Jurnal Inovasi dan Kolaborasi Nusantara
https://ejurnals.com/ojs/index.php/jikn

Volume 07, No. 1, Maret 2026

SISTEM PENDETEKSI DINI KEBAKARAN PADA RUMAH MENGGUNAKAN TEKNOLOGI IOT

Ab. Ricky Handika[1], Nurman Hariyanto[2], Denny Primanda[3]

[1,2,3]Universitas Nahdlatul Ulama Kalimantan Barat

Email: abrickyhandika21@gmail.com

ABSTRAK: Kebakaran merupakan salah satu bencana yang sering terjadi di lingkungan permukiman dan menyebabkan kerugian besar baik materil maupun korban jiwa. Deteksi kebakaran yang masih dilakukan secara manual, seperti memukul kentungan atau menelepon Dinas Pemadam Kebakaran (DAMKAR), seringkali terlambat dan kurang efektif. Oleh karena itu, diperlukan sistem deteksi dini berbasis teknologi untuk meningkatkan respons terhadap potensi kebakaran. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem pendeteksi dini kebakaran pada rumah berbasis Internet of Things (IoT) menggunakan mikrokontroler NodeMCU ESP8266. Sistem ini menggunakan sensor api dan sensor asap untuk mendeteksi kebakaran, serta dilengkapi dengan GPS untuk membagikan lokasi kebakaran. Data dari sensor dikirimkan secara realtime melalui jaringan WiFi ke Firebase Realtime Database dan Firestore, lalu diteruskan ke aplikasi Android berbasis Flutter yang dirancang untuk dua pengguna utama: pemilik rumah dan pihak DAMKAR. Sistem akan mengirimkan notifikasi dan lokasi kejadian secara langsung kepada keduanya jika terjadi indikasi kebakaran. Metode pengembangan sistem yang digunakan dalam penelitian ini adalah Waterfall, yang mencakup tahapan analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Pengujian dilakukan terhadap perangkat keras dan perangkat lunak untuk memastikan akurasi sistem dalam mendeteksi kebakaran dan mengirimkan notifikasi secara realtime. Hasil pengujian menunjukkan bahwa sistem berhasil mendeteksi keberadaan api dan asap, serta mengirimkan notifikasi disertai lokasi secara akurat ke aplikasi Android. Sistem ini berpotensi memberikan solusi efektif dalam mendukung upaya mitigasi kebakaran di lingkungan rumah tinggal.
Kata Kunci: Iot, Kebakaran, NodeMCU ESP8266, Sensor Api, Sensor Asap, *Firebase*, Android, Notifikasi, *Flutter*.

ABSTRACT: *Fire is a frequent disaster in residential areas and causes significant losses, both material and human casualties. Manual fire detection, such as banging a wooden gong or calling the Fire Department (DAMKAR), is often delayed and ineffective. Therefore, a technology-based early detection system is needed to improve the response to potential fires. This study aims to design and implement an Internet of Things (IoT)-based home fire early detection system using a NodeMCU ESP8266 microcontroller. This system uses fire and smoke sensors to detect fires, and is equipped with GPS to share the location of the fire. Data from the sensors is sent in real time via a WiFi network to the Firebase Realtime Database and Firestore, then forwarded to a Flutter-based Android application designed for two main users: the homeowner and the DAMKAR. The system will send notifications and the location of the incident directly to both if there is an indication of a fire. The system development method used in this study is the Waterfall*

method, which includes the stages of requirements analysis, design, implementation, testing, and maintenance. Testing was carried out on hardware and software to ensure the accuracy of the system in detecting fires and sending notifications in real time. Test results show that the system successfully detects fire and smoke and sends accurate location notifications to an Android application. This system has the potential to provide an effective solution to support fire mitigation efforts in residential environments.
Keywords: *IoT, Fire, NodeMCU ESP8266, Fire Sensor, Smoke Sensor, Firebase, Android, Notifications, Flutter.*

