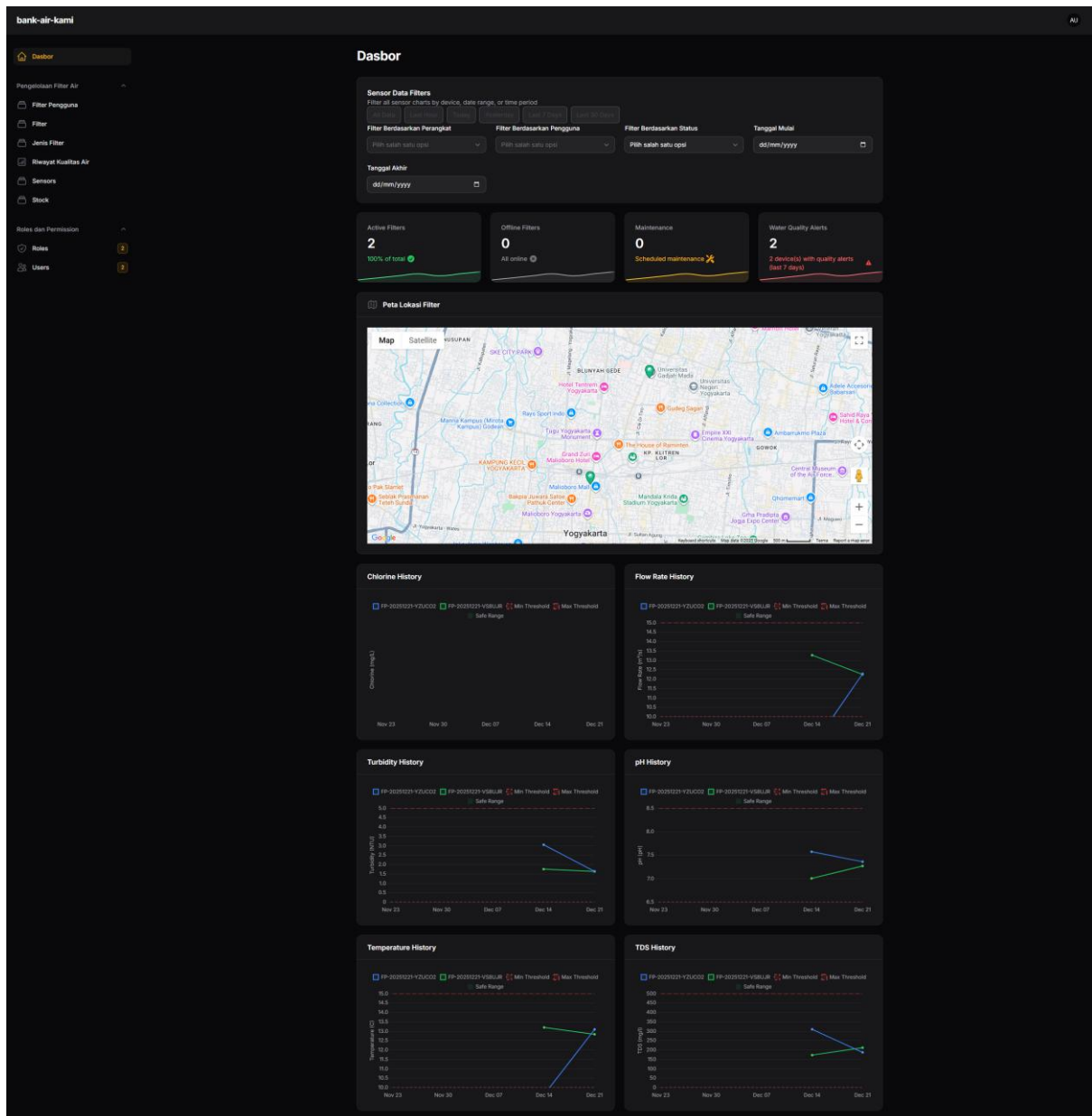
- Al-Said Ahmad, A., & Andras, P. (2019). Scalability analysis comparisons of cloud-based software services. *Journal of Cloud Computing*, 8(1). doi:10.1186/s13677-019-0134-y
- Atalaric, K. F., & Sujarwo, A. (2024). *Sistem IoT Pemantauan Kualitas Air V2 Bank Air Kami*.
- Böhlen, M., & Maharika, I. F. (2018). *Learning from WaterBank*. *Int. J. Electronic Governance* (Vol. 10).
- Carmi, Ofer., & Theocharidou, Marianthi. (2019). *Practical guidelines on requirements of continuous online water-quality monitoring system in drinking-water supply systems*. Publications Office of the European Union.
- Centers for Disease Control and Prevention. (2024). About Choosing Home Water Filters.
- Darise, F. (2016). *TEKNOLOGI PEMROSESAN AIR MINUM DALAM KEMASAN (AMDK) 220 ML MEREK "GC"*. *Jtech* (Vol. 2016).
- Ireland's Environmental Protection Agency. (2020). *Water Treatment Manual: Filtration*. Retrieved from [www.epa.ie](http://www.epa.ie)
- U.S. Environmental Protection Agency. (2005). *Water Health Series Filtration Facts*.
- European Economic Community. (1980). *Official Journal of the European Communities*.
- Gupta, B., Mittal, P., & Mufti, T. (2021). *A Review on Amazon Web Service (AWS), Microsoft Azure & Google Cloud Platform (GCP) Services*. European Alliance for Innovation n.o. doi:10.4108/eai.27-2-2020.2303255
- Javed, A., Malhi, A., Kinnunen, T., & Framling, K. (2020). Scalable IoT Platform for Heterogeneous Devices in Smart Environments. *IEEE Access*, 8, 211973–211985. doi:10.1109/ACCESS.2020.3039368
- Levin, R., Villanueva, C. M., Beene, D., Cradock, A. L., Donat-Vargas, C., Lewis, J., ... Deziel, N. C. (2024, January 1). US drinking water quality: exposure risk profiles for seven legacy and emerging contaminants. *Journal of Exposure Science and Environmental Epidemiology*. Springer Nature. doi:10.1038/s41370-023-00597-z
- Li, W. J., Yen, C., Lin, Y. S., Tung, S. C., & Huang, S. M. (2018). *JustIoT Internet of Things based on the Firebase real-time database*. In *Proceedings - 2018 IEEE International Conference on Smart Manufacturing, Industrial and Logistics*

- Engineering, SMILE 2018* (Vol. 2018-January, pp. 43–47). Institute of Electrical and Electronics Engineers Inc. doi:10.1109/SMILE.2018.8353979
- Malawski, M., Gajek, A., Zima, A., Balis, B., & Figiela, K. (2020). *Serverless Execution of Scientific Workflows: Experiments with HyperFlow, AWS Lambda and Google Cloud Functions*.
- Marcotte, Ethan., & Keith, Jeremy. (2011). *Responsive web design*. A Book Apart.
- Noorkaran Bhanarkar, Aditi Paul, & Dr. Ashima Mehta. (2023). Responsive Web Design and Its Impact on User Experience. *International Journal of Advanced Research in Science, Communication and Technology*, 50–55. doi:10.48175/ijarsct-9259
- Patrick J. Lynch, & Sarah Horton. (2016). *Web Style Guide Foundations of User Experience Design 4th edition*.
- Peraturan Menteri Perindustrian R.I. (2011). Peraturan Menteri Perindustrian RI Nomor 96/M-IND/PER/12/2011.
- Permenkes-No-492-Tahun-2010-tentang-Persyaratan-Kualitas-Air-Minum. (n.d.).
- Pratama, B., & Sujarwo Ari. (2020). Aplikasi Bergerak Pemantau Kualitas Produk Bank Air Kami.
- Rajan, R. A. P. (2020). A review on serverless architectures-Function as a service (FaaS) in cloud computing. *Telkomnika (Telecommunication Computing Electronics and Control)*, 18(1), 530–537. doi:10.12928/TELKOMNIKA.v18i1.12169
- U.S. Environmental Protection Agency. (2024, October 9). National Primary Drinking Water Regulations. Retrieved 16 October 2024, from <https://www.epa.gov/ground-water-and-drinking-water/national-primary-drinking-water-regulations>
- U.S. Environmental Protection Agency. (2015). *Dashboard Design Guidance For Water Quality Surveillance and Response Systems*.
- Van Der Bruggen, B., Verberk, J., & Verhack, J. (2004). *Comparison of pressure-driven membrane processes and traditional processes for drinking water production in Europe based on specific impact criteria* (Vol. 30). Retrieved from <http://www.wrc.org.za>
- Watson, S. E., Taylor, C. H., Bell, V., Hooper, A. S., Bellamy, T. R., Kille, P., & Perkins, R. G. (2024). Utilising eDNA Methods and Interactive Data Dashboards for Managing Sustainable Drinking Water. *Sustainability (Switzerland)* , 16(5). doi:10.3390/su16052043

