

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Sistem Pengelolaan Sampah Otomatis

Pengelolaan sampah merupakan salah satu tantangan utama dalam sektor kebersihan dan kesehatan lingkungan, khususnya di wilayah perkotaan yang mengalami peningkatan populasi dan konsumsi. Sistem pengelolaan sampah otomatis adalah pendekatan modern yang memanfaatkan teknologi untuk meningkatkan efektivitas, efisiensi, dan keamanan dalam proses pengumpulan dan pemantauan sampah.

Secara teknis, sistem ini menggunakan kombinasi sensor (seperti ultrasonik, load cell, dan gas sensor), mikrokontroler (seperti Arduino atau NodeMCU), aktuator (seperti motor servo), serta konektivitas berbasis Internet of Things (IoT). Sistem otomatis ini memungkinkan deteksi kondisi tempat sampah secara real-time, aktivasi mekanisme buka-tutup secara otomatis, serta pengiriman data ke platform digital seperti Firebase atau Google Maps API untuk pemantauan jarak jauh.

Menurut penelitian oleh Armanda et al. (2021), pemanfaatan Google Firebase sebagai sistem backend berbasis real-time memungkinkan pemantauan ketinggian sampah secara kontinu tanpa interaksi langsung manusia. Sementara itu, Bekti et al. (2020) menunjukkan bahwa integrasi sistem pemantauan dengan Google Maps API mampu memvisualisasikan lokasi tempat sampah penuh dan menentukan rute optimal pengangkutan. Oleh karena itu, sistem pengelolaan sampah otomatis

menjadi salah satu solusi berbasis teknologi yang menjanjikan dalam mewujudkan kota pintar (*smart city*).

2.1.1.1 Pengertian Pengelolaan Sampah

Pengelolaan sampah adalah serangkaian kegiatan sistematis yang mencakup pengumpulan, pemisahan, pengangkutan, pemrosesan, daur ulang, dan pembuangan akhir sampah, baik dari rumah tangga maupun industri, dengan tujuan untuk menjaga kebersihan, kesehatan masyarakat, dan kelestarian lingkungan (Peraturan Pemerintah No. 81 Tahun 2012).

Dalam konteks modern, pengelolaan sampah tidak hanya dipandang sebagai proses logistik, tetapi juga sebagai sistem terpadu berbasis data dan teknologi. Abdul et al. (2024) menjelaskan bahwa pengelolaan sampah berbasis mikrokontroler dan sensor dapat meningkatkan akurasi pemantauan volume dan berat sampah, serta mempermudah visualisasi data melalui platform berbasis web secara *real-time*. Dengan demikian, pengelolaan sampah berbasis teknologi menghadirkan efisiensi, akuntabilitas, serta kemampuan untuk mengambil keputusan berbasis data (*data-driven decision making*).

2.1.1.2 Permasalahan pada Pengelolaan Sampah Konvensional

Meskipun pengelolaan sampah merupakan fungsi vital dalam sistem perkotaan, pendekatan konvensional masih menghadapi berbagai kendala signifikan. Di antaranya adalah:

1. Keterlambatan Pengangkutan dan Penumpukan Sampah

Minimnya informasi kondisi tempat sampah menyebabkan penjadwalan pengangkutan yang tidak optimal. Hal ini berakibat pada penumpukan sampah

yang berdampak pada pencemaran lingkungan dan kesehatan masyarakat (Bekti et al., 2020).

2. Risiko Kesehatan Akibat Kontak Langsung

Sistem konvensional seringkali mengharuskan masyarakat membuka tutup tempat sampah secara manual, yang meningkatkan risiko paparan terhadap bakteri, virus, dan bau tidak sedap (Armanda et al., 2021).

3. Inefisiensi Biaya Dan Tenaga

Armada pengangkut sampah sering kali bergerak tanpa arah atau data yang jelas, sehingga menyebabkan pemborosan bahan bakar dan waktu tempuh, terutama bila lokasi tempat sampah yang dituju ternyata belum penuh (Bekti et al., 2020).

Dengan demikian, pendekatan konvensional cenderung bersifat reaktif dan tidak adaptif terhadap kondisi real-time di lapangan, yang menurunkan efisiensi operasional pengelolaan sampah.

2.1.1.3 Solusi Otomatisasi dalam Pengelolaan Sampah

Solusi berbasis otomatisasi hadir sebagai alternatif inovatif terhadap berbagai kelemahan sistem konvensional. Sistem pengelolaan sampah otomatis menawarkan beberapa pendekatan teknologi kunci, di antaranya:

1. Sensorisasi Tempat Sampah

Penggunaan sensor ultrasonik untuk mendeteksi ketinggian sampah, serta sensor load cell untuk mengukur berat sampah. Abdul et al. (2024) mencatat bahwa sensor ultrasonik memiliki akurasi hingga 100% dalam pengujian deteksi volume sampah.