A. PENDAHULUAN

Kebakaran merupakan salah satu bencana yang memiliki dampak serius terhadap kehidupan, baik dari sisi keselamatan jiwa, kerugian material, maupun lingkungan. "Terdapat 17.768 kasus kebakaran yang tercatat sepanjang tahun 2021 di Indonesia", yang sebagian besar terjadi di lingkungan permukiman, pasar, dan gedung-gedung umum. Kebakaran rumah umumnya disebabkan oleh berbagai faktor seperti kebocoran gas, korsleting listrik, hingga kelalaian manusia seperti puntung rokok yang masih menyala, penggunaan lilin, atau kompor yang ditinggalkan tanpa pengawasan.

Kesiapsiagaan masyarakat terhadap potensi kebakaran di rumah saat ini masih tergolong rendah. Umumnya, masyarakat baru bereaksi setelah api membesar, dengan cara memukul kentongan, berteriak, atau menghubungi Dinas Pemadam Kebakaran (DAMKAR) secara manual. Cara ini sangat bergantung pada kehadiran manusia dan kesadaran situasional, sehingga tidak efektif dalam mencegah kebakaran sejak awal. Sistem notifikasi berbasis Telegram, email, atau *website* yang sudah dikembangkan pun masih memiliki keterbatasan, karena hanya mengirimkan informasi kepada pengguna dan tidak secara langsung kepada pihak terkait seperti DAMKAR. Selain itu, notifikasi berbasis *email* dan *website* juga kurang *responsif* karena pengguna tidak selalu aktif memantau notifikasi tersebut.

Untuk menjawab permasalahan tersebut, teknologi *Internet of Things* (IoT) menjadi solusi yang menjanjikan. NodeMCU ESP8266, sebagai *mikrokontroler* berbasis IoT, memiliki kemampuan untuk mengintegrasikan berbagai sensor dan mengirimkan data secara *real-time* melalui koneksi WiFi. "penggunaan teknologi IoT menawarkan solusi dengan kemampuan pemantauan *real-time*, deteksi cerdas, dan *responsif*". Dengan menggunakan kombinasi sensor asap (MQ2), sensor api (*flame sensor*), serta modul GPS

dan buzzer, sistem dapat mendeteksi indikasi kebakaran sejak dini dan mengirimkan notifikasi serta lokasi kejadian kepada pengguna dan instansi DAMKAR secara otomatis.

Penelitian ini bertujuan untuk merancang dan membangun sistem pendeteksi dini kebakaran berbasis IoT pada lingkungan rumah tinggal. Sistem ini diharapkan dapat meningkatkan kecepatan respons terhadap kebakaran, mengurangi risiko kerugian, dan memberikan pemantauan secara *real-time* terhadap kondisi rumah. Selain itu, sistem ini juga dapat berperan sebagai solusi teknologi terapan yang dapat diimplementasikan dalam skala masyarakat luas.

B. METODE PENELITIAN

Penelitian ini dilaksanakan melalui dua pendekatan yang saling berkaitan, yakni proses pengumpulan data dan tahap pengembangan sistem. Pengumpulan data dilakukan untuk mengidentifikasi serta memperoleh informasi yang relevan sebagai dasar dalam penelitian. Setelah data terkumpul, langkah selanjutnya adalah merancang dan mengimplementasikan sistem yang dibangun berdasarkan hasil data tersebut.

1) Metode Pengumpulan Data

1. Observasi

Observasi dalam penelitian Sistem Pendeteksi Dini Kebakaran pada Rumah berbasis IoT dilakukan karena masih banyak masyarakat yang mengandalkan cara manual dalam mencegah kebakaran, yang dinilai kurang efektif. Penelitian ini mengumpulkan data terkait alat-alat yang digunakan untuk mendeteksi kebakaran serta sistem peringatan yang akan dikirim ke pengguna. Alat yang digunakan meliputi NodeMCU ESP8266, sensor api, sensor asap, buzzer, GPS, dan melibatkan dua pengguna (Pemilik Rumah dan Damkar). Tujuan dari pengumpulan data ini adalah untuk mendukung upaya pencegahan kebakaran besar. Observasi ini juga membantu dalam memahami kebutuhan teknis serta menentukan metode yang tepat untuk mengembangkan sistem, yang mencakup penggunaan teknologi IoT, *Firebase* untuk pengelolaan data, *Flutter* untuk pengembangan aplikasi, dan NodeMCU ESP8266 sebagai kontroler perangkat keras.