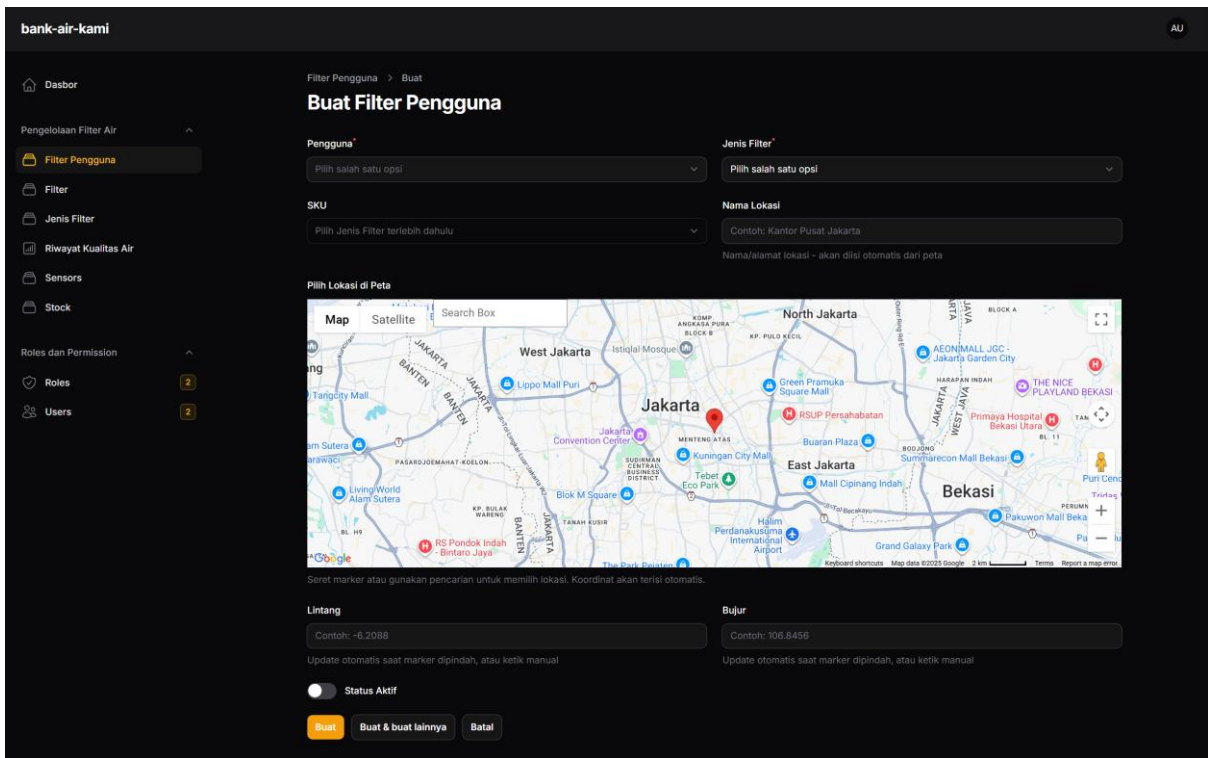
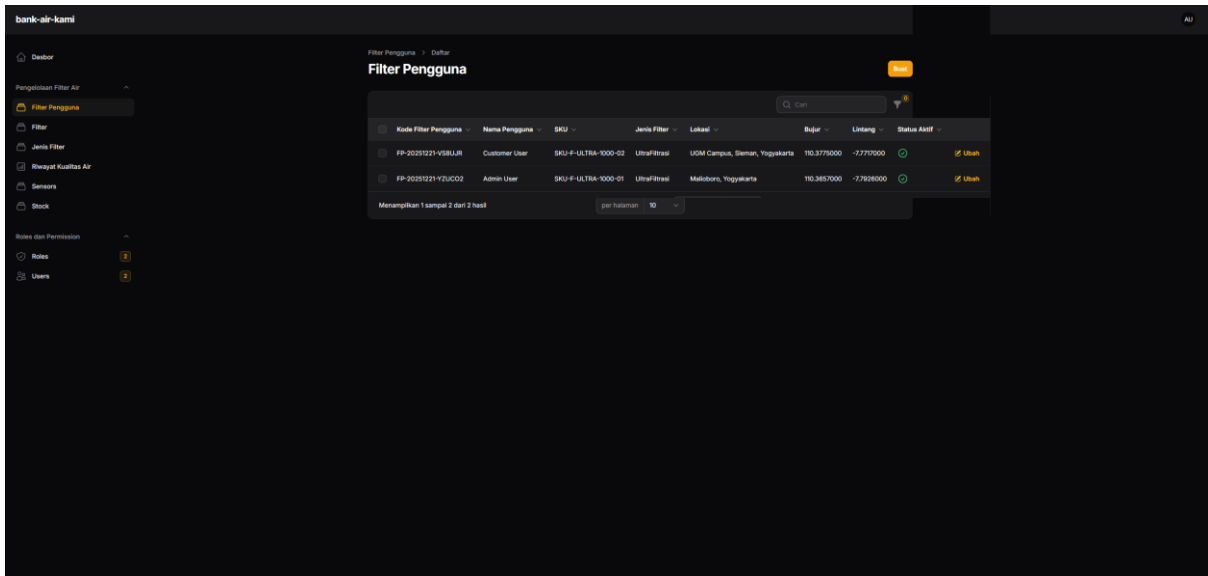
- Wen, X., Chen, F., Lin, Y., Zhu, H., Yuan, F., Kuang, D., ... Yuan, Z. (2020, March 1). Microbial indicators and their use for monitoring drinkingwater quality-A review. *Sustainability (Switzerland)*. MDPI. doi:10.3390/su12062249
- World Health Organization. (2011). *EVALUATING HOUSEHOLD WATER TREATMENT OPTIONS*.
- World Health Organization. (2011). *Guidelines for drinking-water quality*. World Health Organization.
- World Meteorological Organization. (2021). *Lead authors and contributors (in alphabetical order): Report Editorial Board (WMO)*.
- Wu, J., Cao, M., Tong, D., Finkelstein, Z., & Hoek, E. M. V. (2021, December 1). A critical review of point-of-use drinking water treatment in the United States. *Npj Clean Water*. Nature Research. doi:10.1038/s41545-021-00128-z
- Zhang, X., Yang, Y., Ngo, H. H., Guo, W., Wen, H., Wang, X., ... Long, T. (2021, September 1). A critical review on challenges and trend of ultrapure water production process. *Science of the Total Environment*. Elsevier B.V. doi:10.1016/j.scitotenv.2021.147254