2. Aktuator Buka-Tutup Otomatis

Mekanisme servo motor diintegrasikan dengan sensor gerak (seperti HC-SR04) untuk membuka tutup tempat sampah secara otomatis ketika objek mendekat, sehingga mengurangi kontak langsung pengguna (Armanda et al., 2021).

3. Pemantauan Berbasis Iot dan Cloud Platform

Sistem yang terhubung dengan Firebase atau Thingspeak memungkinkan pengiriman data *real-time* ke dashboard web atau aplikasi mobile. Ini memudahkan pemantauan kondisi tempat sampah dari jarak jauh, serta notifikasi jika sudah penuh (Abdul et al., 2024; Armanda et al., 2021).

4. Integrasi Dengan Sistem Pemetaan Digital

Google Maps API digunakan untuk menampilkan lokasi tempat sampah dan mengarahkan petugas melalui rute terpendek (Bekti et al., 2020). Hal ini mempercepat proses pengambilan sampah dan menghemat waktu tempuh armada.

Dengan menerapkan teknologi tersebut, sistem otomatisasi pengelolaan sampah mampu menjawab tantangan konvensional secara efektif dan mendukung transisi menuju sistem pengelolaan kota yang lebih cerdas, efisien, dan berkelanjutan.

2.1.2 Teknologi Internet of Things

2.1.2.1 Pengertian dan Manfaat IoT

Internet of thing (IoT) merupakan suatu konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus menerus. Internet of thing (IoT) bisa dimanfaatkan pada gedung untuk

mengendalikan peralatan elektronik seperti lampu ruangan yang dapat dioperasikan dari jarak jauh melalui jaringan komputer, tidak dapat dipungkiri kemajuan teknologi yang sedemikian dimanfaatkan, dipelajari cepat serta harus diterapkan.

Menurut Arafat (2016), Internet of Things atau dikenal juga dengan singkatan IoT, merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus yang memungkinkan kita untuk menghubungkan mesin, peralatan, dan benda fisik lainnya dengan sensor jaringan dan aktuator untuk memperoleh data dan mengelola kinerjanya sendiri, sehingga memungkinkan mesin untuk berkolaborasi dan bahkan bertindak berdasarkan informasi baru yang diperoleh secara independen.

Cara Kerja Internet of Things yaitu dengan memanfaatkan sebuah argumentasi pemrograman yang dimana tiap-tiap perintah argumennya itu menghasilkan sebuah interaksi antara sesama mesin yang terhubung secara otomatis tanpa campur tangan manusia dan dalam jarak berapa pun. Internetlah yang menjadi penghubung di antara kedua interaksi mesin tersebut, sementara manusia hanya bertugas sebagai pengatur dan pengawas bekerjanya alat tersebut secara langsung.

Menurut Abdul et al. (2024), IoT dapat memberikan manfaat nyata dalam hal:

1. Pemantauan *real-time* seperti kondisi lingkungan, termasuk status sampah, dapat diamati secara langsung melalui jaringan.
2. Efisiensi operasional seperti proses manual digantikan oleh sistem otomatis yang responsif terhadap data.
3. Pengurangan risiko kesehatan karena interaksi manusia dapat diminimalkan.

4. Optimasi sumber daya seperti pengangkutan atau pengelolaan dapat dijadwalkan secara cerdas berdasarkan kondisi yang sebenarnya di lapangan.
5. Aksesibilitas data melalui aplikasi web atau mobile, informasi dapat diakses dari mana saja.

Dalam implementasi smart trash, IoT berperan dalam menghubungkan sensor ketinggian, beban sampah, dan posisi GPS ke server cloud seperti Firebase, sehingga data dapat dikumpulkan, disimpan, dan dianalisis untuk membantu proses pengambilan keputusan.

2.1.2.2 Arsitektur dan Jenis Perangkat IoT

Arsitektur sistem IoT secara umum terdiri atas empat lapisan utama:

1. *Perception Layer* (Lapisan Persepsi)

Lapisan ini bertugas mengumpulkan informasi dari lingkungan fisik menggunakan sensor dan perangkat input seperti kamera, RFID, GPS, dan sensor ultrasonik.