2. Studi pustaka

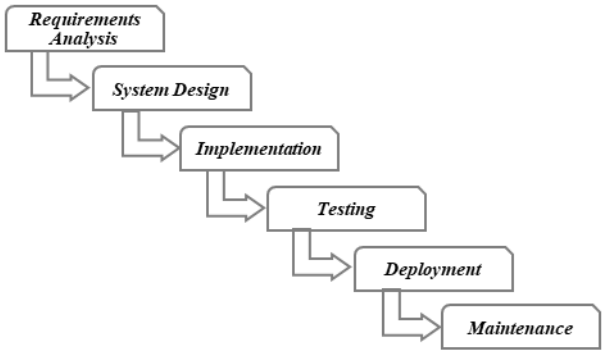
Studi pustaka merupakan kegiatan menelusuri dan menelaah berbagai sumber informasi seperti buku, jurnal, laporan, dan tulisan ilmiah lainnya yang berkaitan dengan topik penelitian. "Metode dengan pengumpulan data dengan cara memahami dan mempelajari teori-teori dari berbagai literatur yang berhubungan dengan penelitian tersebut" [5]. Tujuannya adalah untuk memahami dasar teori yang relevan, menemukan celah atau kekurangan dalam penelitian sebelumnya, menyusun metodologi yang tepat, serta menghindari pengulangan terhadap penelitian yang sudah ada.

3. Wawancara

Dalam penelitian ini, penulis melakukan wawancara langsung dengan Bapak Ade Khairul Arya Putra selaku Sekretaris Umum Dinas Pemadam Kebakaran (DAMKAR) untuk memperoleh pemahaman yang lebih mendalam tentang sistem deteksi dini kebakaran. Wawancara ini membahas prosedur DAMKAR dalam menangani laporan kebakaran, sistem yang saat ini digunakan, serta pandangan mereka terhadap penerapan teknologi IoT dalam deteksi dini. Hasil wawancara menunjukkan bahwa fitur yang dibutuhkan DAMKAR meliputi *share location*, notifikasi *real-time*, serta deteksi asap dan api. Untuk komunikasi, mereka masih mengandalkan *Handy Talkie* (HT), *WhatsApp*, dan Telepon.

2) Metode Perancangan Sistem

Metode yang digunakan pada penelitian ini adalah menggunakan metode *Software Development Life Cycle* (SDLC) Waterfall. "*Waterfall* adalah pendekatan manajemen proyek yang bersifat linier dan berurutan. Setiap tahap proyek harus diselesaikan sepenuhnya sebelum melanjutkan ke tahap berikutnya"[6]. "*Waterfall* adalah model dalam *System Development Life Cycle* (SDLC) yang digunakan untuk mengembangkan sistem informasi atau perangkat lunak secara bertahap. Model ini berjalan secara berurutan mulai dari *requirement analisis*, *system design*, *implementation*, *testing*, hingga *maintenance*" [7]. Penjelasan lebih rinci mengenai setiap tahap dalam metode *Waterfall* dapat dilihat pada gambar 1.