# LAMPIRAN

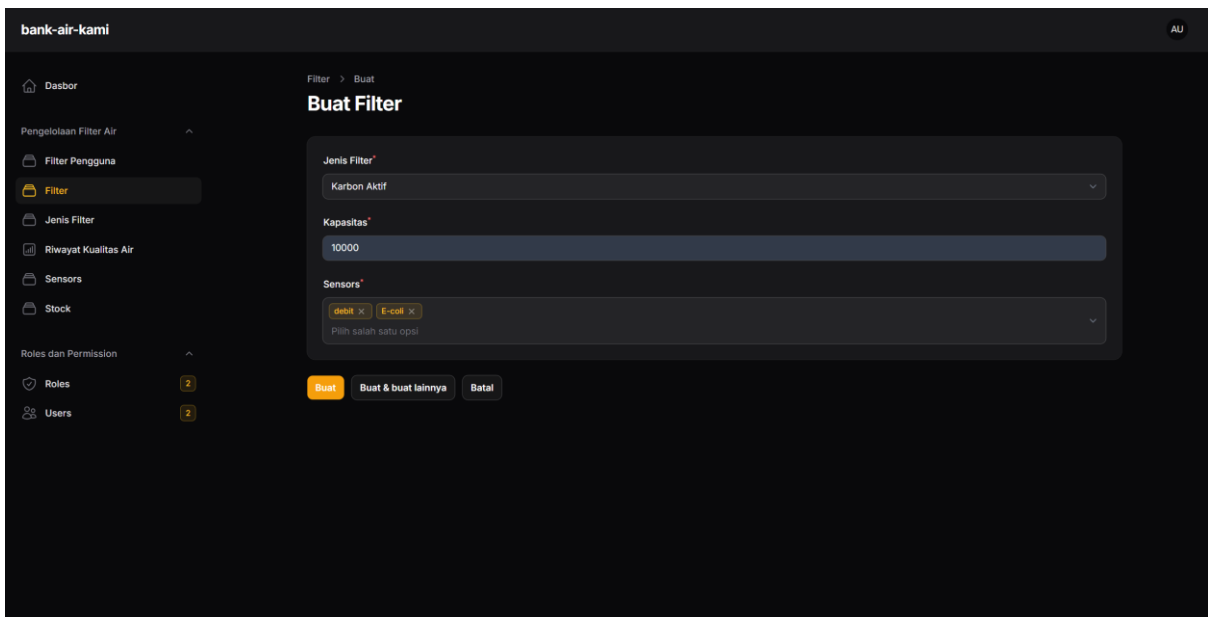
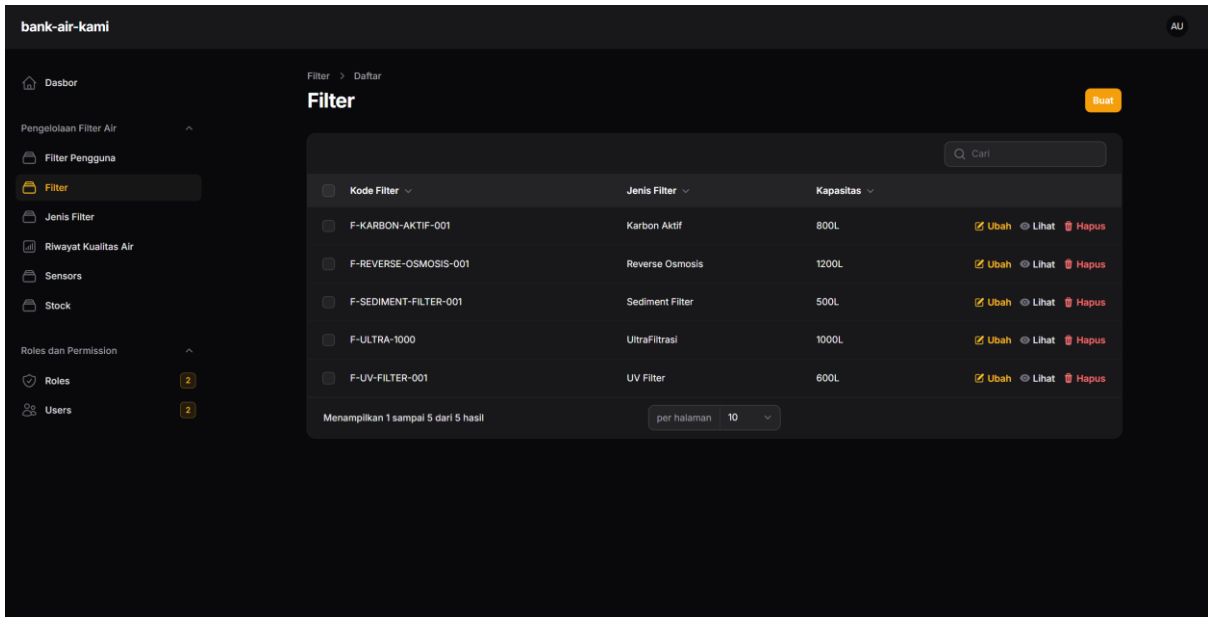
## Lampiran A - Tampilan Fitur Dashboard



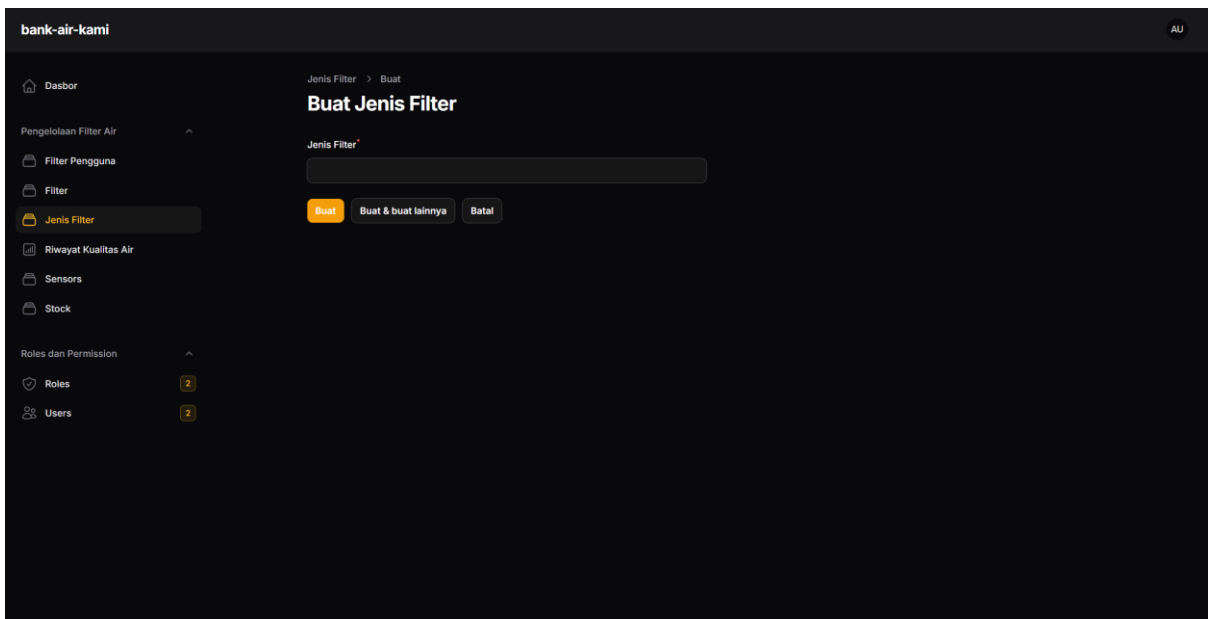
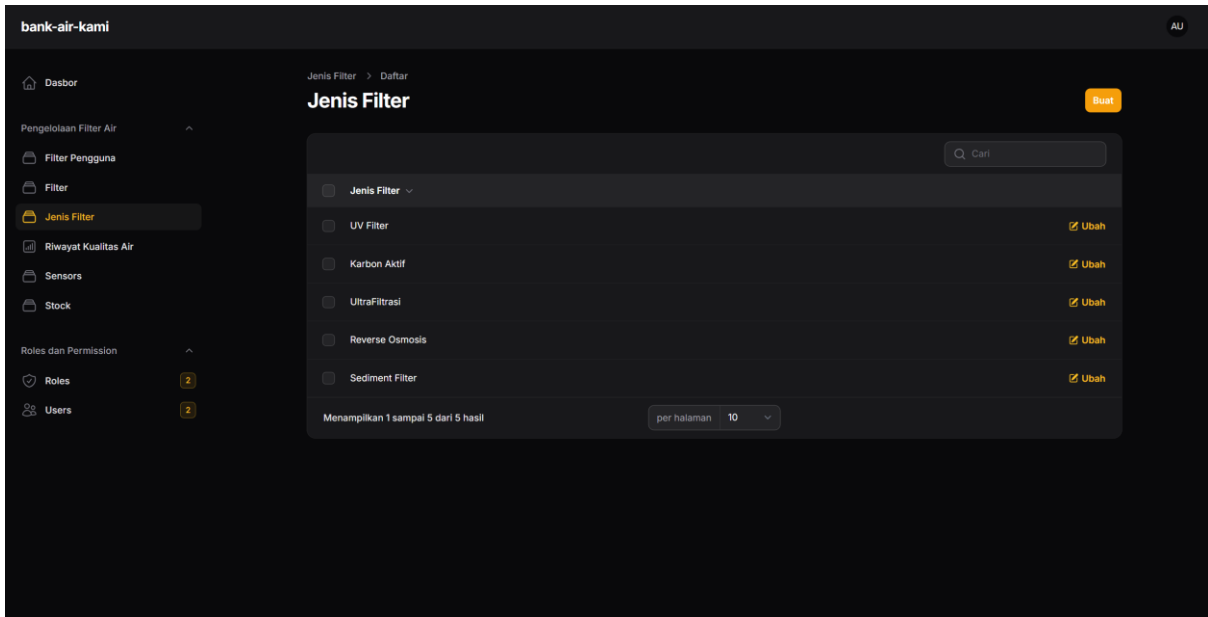
## Lampiran B - Tampilan Fitur Filter Pengguna