2. *Network Layer* (Lapisan Jaringan)

Lapisan ini bertanggung jawab untuk mentransmisikan data dari perception layer ke pusat pemrosesan data melalui jaringan Wi-Fi, GSM, LoRa, atau NB-IoT.

3. *Middleware Layer* (Lapisan Perantara)

Berfungsi sebagai pengelola data, lapisan ini biasanya berupa cloud service (misalnya: Firebase, Blynk, Thingspeak) yang menerima, menyimpan, dan meneruskan data ke aplikasi.

4. *Application Layer* (Lapisan Aplikasi)

Merupakan antarmuka pengguna (*user interface*) di mana data dari sistem dapat diakses, dianalisis, atau dimonitor oleh pengguna, biasanya melalui website atau aplikasi mobile (Bekti et al., 2020; Armanda et al., 2021).

Selanjutnya, jenis perangkat IoT terbagi atas:

1. Sensor dan Aktuator seperti HC-SR04 (ultrasonik), load cell, DHT11, dan motor servo.
2. Perangkat komunikasi seperti ESP8266, GSM module, WiFi module.
3. Platform cloud dan visualisasi seperti Google Firebase, Google Maps API, Blynk, Thingspeak.
4. Antarmuka pengguna (UI) seperti aplikasi mobile, dashboard berbasis web.

2.1.2.3 Microcontroller, Sensor dan Aktuator pada Sistem IoT

Dalam sistem IoT, komponen utama yang berperan dalam interaksi antara dunia fisik dan digital terdiri dari mikrokontroler, sensor, dan aktuator.

1. Mikrokontroler

Mikrokontroler merupakan otak dari sistem IoT yang bertugas memproses input dari sensor, menjalankan logika program, dan mengendalikan keluaran melalui aktuator. NodeMCU ESP8266 menjadi pilihan populer karena memiliki kemampuan Wi-Fi terintegrasi, hemat daya, serta mendukung komunikasi dengan platform cloud (Armanda et al., 2021).

2. Sensor

Sensor merupakan komponen yang bertugas mengubah fenomena fisik (seperti jarak, suhu, atau tekanan) menjadi sinyal listrik. Dalam konteks smart waste system,

sensor HC-SR04 digunakan untuk mengukur jarak atau ketinggian sampah dalam tempat sampah. Load cell juga digunakan untuk mendeteksi berat sampah secara presisi (Abdul et al., 2024).

3. Aktuator

Aktuator berfungsi mengubah sinyal listrik dari mikrokontroler menjadi gerakan fisik. Motor servo digunakan sebagai aktuator untuk membuka atau menutup tutup tempat sampah secara otomatis saat sensor mendeteksi keberadaan objek pada jarak tertentu. Aktuator ini memungkinkan sistem bekerja tanpa keterlibatan tangan manusia dan memperbaiki aspek higienitas (Armanda et al., 2021).

2.1.3 Langkah Pengembangan Smart Waste Management System

Pengembangan sistem *Smart Waste Management* membutuhkan pendekatan rekayasa perangkat lunak yang iteratif dan fleksibel. Hal ini dikarenakan sistem melibatkan komponen hardware (sensor, mikrokontroler, aktuator) serta komponen software (aplikasi web, sistem cloud, antarmuka pengguna). Untuk menjamin kehandalan dan kemudahan adaptasi dalam pengembangan, digunakan pendekatan metode prototyping dan pemodelan sistem menggunakan Unified Modeling Language (UML).

Menurut Abdul et al. (2024), proses perancangan sistem monitoring tempat sampah berbasis IoT yang terintegrasi dengan Google Maps membutuhkan tahapan-tahapan iteratif berupa rancangan cepat, pengujian awal, perbaikan, dan implementasi akhir. Hal ini sejalan dengan prinsip metode prototype yang

memungkinkan interaksi langsung dengan pengguna sebelum sistem dikembangkan sepenuhnya.

2.1.3.1 Metode Prototype

Metode prototyping adalah salah satu model dalam pengembangan sistem yang berfokus pada pembuatan versi awal atau mock-up sistem secara cepat untuk diuji, dinilai, dan disempurnakan berdasarkan umpan balik pengguna. Menurut Pressman (2014), metode ini sangat sesuai untuk proyek yang belum memiliki kebutuhan sistem yang lengkap dan stabil di awal pengembangan.

Adapun tahapan metode prototype secara umum adalah sebagai berikut:

1. Identifikasi Kebutuhan Awal

Tim merancang spesifikasi dasar sistem berdasarkan kebutuhan umum seperti fitur buka-tutup otomatis, pengukuran sampah, dan pengiriman data ke server.