Gambar 1. Tahapan Metode *Waterfall*

a. *Requirement Analysis*

Requirements analysis merupakan tahap awal dalam metode *Waterfall* yang berfungsi untuk mengidentifikasi dan merumuskan kebutuhan sistem, perangkat lunak (*Software*), dan perangkat keras (*Hardware*) secara rinci untuk memastikan bahwa setiap komponen yang diperlukan telah terpenuhi sebelum melanjutkan ke tahap pengembangan selanjutnya [8]. Tujuan dari analisis kebutuhan sistem dan alat adalah untuk mengetahui, menjelaskan, dan mencatat semua kebutuhan perangkat lunak, perangkat keras, serta komponen lainnya secara rinci. Hal ini dilakukan untuk memastikan bahwa sistem yang akan dibuat sesuai dengan harapan dan kebutuhan pengguna, serta dapat berfungsi dengan baik dalam situasi yang diharapkan.

Dalam menganalisis kebutuhan fungsional untuk sistem pendeteksi dini kebakaran pada rumah menggunakan teknologi iot, terdapat dua aspek utama yang perlu diperhatikan: kebutuhan untuk aplikasi Android dan *mikrokontroler NodeMCU ESP8266* yang terhubung dengan *Firebase* sebagai *platform cloud*. *Firebase Realtime Database* menggunakan struktur *NoSQL* berbasis *JSON* yang *fleksibel* dan mendukung sinkronisasi data secara *real-time* tanpa membutuhkan *server backend*. Berbeda dengan SQL yang berbasis tabel dan memerlukan *server* untuk proses *query*, *Firebase* memungkinkan akses langsung dari klien, mendukung mode *offline*, dan cocok untuk aplikasi yang membutuhkan respon cepat [9].

Cara kerja sistem secara singkat adalah pengguna menggunakan aplikasi android untuk notifikasi kebakaran. Aplikasi ini terhubung dengan *Firebase Realtime Database* untuk memperbarui status sensor api, sensor asap dan gps serta NodeMCU ESP8266 untuk mengirim data sensor dan gps ke *Firebase* melalui koneksi wifi. NodeMCU mengirim data ke *Realtime Database* serta menggontrol pin yang terhubung ke sensor dan gps untuk

memperbahrui data ke *Firebase* secara *realtime*. Sementar itu, setiap setatus sensor aktif data kejadian akan dicatat ke dalam *Firestore* sebagai riwayat. Sistem juga akan menampilkan setatus secara *realtime* dan memberikan notifikasi jika sensor aktif. Aplikasi juga berfungsi untuk memantau perubahan setatus sensor apakah rumah dalam kondisi aman atau kebakaran. Bedasarkan kebutuhan yang telah dijabarkan, rincian kebutuhan fungsional sistem dapat dipetakan pada Tabel 1.

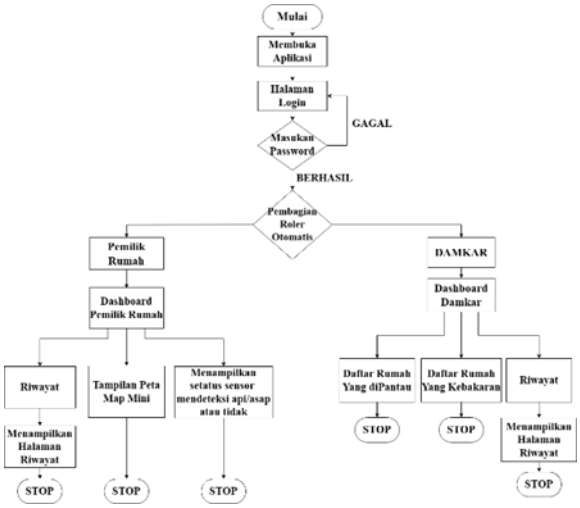
Tabel 1. Analisis Kebutuhan Fungsional Sistem dan Perangkat Keras