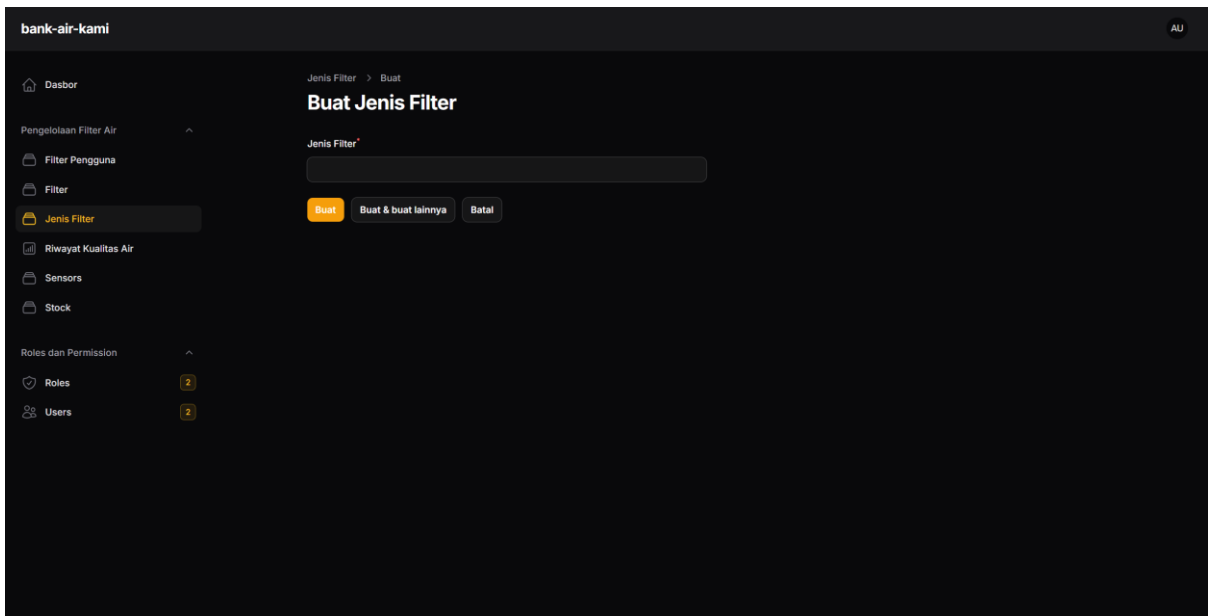
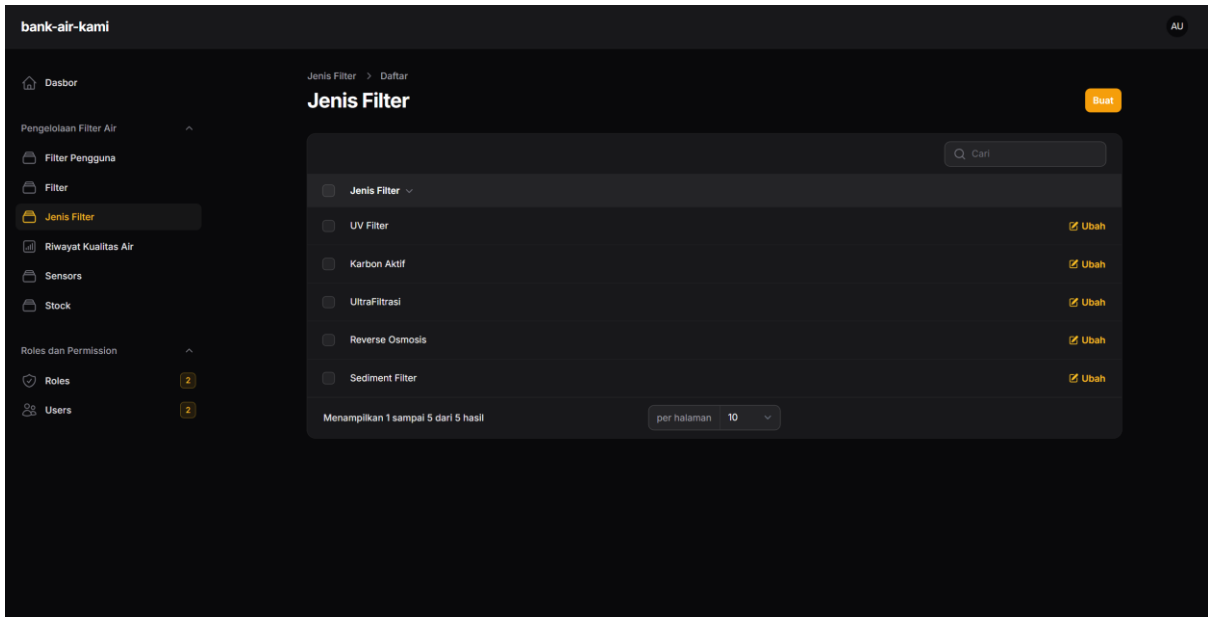
Lampiran C - Tampilan Fitur Filter



Lampiran D - Tampilan Fitur Jenis Filter



Lampiran E - Tampilan Fitur Jenis Filter



Lampiran F - Tampilan Riwayat Kualitas Air

bank-air-kami AU

Dasbor Riwayat Kualitas Air > Daftar

## Riwayat Kualitas Air

All Records (83)
OK (30)
Alert (53)
Ekspor

Waktu	Lokasi	Kode Perangkat	Parameter	Status	Sensor Bermasalah	
Des 21, 2025 08:00:00	Malioboro, Yogyakarta	SKU-F-ULTRA-1000-01	<span>Chlorine</span> <span>debit</span> <span>Kekeruhan</span> <span>pH</span> <span>Suhu</span> <span>TDS</span>	ALERT	Chlorine	Detail
Des 21, 2025 08:00:00	UGM Campus, Sleman, Yogyakarta	SKU-F-ULTRA-1000-02	<span>Chlorine</span> <span>debit</span> <span>Kekeruhan</span> <span>pH</span> <span>Suhu</span> <span>TDS</span>	OK	-	Detail
Des 21, 2025 07:00:00	UGM Campus, Sleman, Yogyakarta	SKU-F-ULTRA-1000-02	<span>Chlorine</span> <span>debit</span> <span>Kekeruhan</span> <span>pH</span> <span>Suhu</span> <span>TDS</span>	ALERT	Chlorine	Detail
Des 21, 2025 07:00:00	Malioboro, Yogyakarta	SKU-F-ULTRA-1000-01	<span>Chlorine</span> <span>debit</span> <span>Kekeruhan</span> <span>pH</span> <span>Suhu</span> <span>TDS</span>	ALERT	Chlorine	Detail
Des 21, 2025 06:59:29	Malioboro, Yogyakarta	SKU-F-ULTRA-1000-01	<span>Chlorine</span> <span>debit</span> <span>Kekeruhan</span> <span>pH</span> <span>Suhu</span> <span>TDS</span>	ALERT	Chlorine	Detail
Des 21, 2025 06:00:00	Malioboro, Yogyakarta	SKU-F-ULTRA-1000-01	<span>Chlorine</span> <span>debit</span> <span>Kekeruhan</span> <span>pH</span> <span>Suhu</span> <span>TDS</span>	ALERT	Chlorine	Detail
Des 21, 2025 06:00:00	UGM Campus, Sleman, Yogyakarta	SKU-F-ULTRA-1000-02	<span>Chlorine</span> <span>debit</span> <span>Kekeruhan</span> <span>pH</span> <span>Suhu</span> <span>TDS</span>	ALERT	Chlorine	Detail
Des 21, 2025 05:00:00	Malioboro, Yogyakarta	SKU-F-ULTRA-1000-01	<span>Chlorine</span> <span>debit</span> <span>Kekeruhan</span> <span>pH</span> <span>Suhu</span> <span>TDS</span>	ALERT	Chlorine	Detail
Des 21, 2025 05:00:00	UGM Campus, Sleman, Yogyakarta	SKU-F-ULTRA-1000-02	<span>Chlorine</span> <span>debit</span> <span>Kekeruhan</span> <span>pH</span> <span>Suhu</span> <span>TDS</span>	OK	-	Detail
Des 21, 2025 04:00:00	UGM Campus, Sleman, Yogyakarta	SKU-F-ULTRA-1000-02	<span>Chlorine</span> <span>debit</span> <span>Kekeruhan</span> <span>pH</span> <span>Suhu</span> <span>TDS</span>	OK	-	Detail

Menampilkan 1 sampai 10 dari 83 hasil

per halaman 10
 1 2 3 4 ... 8 9 >

### Detail Kualitas Air - Malioboro, Yogyakarta (SKU-F-ULTRA-1000-01)

Waktu: Dec 21, 2025 08:00:00
 Lokasi: Malioboro, Yogyakarta
 Kode Perangkat: SKU-F-ULTRA-1000-01

Parameter	Nilai	Satuan	Batas	Status	Keterangan
Chlorine	0.03	mg/L	0.00 - 0.00	⚠ Peringatan	-
debit	13.00	m <sup>3</sup> /s	10.00 - 15.00	✓ OK	-
Kekeruhan	0.00	NTU	0.00 - 5.00	✓ OK	-
pH	7.40	pH	6.50 - 8.50	✓ OK	-
Suhu	13.60	C	10.00 - 15.00	✓ OK	-
TDS	180.00	mg/l	0.00 - 500.00	✓ OK	-