2. Perancangan Prototipe

Rancang bangun sistem awal dilakukan dengan menyusun alur logika sistem serta merakit komponen sensor dan mikrokontroler.

3. Evaluasi Pengguna

Prototipe diuji oleh calon pengguna atau evaluator untuk memperoleh umpan balik awal.

4. Penyempurnaan Sistem

Berdasarkan hasil evaluasi, sistem diperbaiki dan disesuaikan agar mendekati sistem final.

5. Pengembangan Akhir

Sistem dikembangkan lebih lanjut dan diuji pada lingkungan operasional sebenarnya.

Pendekatan ini terbukti efektif untuk sistem berbasis IoT karena memungkinkan evaluasi langsung terhadap performa hardware dan software sejak tahap awal (Armanda et al., 2021).

2.1.4 Pemodelan Sistem UML

Unified Modeling Language (UML) merupakan bahasa pemodelan standar yang digunakan dalam rekayasa perangkat lunak untuk memvisualisasikan, merancang, dan mendokumentasikan sistem. UML membantu pengembang dalam memahami struktur sistem, alur logika, dan relasi antar komponen.

Dalam konteks pengembangan *Smart Waste Management System*, beberapa diagram UML yang umum digunakan antara lain:

1. *Use Case Diagram*

Menggambarkan relasi antara aktor (misalnya: pengguna, petugas kebersihan) dan fungsionalitas sistem (misalnya: memantau status sampah, membuka tutup otomatis, membaca data sensor). Diagram ini menunjukkan kebutuhan sistem secara fungsional.

2. *Activity Diagram*

Menjelaskan alur aktivitas atau proses dalam sistem, seperti proses pembacaan data sensor, pengiriman data ke Firebase, hingga pengambilan keputusan buka-tutup tutup otomatis.

3. *Sequence Diagram*

Menjelaskan urutan interaksi antar komponen atau objek dalam sistem secara kronologis, seperti alur data dari sensor ke mikrokontroler, kemudian ke cloud, dan ditampilkan di dashboard aplikasi.

Penggunaan UML sangat membantu dalam mendokumentasikan proses pengembangan sistem berbasis IoT agar sistem yang dibangun bersifat modular, terstruktur, dan mudah dikembangkan lebih lanjut.

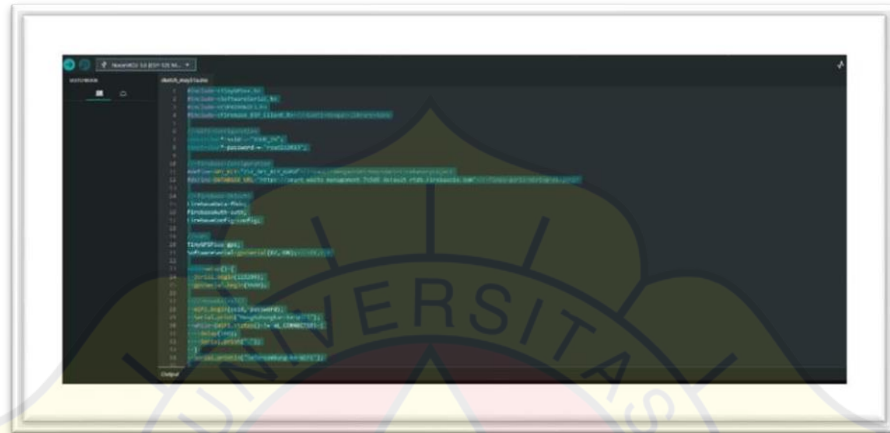
2.1.5 Software dan Pemrograman Terkait

Fokus utama adalah pada Arduino IDE sebagai lingkungan pengembangan terpadu, serta detail pemrograman untuk berbagai komponen perangkat keras yang esensial, seperti servo, sensor ultrasonik HC-SR04, sensor DHT11, dan modul GPS Ublox Neo 7M.

2.1.5.1 Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah perangkat lunak sumber terbuka yang digunakan untuk menulis, mengkompilasi, dan mengunggah kode ke papan mikrokontroler Arduino. Lingkungan ini dirancang untuk kemudahan penggunaan, menjadikannya pilihan populer bagi pemula maupun pengembang berpengalaman dalam proyek elektronika dan otomasi. Fitur utamanya meliputi editor teks untuk menulis kode dalam bahasa pemrograman berbasis C/C++ yang disederhanakan, kompilator untuk mengubah kode menjadi format yang dapat dipahami mikrokontroler, dan pengunggah untuk mentransfer kode ke papan Arduino melalui port USB. Arduino IDE juga menyediakan serial monitor yang berguna untuk *debugging* dan melihat data *output* dari mikrokontroler secara

real-time. Keberadaan library (pustaka) yang kaya memungkinkan pengguna untuk dengan mudah mengintegrasikan berbagai sensor, aktuator, dan modul komunikasi tanpa perlu menulis kode dari awal, sehingga mempercepat proses pengembangan (Kurniawan & Syahputra, 2021; Utomo et al., 2023).