Kebutuhan Fungsional Aplikasi *Android*	
Notifikasi	Sistem mengirimkan notifikasi peringatan kebakaran dan *share location* ke pengguna
Monitoring api	Sistem harus dapat mengirimkan data dan menampilkan setatus adanya api di aplikasi *Android*
Monitori asap	Sistem harus dapat mengirimkan data dan menampilkan setatus asap di aplikasi *Android*
Monitorig rumah	Sistem harus dapat menampikan lokasi rumah yang aktif
Integrasi dengan *Firebase*	Aplikasi harus terhubung dengan *Firebase Realtime Database*
	Aplikasi harus mendukung pembaruhan data secara *Realtime*
Kebutuhan Fungsional Perangkat Keras	
Kontrol pin I/O	*NodeMCU* harus mengontrol pin I/O yang digunakan bedasarkan perangkat pendeteksi dan peringatan kebakaran
Kontrol api	*NodeMCU* harus mengontrol sensor api bedasarkan pin yang telah didefinisikan
Kontrol asap	*NodeMCU* harus mengontrol sensor asap dan berdasarkan pin yang telah didefinisikan
Kontrol buzzer	*NodeMCU* harus mengontrol alaram dirumah bedasarkan pin yang telah didefinisikan
Kontrol led	*NodeMCU* Harus mengontrollampu led bedasarkan pin yang telah didefinisikan

Kebutuhan Fungsional Aplikasi *Android*	
Kontrol wifi	*NodeMCU* harus terhubung jaringan internet/wifi supaya bisa kontak langsung dengan *Firebase.*

b. *System Design*

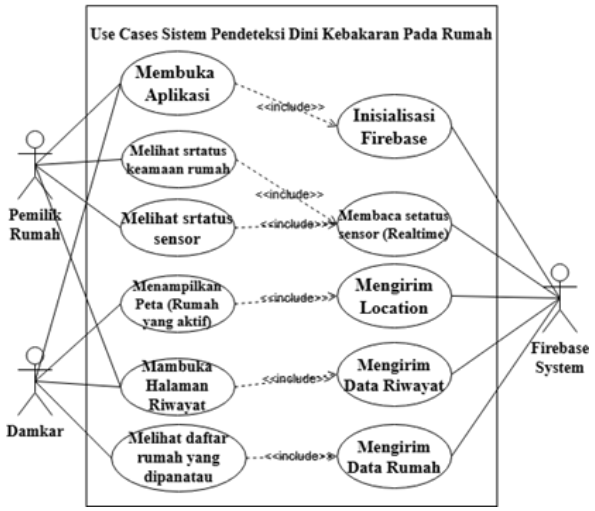
Pada tahap *System Design* dalam model *Waterfall* berperan penting dalam menghubungkan analisis kebutuhan dengan proses implementasi sistem. Pada tahap ini, hasil analisis kebutuhan diterjemahkan ke dalam rancangan teknis yang mencakup arsitektur sistem, alur proses bisnis, desain basis data, serta antarmuka pengguna [10]. Tahap desain ini meliputi pembuatan *Flowchart System*, *Use Case Diagram*, Konfigurasi *JSON* pada *Firebase*, serta Rangkaian Perangkat.