Status Keseluruhan: ⚠ PERINGATAN - Chlorine melebihi batas aman
 6 parameter diukur

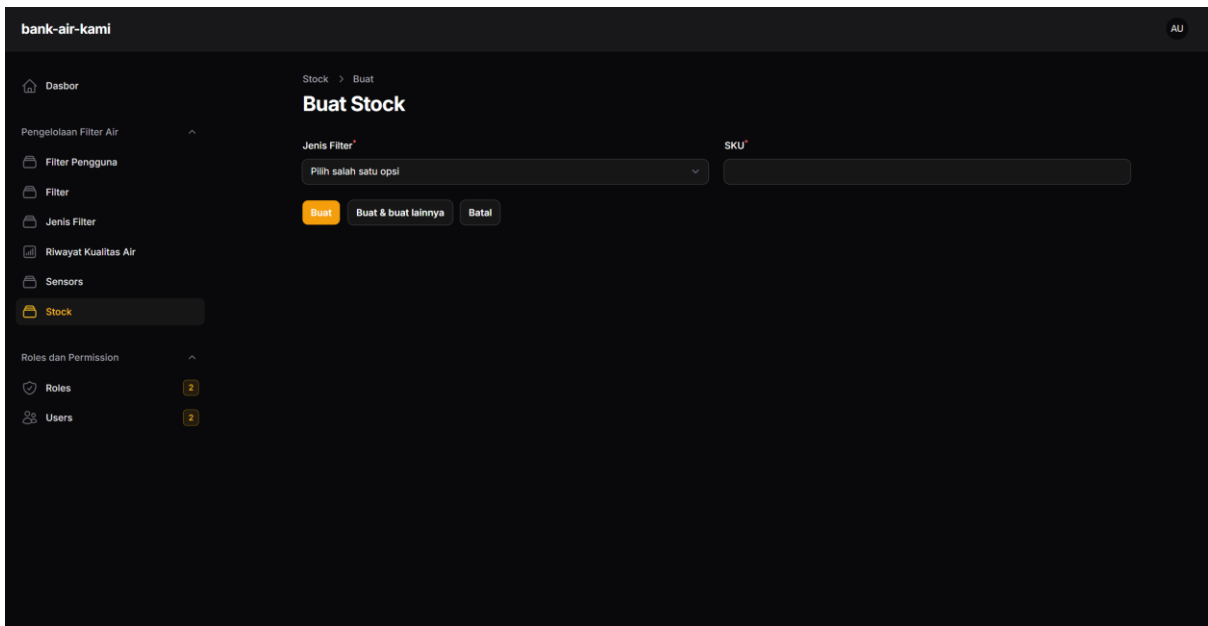
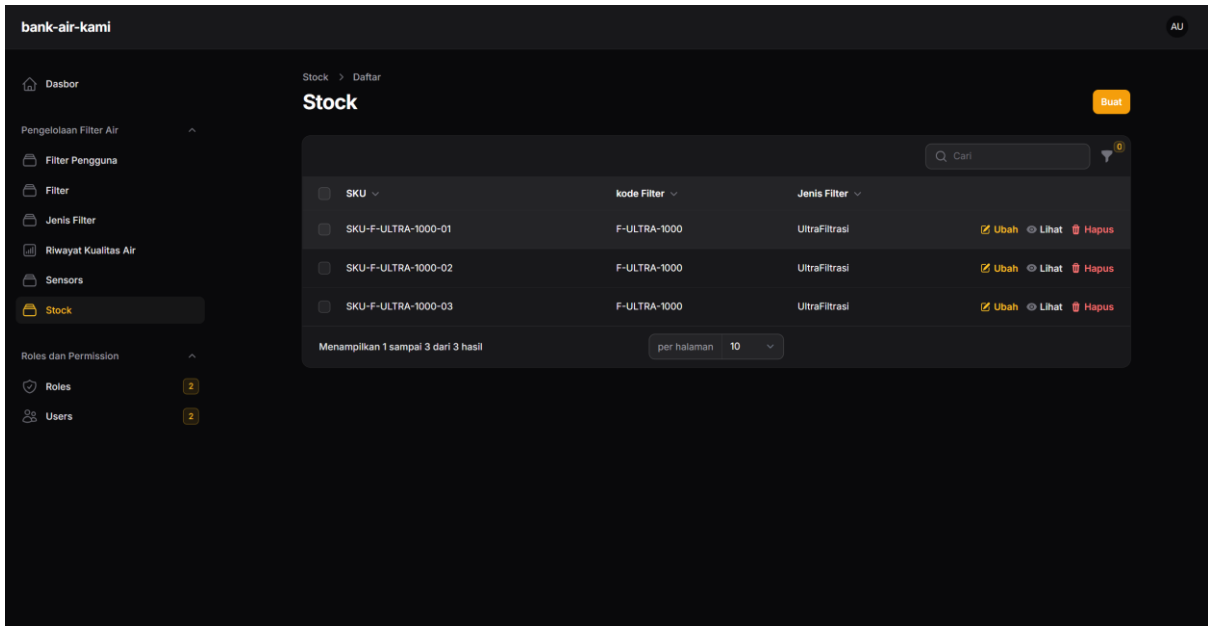
## Lampiran G - Tampilan Fitur Sensor

The screenshot shows the 'Sensors' list page in the 'bank-air-kami' application. The page has a dark theme and a sidebar on the left with navigation options: Dasbor, Pengelolaan Filter Air (Filter Pengguna, Filter, Jenis Filter, Riwayat Kualitas Air), and Roles dan Permission (Roles, Users). The main content area is titled 'Sensors' and contains a table of sensor parameters. A search bar and a 'Buat' button are at the top right. The table has columns for Parameter, Satuan, Batas Minimum, and Batas Maksimum, with action buttons (Ubah, Lihat, Hapus) for each row. At the bottom, it shows 'Menampilkan 1 sampai 9 dari 9 hasil' and a pagination control for 10 items per page.