Gambar 2.1 Tampilan Arduino IDE

2.1.5.2 Pemrograman Servo

Servo motor adalah aktuator yang mampu bergerak dan mempertahankan posisi sudutnya dengan presisi. Pemrograman servo pada Arduino umumnya memanfaatkan library Servo.h yang telah tersedia. Library ini menyederhanakan kontrol servo dengan menyediakan fungsi seperti `attach()` untuk menginisialisasi pin yang terhubung ke servo, dan `write()` untuk menentukan sudut posisi servo dalam derajat (biasanya antara 0 hingga 180 derajat). Prinsip kerjanya melibatkan sinyal Pulse Width Modulation (PWM) yang dikirimkan dari mikrokontroler. Lebar pulse sinyal PWM inilah yang menentukan posisi sudut servo; lebar pulse yang berbeda akan menyebabkan servo berputar ke posisi yang berbeda. Akurasi dalam pengaturan lebar pulse sangat krusial untuk mencapai kontrol posisi yang tepat (Putra & Mulyadi, 2020; Setiawan & Yanto, 2021).

Contoh kode:

```
#include <Servo.h>

Servo tutup;

void setup() {
  tutup.attach(9);
}

void loop() {
  tutup.write(0); // posisi tertutup
  delay(1000);
  tutup.write(90); // posisi terbuka
  delay(1000);
}
```

2.1.5.3 Pemrograman Sensor Ultrasonik HCSR04

Sensor ultrasonik HC-SR04 digunakan untuk mengukur jarak berdasarkan prinsip pemantulan gelombang suara. Sensor ini memiliki dua pin utama untuk operasi: Trig (trigger) untuk memancarkan gelombang suara, dan Echo untuk menerima pantulan gelombang suara. Pemrograman sensor ini pada Arduino melibatkan langkah-langkah berikut: pertama, mengirimkan pulsa singkat (sekitar 10 mikrosekon) ke pin Trig untuk memicu pancaran gelombang ultrasonik. Kedua, mengukur durasi waktu yang dibutuhkan gelombang suara untuk pergi dan kembali (dari saat Trig dipicu hingga Echo menerima pantulan) menggunakan fungsi `pulseIn()`. Jarak kemudian dihitung dengan rumus:

$$\text{Jarak (cm)} = \frac{\text{Durasi waktu (mikrodetik)} \times \text{Kecepatan suara (cm/mikrodetik)}}{2}$$

Kecepatan suara di udara pada suhu ruangan adalah sekitar 0.034 cm/mikrodetik (atau 340 m/detik), sehingga rumus sering disederhanakan menjadi:

$$\text{Jarak (cm)} = \frac{\text{Durasi waktu (mikrodetik)}}{58}$$

Gambar 2.2 Rumus Jarak

Pemrograman yang tepat memastikan pengukuran jarak yang akurat, menjadikannya komponen penting dalam aplikasi deteksi objek atau penghindaran halangan (Pramana et al., 2022; Ramadhan & Darmawan, 2022).

Contoh kode:

```
#define trigPin 7
#define echoPin 6

long durasi;
int jarak;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(trigPin, LOW); delayMicroseconds(2);
  digitalWrite(trigPin, HIGH); delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  durasi = pulseIn(echoPin, HIGH);
```

```

jarak = durasi * 0.034 / 2;

Serial.println(jarak);

}

```

2.1.5.4 Pemrograman Sensor DHT 11

Sensor DHT11 adalah sensor digital yang mampu mengukur suhu dan kelembaban lingkungan. Sensor ini berkomunikasi dengan mikrokontroler melalui satu pin data, menggunakan protokol komunikasi satu kabel yang spesifik. Untuk memudahkan pemrograman, library DHT.h sering digunakan. Library ini menyediakan fungsi untuk membaca data suhu dan kelembaban dari sensor secara langsung. Setelah inisialisasi sensor dan pin yang terhubung, pembacaan data dilakukan dengan memanggil fungsi seperti `readTemperature()` dan `readHumidity()`. Sensor DHT11 dikenal karena biayanya yang rendah dan kemudahan penggunaannya, menjadikannya pilihan populer untuk proyek-proyek monitoring lingkungan (Nurdiansyah et al., 2021; Sari et al., 2021).

Contoh kode:

```

#include <DHT.h>

#define DHTPIN 5

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

void setup() {

  Serial.begin(9600);

  dht.begin();

}

```

```

void loop() {

  float suhu = dht.readTemperature();

  float kelembapan = dht.readHumidity();

  Serial.print("Suhu: "); Serial.println(suhu);

  Serial.print("Kelembapan: "); Serial.println(kelembapan);

}

```

2.1.5.5 Pemrograman Modul GPS Ublox Neo 7M

Modul GPS Ublox Neo 7M adalah perangkat penerima GPS yang digunakan untuk mendapatkan informasi lokasi (lintang, bujur, ketinggian), waktu, dan kecepatan. Komunikasi antara modul GPS dan mikrokontroler Arduino umumnya dilakukan melalui komunikasi serial (UART). Data yang diterima dari modul GPS adalah dalam format NMEA (National Marine Electronics Association), yang merupakan serangkaian *string* teks berisi informasi geografis. Untuk memarsing data NMEA dan mengekstrak informasi yang relevan, library TinyGPS++ atau NeoGPS sangat dianjurkan. Library ini menyederhanakan proses pembacaan dan analisis *string* NMEA, memungkinkan pengembang untuk dengan mudah mengakses data lintang, bujur, tanggal, waktu, dan informasi satelit lainnya. Akurasi data lokasi sangat bergantung pada jumlah satelit yang terdeteksi dan kondisi lingkungan penerimaan sinyal (Amalia et al., 2022; Prasetyo et al., 2021).

Contoh kode:

```

#include <SoftwareSerial.h>

#include <TinyGPSPlus.h>

SoftwareSerial gpsSerial(4, 3);

```

```
TinyGPSPlus gps;

void setup() {

Serial.begin(9600);

  gpsSerial.begin(9600);

}

void loop() {

  while (gpsSerial.available() > 0) {

    gps.encode(gpsSerial.read());

    if (gps.location.isUpdated()) {

      Serial.print("Latitude: "); Serial.println(gps.location.lat(), 6);

      Serial.print("Longitude: "); Serial.println(gps.location.lng(), 6);

    }

  }

}
```

2.1.6 Platform IoT dan Komponen Pendukung

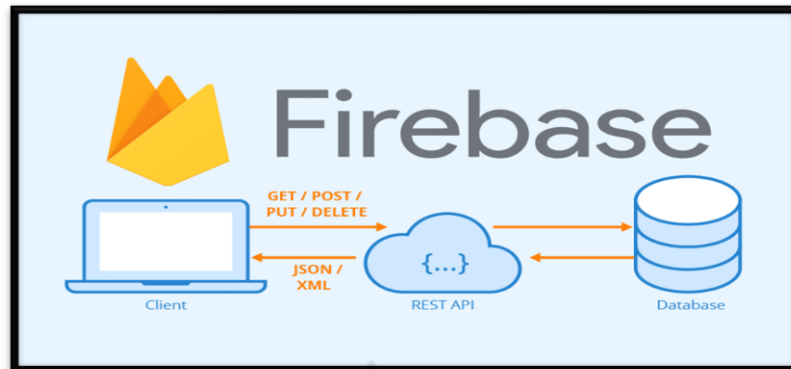
2.1.6.1 Database pada Firebase

Database pada Firebase merupakan solusi basis data *NoSQL* yang dikembangkan oleh Google, menawarkan dua opsi utama: Cloud Firestore dan Realtime Database. Keduanya dirancang untuk mendukung aplikasi *real-time* dan *serverless*, menjadikannya sangat relevan untuk proyek IoT yang memerlukan sinkronisasi data yang cepat dan efisien antara perangkat, *cloud*, dan aplikasi pengguna.

Cloud Firestore adalah basis data dokumen yang fleksibel dan skalabel. Data disimpan dalam dokumen yang terorganisir dalam koleksi, memungkinkan struktur data yang lebih teratur dan query yang lebih kompleks dibandingkan Realtime Database. Firestore sangat cocok untuk aplikasi yang membutuhkan fitur *offline-first*, *scalability* global, dan query yang kuat. Mekanisme sinkronisasi *real-time*nya memastikan bahwa setiap perubahan data secara instan direfleksikan di semua klien yang terhubung, meminimalkan latensi dan meningkatkan responsivitas sistem IoT (Google Cloud, n.d.; Putra et al., 2021).