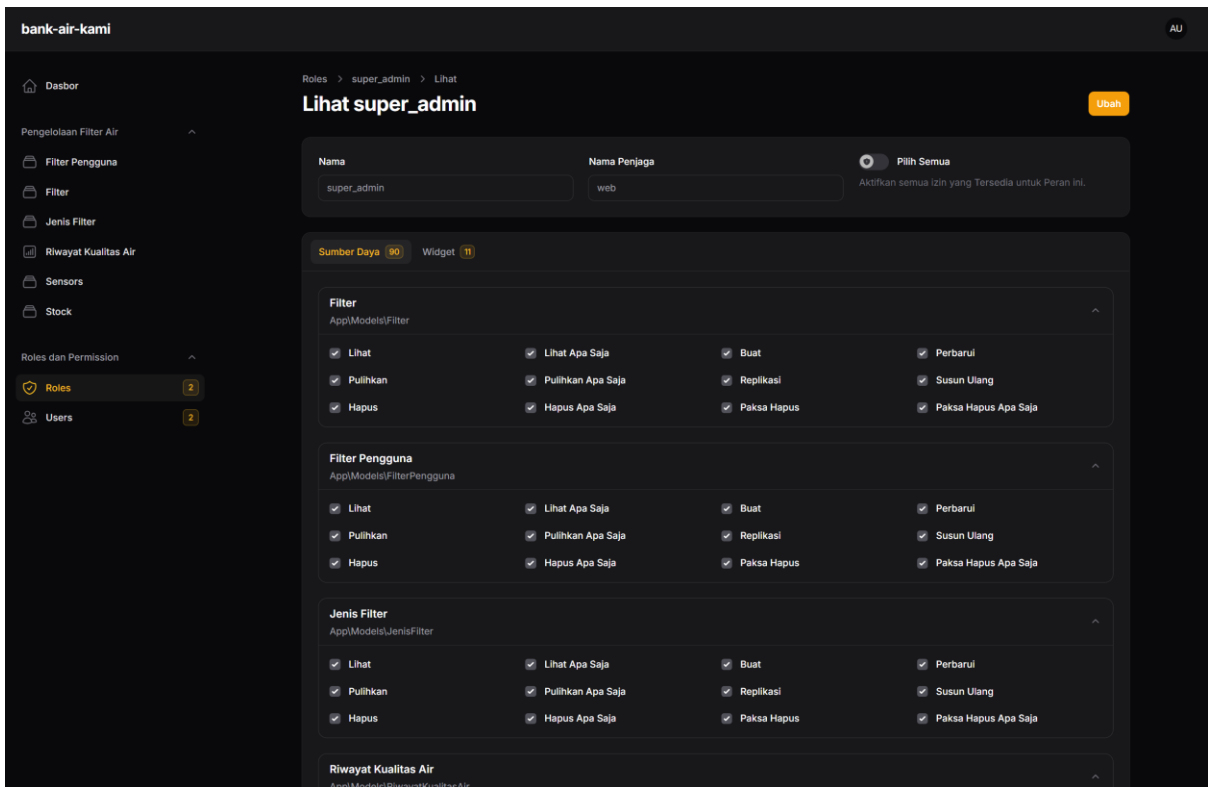
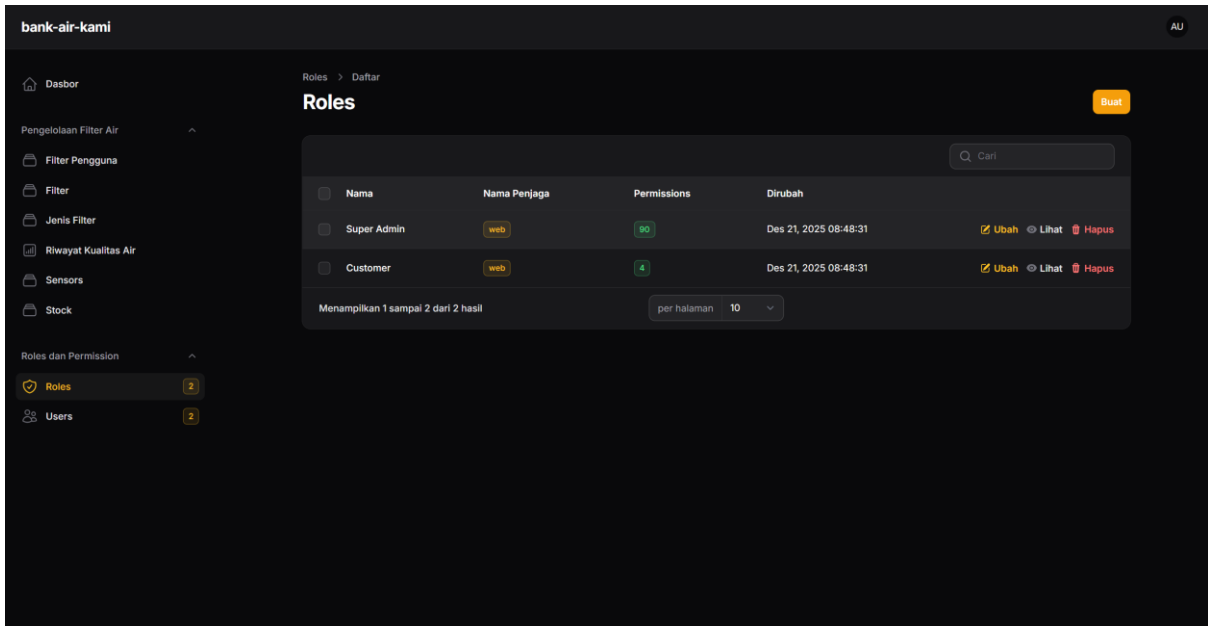
Parameter	Satuan	Batas Minimum	Batas Maksimum	
Chlorine	mg/L	0	0	Ubah Lihat Hapus
debit	m <sup>3</sup> /s	10	15	Ubah Lihat Hapus
E-coli	CFU/100ml	0	100	Ubah Lihat Hapus
Heavy_Metals	mg/L	0	0,05	Ubah Lihat Hapus
Kekeruhan	NTU	0	5	Ubah Lihat Hapus
pH	pH	6,5	8,5	Ubah Lihat Hapus
Salinity	ppt	0	35	Ubah Lihat Hapus
Suhu	C	10	15	Ubah Lihat Hapus
TDS	mg/l	0	500	Ubah Lihat Hapus

The screenshot shows the 'Buat Sensor' form in the 'bank-air-kami' application. The page has a dark theme and a sidebar on the left with navigation options: Dasbor, Pengelolaan Filter Air (Filter Pengguna, Filter, Jenis Filter, Riwayat Kualitas Air), and Roles dan Permission (Roles, Users). The main content area is titled 'Buat Sensor' and contains four input fields: Parameter, Satuan, Batas Minimum, and Batas Maksimum. Below the fields are three buttons: 'Buat', 'Buat & buat lainnya', and 'Batal'.

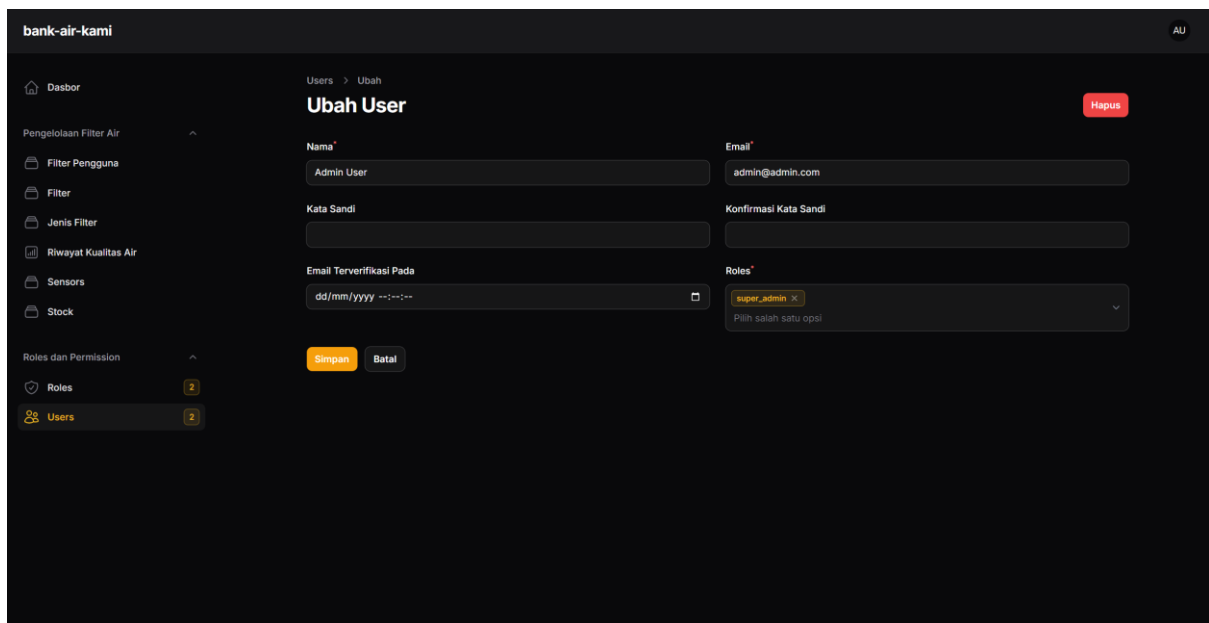
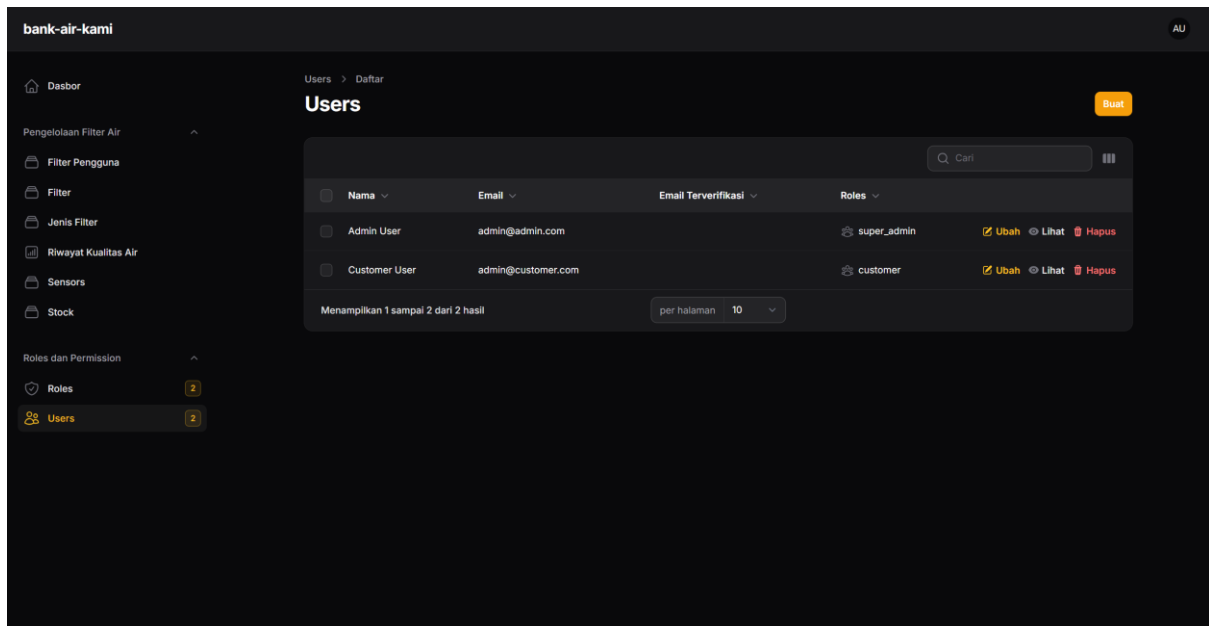
## Lampiran H - Tampilan Fitur Stock