Sementara itu, Realtime Database adalah basis data *JSON tree* yang memungkinkan sinkronisasi data *real-time* ke semua klien yang terhubung. Meskipun strukturnya lebih sederhana (mirip *JSON tree* besar), Realtime Database sangat efisien untuk kasus penggunaan di mana perubahan data terjadi sangat cepat dan diperlukan *event-driven updates* secara instan. Ini ideal untuk aplikasi IoT yang mengalirkan data sensor secara terus-menerus atau memerlukan notifikasi *real-time* yang cepat (Google Cloud, n.d.; Huda et al., 2020).

Kedua jenis database ini menyediakan SDK (Software Development Kit) yang kuat untuk berbagai platform, termasuk web, Android, dan iOS, memfasilitasi integrasi yang mulus dengan aplikasi *frontend* dan *backend*. Fitur otentikasi dan aturan keamanan yang dapat dikonfigurasi pada Firebase memungkinkan kontrol akses data yang ketat, memastikan integritas dan kerahasiaan data IoT.



Gambar 2.3 Struktur Pengiriman Data ke Firebase

2.1.6.1 Firebase

Firebase adalah platform pengembangan aplikasi *mobile* dan web yang komprehensif dari Google, menyediakan berbagai layanan *backend* yang dapat digunakan untuk membangun aplikasi dengan cepat tanpa perlu mengelola infrastruktur *server* secara langsung. Dalam konteks IoT, Firebase menawarkan ekosistem yang kuat untuk mengelola perangkat, menyimpan data, mengotentikasi pengguna, dan memfasilitasi komunikasi *real-time* (Google Cloud, Saputra et al., 2022).

Komponen kunci Firebase yang relevan untuk proyek IoT meliputi:

1. *Authentication*

Menyediakan layanan otentikasi yang aman dan mudah diintegrasikan, mendukung berbagai metode *login* seperti email/password, Google, Facebook, dan lainnya. Ini penting untuk mengamankan akses ke data IoT dan mengelola hak akses pengguna.

2. *Hosting*

Memungkinkan *deployment* aplikasi web statis atau *single-page applications* dengan cepat dan aman melalui *Content Delivery Network* (CDN) global. Ini

dapat digunakan untuk *dashboard* IoT berbasis web yang menampilkan data sensor.

3. *Cloud Functions*

Memungkinkan eksekusi kode *backend* yang *serverless* sebagai respons terhadap peristiwa tertentu (misalnya, perubahan data di database, HTTP requests). *Cloud Functions* sangat berguna untuk memproses data IoT, memicu notifikasi, atau mengintegrasikan dengan layanan pihak ketiga tanpa perlu *server* fisik.

4. *Firebase Cloud Messaging (FCM)*

Layanan pengiriman pesan *cross-platform* yang memungkinkan pengiriman notifikasi ke perangkat *mobile* dan aplikasi web. FCM krusial untuk mengirimkan peringatan atau status dari sistem IoT ke pengguna secara *real-time*.

5. *Storage*

Menyediakan penyimpanan objek yang kuat dan skalabel untuk menyimpan file seperti gambar, video, atau *log* data dari perangkat IoT.

Dengan mengintegrasikan layanan-layanan ini, pengembang dapat fokus pada logika aplikasi inti dan interaksi perangkat IoT, sementara Firebase menangani kompleksitas infrastruktur *backend* yang mendasari. Fleksibilitas, skalabilitas, dan kemudahan penggunaan Firebase menjadikannya pilihan yang sangat efektif untuk membangun solusi IoT yang tangguh dan modern (Huda et al., 2020; Saputra et al., 2022).

2.2 Kajian Penelitian Terdahulu

2.2.1 Paper 1

Judul: *Rancang Bangun Sistem Monitoring Smart Trash Berbasis Website Terintegrasi Dengan Google Maps*

Author: Abdul, Tamam, dan Nuzul

Publikasi: 2024

Klasifikasi Jurnal: Jurnal Nasional – SINTA 4

2.2.1.1 Tujuan Penelitian

Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem monitoring tempat sampah pintar berbasis mikrokontroler yang terintegrasi dengan Google Maps. Sistem ini dimaksudkan untuk mendeteksi kondisi tempat sampah secara real-time dan dapat dimonitor melalui platform website.

2.2.1.2 Metodologi yang Digunakan

Metode yang digunakan adalah perancangan sistem berbasis mikrokontroler dengan integrasi sensor ultrasonik, load cell, dan GPS Neo 6M. Platform Thingspeak digunakan sebagai media tampilan data, yang dapat diakses melalui aplikasi berbasis website. Sistem diuji untuk mengukur akurasi masing-masing sensor dan kemampuan monitoring secara real-time.