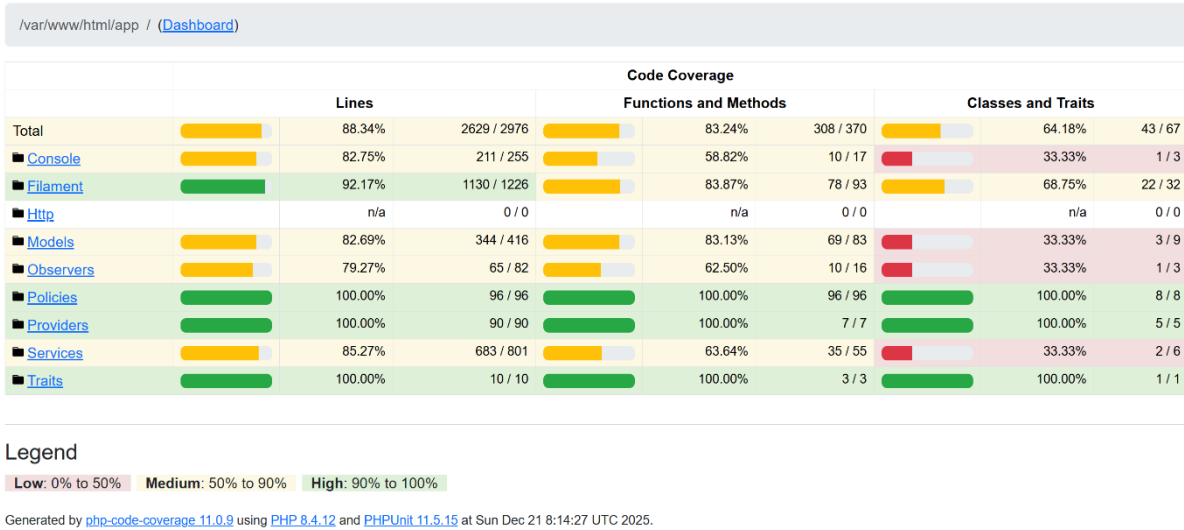
Lampiran I - Tampilan Fitur Role



Lampiran J - Tampilan Fitur Management Pengguna



Lampiran K - Hasil Coverage Dengan Menggunakan Generate PHP-Code-Coverage 11.0.9



## Lampiran L - Hasil Testing Parallel Menggunakan Terminal

```

bank-air-kami on staging [?] via v22.17.1 via v8.4.1 on daud28cloud1@gmail.com took 12m56s
) ./test-runners.sh all
[INFO] Running all tests...
[+] restart 0/2
i: Container bank-air-kami-mysql-1 Restarting 3.7s
i: Container bank-air-kami-laravel.test-1 Restarting 3.7s

Dropping all tables ..... 305.45ms DONE

[INFO] Preparing database.

Creating migration table ..... 81.28ms DONE

[INFO] Running migrations.

0001_01_01_000000_create_users_table ..... 124.82ms DONE
0001_01_01_000001_create_cache_table ..... 45.75ms DONE
0001_01_01_000002_create_jobs_table ..... 119.26ms DONE
2025_05_16_145943_create_permission_tables ..... 523.78ms DONE
2025_06_14_012506_create_jenis_filters_table ..... 64.90ms DONE
2025_06_15_144605_create_filters_table ..... 145.99ms DONE
2025_06_15_144606_create_sensors_table ..... 62.83ms DONE
2025_06_16_000001_create_filter_sensor_table ..... 166.50ms DONE
2025_06_16_130429_create_stocks_table ..... 77.03ms DONE
2025_06_20_152803_create_filter_pengguna_table ..... 32.20ms DONE
2025_09_05_110000_add_device_status_to_filter_pengguna ..... 443.16ms DONE
2025_09_17_023223_normalize_water_quality_history_tables ..... 528.73ms DONE
2025_10_14_100000_add_cascade_to_niwayat_kualitas_air_foreign_key ..... 150.08ms DONE
2025_10_14_200000_add_foreign_keys_to_filter_pengguna ..... 329.17ms DONE

.....

.....

.....

Tests: 772 passed (3034 assertions)
Duration: 1017.68s
Parallel: 8 processes
  
```

## Lampiran M - Tabel Hasil Detail Pengujian

Test suite	Jumlah test	Total durasi (s)	Rata-rata (s)	Maks (s)	Kategori
------------	-------------	------------------	---------------	----------	----------

Tests\Feature\Commands\FirebaseSimulationCommandTest	10	15.63	1.563	3.48	<i>Feature</i>
Tests\Feature\Commands\SyncFirebaseToMysqlTest	9	5.14	0.571	0.62	<i>Feature</i>
Tests\Feature\Console\Commands\AutoSensorDataCacheCommandTest	16	26.32	1.645	3.20	<i>Feature</i>
Tests\Feature\Console\Commands\GenerateSensorWidgetsCommandTest	15	1.97	0.131	0.21	<i>Feature</i>
Tests\Feature\Console\Commands\ManageSensorConfidenceCommandTest	25	2.79	0.112	0.14	<i>Feature</i>
Tests\Feature\Console\Commands\SyncFirebaseCommandTest	14	21.77	1.555	2.29	<i>Feature</i>
Tests\Feature\Dashboard\DashboardDataTest	12	19.71	1.642	2.38	<i>Feature</i>
Tests\Feature\Resources\FilterPenggunaResourceTest	31	80.69	2.603	4.03	<i>Feature</i>
Tests\Feature\Resources\FilterResourceTest	20	29.30	1.465	1.97	<i>Feature</i>
Tests\Feature\Resources\JenisFilterResourceTest	16	18.91	1.182	1.96	<i>Feature</i>
Tests\Feature\Resources\RiwayatKualitasAirFilterTest	18	53.18	2.954	3.73	<i>Feature</i>
Tests\Feature\Resources\RiwayatKualitasAirResourceTest	45	113.37	2.519	7.17	<i>Feature</i>
Tests\Feature\Resources\RoleResourceTest	37	63.33	1.712	4.19	<i>Feature</i>
Tests\Feature\Resources\SensorResourceTest	14	20.35	1.454	2.45	<i>Feature</i>
Tests\Feature\Resources\StockResourceTest	22	32.82	1.492	3.55	<i>Feature</i>
Tests\Feature\Resources\UsersResourceTest	13	25.63	1.972	2.64	<i>Feature</i>
Tests\Feature\Security\AuthenticationAndAuthorizationTest	9	10.52	1.169	1.88	<i>Feature</i>
Tests\Feature\Security\DataIntegrityAndValidationTest	14	24.66	1.761	3.17	<i>Feature</i>
Tests\Feature\Security\ValidationAndErrorHandlingTest	17	34.39	2.023	6.05	<i>Feature</i>
Tests\Feature\WaterQuality\AnomalyDetectionTest	15	33.43	2.229	3.98	<i>Feature</i>
Tests\Feature\WaterQuality\FilterHealthMonitoringTest	15	24.73	1.649	2.46	<i>Feature</i>
Tests\Feature\WaterQuality\QualityStatusCalculationTest	28	48.25	1.723	3.37	<i>Feature</i>
Tests\Unit\Middleware\BasicMiddlewareTest	4	4.35	1.088	3.74	<i>Unit</i>
Tests\Unit\Middleware\HandleUnauthorizedAccessSimpleTest	11	1.06	0.096	0.12	<i>Unit</i>
Tests\Unit\Middleware\MiddlewareTest	17	1.55	0.091	0.13	<i>Unit</i>

Tests\Unit\Middleware\RedirectCustomerToCustomerPanelTest	9	1.86	0.207	1.17	<i>Unit</i>
Tests\Unit\Models\DeviceStatusHistoryTest	10	4.20	0.420	3.12	<i>Unit</i>
Tests\Unit\Models\FilterPeggunaDeviceStatusTest	19	7.74	0.407	0.49	<i>Unit</i>
Tests\Unit\Models\FilterPeggunaModelTest	12	22.58	1.882	1.99	<i>Unit</i>
Tests\Unit\Models\FilterPeggunaTest	15	5.86	0.391	0.43	<i>Unit</i>
Tests\Unit\Models\FilterTest	10	4.18	0.418	0.76	<i>Unit</i>
Tests\Unit\Models\JenisFilterTest	10	4.16	0.416	0.61	<i>Unit</i>
Tests\Unit\Models\RiwayatKualitasAirDetailModelTest	24	2.86	0.119	0.16	<i>Unit</i>
Tests\Unit\Models\RiwayatKualitasAirTest	4	0.56	0.140	0.17	<i>Unit</i>
Tests\Unit\Models\SensorModelTest	25	2.27	0.091	0.11	<i>Unit</i>
Tests\Unit\Models\StockModelTest	9	1.18	0.131	0.30	<i>Unit</i>
Tests\Unit\Models\StockTest	2	0.27	0.135	0.14	<i>Unit</i>
Tests\Unit\Models\UserTest	7	1.20	0.171	0.67	<i>Unit</i>
Tests\Unit\Observers\FilterPeggunaObserverTest	10	10.82	1.082	1.60	<i>Unit</i>
Tests\Unit\Observers\RiwayatKualitasAirDetailObserverTest	7	4.90	0.700	0.79	<i>Unit</i>
Tests\Unit\Observers\SensorObserverTest	11	1.07	0.097	0.12	<i>Unit</i>
Tests\Unit\Policies\FilterPeggunaPolicyTest	5	6.08	1.216	1.56	<i>Unit</i>
Tests\Unit\Policies\FilterPolicyTest	5	6.25	1.250	1.71	<i>Unit</i>
Tests\Unit\Policies\JenisFilterPolicyTest	5	6.10	1.220	1.52	<i>Unit</i>
Tests\Unit\Policies\RiwayatKualitasAirPolicyTest	5	6.24	1.248	1.49	<i>Unit</i>
Tests\Unit\Policies\RolePolicyTest	3	4.49	1.497	1.56	<i>Unit</i>
Tests\Unit\Policies\SensorPolicyTest	4	5.60	1.400	1.52	<i>Unit</i>
Tests\Unit\Policies\StockPolicyTest	3	4.41	1.470	1.52	<i>Unit</i>
Tests\Unit\Policies\UserPolicyTest	3	4.48	1.493	1.63	<i>Unit</i>
Tests\Unit\Providers\AppServiceProviderTest	8	0.66	0.083	0.09	<i>Unit</i>
Tests\Unit\Providers\AuthServiceProviderTest	9	0.72	0.080	0.09	<i>Unit</i>
Tests\Unit\Resources\FilterPeggunaResourceTest	14	19.50	1.393	2.42	<i>Unit</i>
Tests\Unit\Resources\FilterResourceTest	8	0.86	0.108	0.13	<i>Unit</i>
Tests\Unit\Resources\JenisFilterResourceTest	8	0.84	0.105	0.13	<i>Unit</i>
Tests\Unit\Resources\RiwayatKualitasAirResourceTest	20	22.70	1.135	1.58	<i>Unit</i>

Tests\Unit\Resources\RoleResourceTest	8	0.90	0.113	0.14	<i>Unit</i>
Tests\Unit\Resources\SensorResourceTest	8	0.87	0.109	0.14	<i>Unit</i>
Tests\Unit\Resources\StockResourceTest	8	0.90	0.113	0.14	<i>Unit</i>
Tests\Unit\Resources\UsersResourceTest	8	0.93	0.116	0.14	<i>Unit</i>
Tests\Unit\Services\FirebaseServiceTest	11	4.78	0.435	1.60	<i>Unit</i>
Tests\Unit\Services\SensorDataGenerationServiceTest	10	10.04	1.004	2.54	<i>Unit</i>
Tests\Unit\Services\WidgetGenerationServiceTest	10	9.88	0.988	2.44	<i>Unit</i>

## Lampiran N – Kode CI/CD GitHub Action

```

name: CI - Laravel Tests, Coverage & Deploy

on:
  push:
    branches: [ develop ]
  pull_request:
    branches: [ develop ]

jobs:
  tests:
    runs-on: ubuntu-latest
    services:
      mysql:
        image: mysql:8.0
        env:
          MYSQL_ROOT_PASSWORD: password
          MYSQL_DATABASE: bank_air_kami
          MYSQL_USER: sail
          MYSQL_PASSWORD: password
        ports:
          - 3306:3306
        options: >-
          --health-cmd="mysqladmin ping --silent"
          --health-interval=10s
          --health-timeout=5s
          --health-retries=3

    steps:
      - name: Checkout repository
        uses: actions/checkout@v3

      - name: Cache Composer deps
        uses: actions/cache@v3
        with:
          path: vendor
          key: ${{ runner.os }}-composer-${{
hashFiles('**/composer.lock') }}
          restore-keys: ${{ runner.os }}-composer-

```

```

- name: Set up PHP 8.3 + Xdebug
  uses: shivammathur/setup-php@v2
  with:
    php-version: '8.3'
    coverage: xdebug
    extensions: mbstring, bcmath, xml, pdo_mysql
    ini-values: xdebug.mode=coverage

- name: Install Composer dependencies
  run: composer install --no-interaction --prefer-dist

- name: Setup .env.testing for GitHub Actions
  run: |
    cp .env.testing .env
    php artisan key:generate --ansi
    # Update MySQL host for GitHub Actions in both .env and
.env.testing
    sed -i 's/^DB_HOST=mysql/DB_HOST=127.0.0.1/' .env
    sed -i 's/^DB_HOST=mysql/DB_HOST=127.0.0.1/' .env.testing
    # Update phpunit.xml to use correct database for CI/CD
    sed -i 's/<env name="DB_DATABASE" value="testing"\/><env
name="DB_DATABASE" value="bank_air_kami"\/>/' phpunit.xml
    # Use root user for testing to avoid permission issues with
parallel testing
    sed -i 's/^DB_USERNAME=sail/DB_USERNAME=root/' .env
    sed -i 's/^DB_USERNAME=sail/DB_USERNAME=root/' .env.testing
    # Create Firebase credentials file from secret
    mkdir -p storage/app
    echo "${{ secrets.FIREBASE_CREDENTIALS_BASE64 }}" >
storage/app/firebase-admin.json

- name: Wait for MySQL to be ready
  run: |
    until mysqladmin ping -h127.0.0.1 --silent; do
      echo 'Waiting for MySQL...'
      sleep 1
    done

- name: Run database migrations
  run: php artisan migrate:fresh --seed --force

- name: Run PHPUnit (generate coverage.xml)
  run: php artisan test --coverage --parallel --processes=8 --
min=80.0
  env:
    XDEBUG_MODE: coverage

# Deployment job for Google Cloud VM (disabled)
deploy:
  needs: tests
  if: github.event_name == 'push'
  runs-on: ubuntu-latest
  steps:
    # Set up the SSH key and add the VM host to known_hosts
    - name: Set up SSH key
      run: |
        mkdir -p ~/.ssh
        echo -e "${{ secrets.GCP_SSH_PRIVATE_KEY }}" > ~/.ssh/bank-

```

```

air-kami
    chmod 600 ~/.ssh/bank-air-kami
    # Add the VM's host key so SSH won't prompt
    ssh-keyscan ${ secrets.HOST_GCP } >> ~/.ssh/known_hosts

- name: show private-key header and fingerprint
  run: |
    # Show the first line of the key (should be something like "-
----BEGIN OPENSSH PRIVATE KEY-----" or "-----BEGIN RSA PRIVATE KEY-----")
    head -n1 ~/.ssh/bank-air-kami
    # Show the fingerprint of that key
    ssh-keygen -lf ~/.ssh/bank-air-kami

# Execute the deployment commands on the VM via SSH
- name: Debug SSH with hardcoded values
  run: |
    ssh -vvv \
      -o StrictHostKeyChecking=no \
      -o IdentitiesOnly=yes \
      -i ~/.ssh/bank-air-kami \
      ${ secrets.USERNAME_GCP }@${ secrets.HOST_GCP } \
      'echo "Connected as $(whoami) "'

- name: Run deployment commands via SSH
  run: |
    ssh -i ~/.ssh/bank-air-kami ${ secrets.USERNAME_GCP }@${
secrets.HOST_GCP } <<'EOF'
    set -e
    cd /home/sagash/bank-air-kami

    echo "Pulling latest code..."
    git pull --rebase origin develop

    echo "Stopping existing containers..."
    ./vendor/bin/sail down

    echo "Building containers..."
    ./vendor/bin/sail build --no-cache

    echo "Starting containers..."
    ./vendor/bin/sail up -d

    echo "Waiting for MySQL to be ready..."
    until ./vendor/bin/sail exec mysql mysqladmin ping --silent;
do
    echo 'Waiting for MySQL...'
    sleep 5
done

    echo "Running database migrations..."
    ./vendor/bin/sail artisan migrate:fresh --seed --force

    echo "Deployment completed successfully!"
EOF

```