2.2.1.3 Temuan Utama

Sensor ultrasonik menunjukkan akurasi deteksi volume sampah sebesar 100%. Sensor loadcell hanya memiliki akurasi sebesar 6% untuk berat sampah. GPS Neo 6M menghasilkan akurasi lokasi dengan tingkat error sebesar 2,997 meter.

Sistem secara umum berhasil menampilkan data volume, berat, dan lokasi sampah ke dalam dashboard website secara real-time.

2.2.1.4 Kesimpulan

Sistem yang dirancang mampu menjadi solusi efektif dalam mendeteksi dan memantau kondisi tempat sampah. Penggunaan integrasi dengan Google Maps dan platform Thingspeak memudahkan pengelolaan serta monitoring sampah secara efisien, sehingga dapat membantu mengurangi penumpukan sampah dan meningkatkan kenyamanan lingkungan.

2.2.2 Paper 2

Judul: *Aplikasi Penentuan Rute Terpendek Pengambilan Sampah Berbasis Internet of Things Terintegrasi Dengan Google Maps API*

Author: Bekti M., Agus H., M. Ibnu, dan Sisca

Publikasi: 2020

Klasifikasi Jurnal: Jurnal Nasional – SINTA 4

2.2.2.1 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk memantau tingkat kepenuhan tempat sampah secara real-time dan menampilkan lokasi tempat sampah yang penuh pada peta menggunakan Google Maps API. Selain itu, penelitian ini juga bertujuan menentukan rute terpendek pengambilan sampah berdasarkan data kepenuhan tersebut.

2.2.2.2 Metodologi yang Digunakan

Penelitian ini menggunakan pendekatan berbasis Internet of Things (IoT) dengan menggabungkan sensor tingkat kepenuhan dan integrasi Google Maps API.

Aplikasi dikembangkan untuk menerima data dari perangkat IoT, memvisualisasikan lokasi dan status kepenuhan tempat sampah, serta menghitung rute optimal menggunakan fitur Google Maps.

2.2.2.3 Temuan Utama

Aplikasi yang dirancang dapat secara otomatis menampilkan tempat sampah yang penuh kepada pengguna dan menghitung rute tercepat untuk pengambilan berdasarkan lokasi-lokasi tersebut. Sistem bekerja secara efektif dalam membantu proses pengambilan sampah menjadi lebih efisien.

2.2.2.4 Kesimpulan

Sistem ini terbukti dapat membantu pengelola kebersihan dalam menentukan prioritas pengambilan sampah dan mempercepat proses rute perjalanan armada sampah. Penggunaan Google Maps API mendukung pengambilan keputusan secara dinamis berdasarkan kondisi lapangan.

2.2.3 Paper 3

Judul: *Pemanfaatan Google Firebase Pada Sistem Tempat Sampah Pintar Berbasis Internet of Things*

Author: Armanda S., Danny K., dan Fikra T.

Publikasi: 2021

Klasifikasi Jurnal: Jurnal Nasional – SINTA 4

2.2.3.1 Tujuan Penelitian

Penelitian ini bertujuan untuk memanfaatkan platform Google Firebase dalam membangun sistem tempat sampah pintar berbasis Internet of Things (IoT) yang mampu memantau ketinggian sampah secara real-time dari jarak jauh.

2.2.3.2 Metodologi yang Digunakan

Sistem dirancang dengan menggunakan sensor ultrasonik HC-SR04 untuk mengukur ketinggian sampah dalam wadah, dan mikrokontroler NodeMCU ESP8266 untuk mengirimkan data secara nirkabel ke Google Firebase. Firebase digunakan sebagai database berbasis cloud dengan konsep BaaS (Backend as a Service) yang mendukung pembacaan data secara real-time.

2.2.3.3 Temuan Utama

Sistem mampu mengirimkan data ketinggian sampah secara real-time dan dapat diakses dari mana saja melalui koneksi internet. Firebase terbukti mampu menjadi solusi efisien dalam menangani penyimpanan dan pengambilan data pada sistem monitoring berbasis IoT.

2.2.3.4 Kesimpulan

Pemanfaatan Google Firebase memberikan fleksibilitas dan kemudahan dalam membangun sistem pemantauan tempat sampah pintar. Dengan teknologi ini, pengguna dapat mengetahui kondisi tempat sampah tanpa harus datang langsung ke lokasi, sehingga sistem ini efektif untuk mendukung pengelolaan sampah yang lebih cerdas dan terhubung.