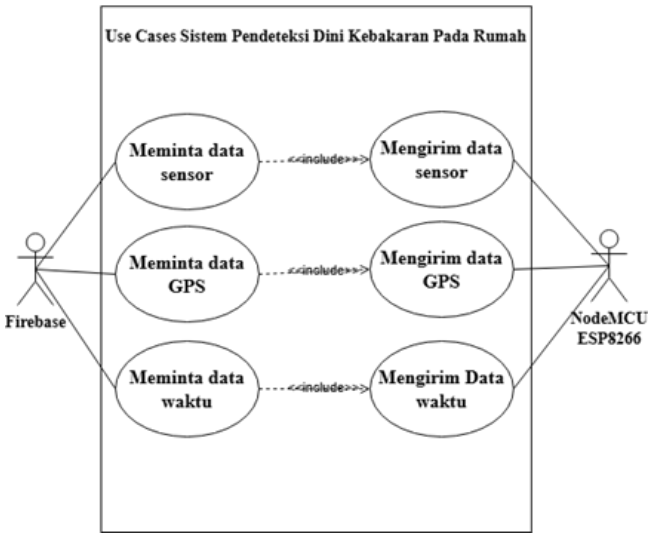
Gambar 2. *Flowchart System*

Flowchart merupakan representasi grafis yang menggambarkan urutan alur proses atau langkah-langkah dalam suatu sistem secara terstruktur. *Flowchart* menggunakan simbol-simbol standar, seperti persegi panjang untuk proses, belah ketupat untuk keputusan, dan panah untuk menunjukkan arah alur kerja. Tujuan utama penggunaan *flowchart* adalah untuk mempermudah pemahaman, analisis, dan komunikasi mengenai logika atau prosedur suatu sistem, baik untuk keperluan perancangan, dokumentasi, maupun evaluasi. Dengan *flowchart*, pengembang dan pemangku kepentingan dapat melihat gambaran menyeluruh alur proses secara jelas sehingga meminimalkan kesalahan dalam pengembangan maupun implementasi sistem.

Use Case Diagram adalah diagram dalam UML yang menggambarkan interaksi antara aktor (seperti pengguna atau sistem lain) dan sistem untuk menunjukkan fungsionalitas yang disediakan oleh sistem [14]. *Use Case* berfungsi untuk menggambarkan kebutuhan dan perilaku sistem dari sudut pandang pengguna, sehingga memudahkan identifikasi fungsionalitas dan alur interaksi sebelum sistem dirancang [15]. *Use Case Diagram* untuk sistem pendeteksi dini kebakaran pada rumah menggunakan teknologi IoT dapat dilihat pada gambar 3 dan 4.



Gambar 3. *Use Case Diagram User* dan *System*

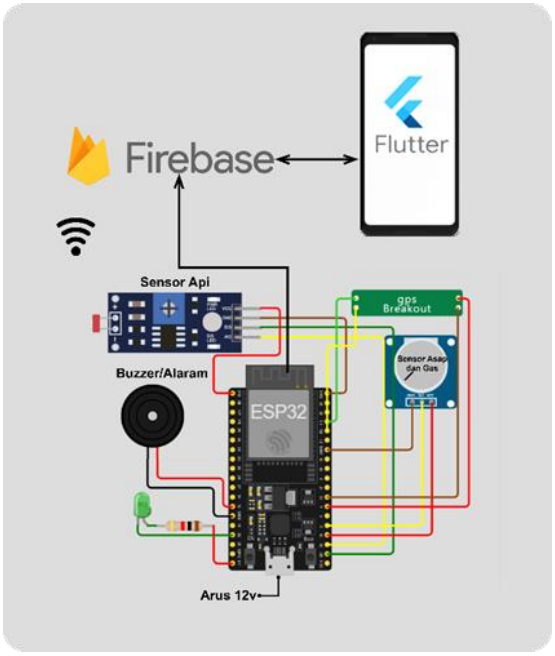


Gambar 4. Use Case System dan Mikrokontroler

Firebase Realtime Database adalah sistem basis data berbasis cloud dengan model *NoSQL* yang dirancang untuk mendukung proses penyimpanan dan sinkronisasi data antar pengguna secara *realtime*. Pada sistem ini, *Realtime Database* berfungsi untuk menyimpan data user dan memastikan perubahan data langsung tersingkronisasi dari perangkat *NodeMCU* dan Aplikasi. *Firebase Realtime Databases* mendukung perancangan serta pengelolaan struktur data menggunakan pasangan *key-value* yang disimpan dalam format *JSON*, yang digunakan pada sistem otomatisasi pendeteksi dini kebakaran berbasis teknologi IoT dapat dilihat pada gambar 5.

Gambar 5. Format Data *JSON Firebase Relatime Database*

Rangkaian perangkat merupakan susunan beberapa perangkat yang saling terhubung, baik secara berurutan maupun dalam suatu jaringan, dengan tujuan mendukung pencapaian fungsi tertentu. Susunan ini memungkinkan setiap perangkat untuk saling berinteraksi dan bekerja sama dalam menjalankan tugas yang telah ditentukan. Berikut adalah rangkaian perangkat dari sistem pendeteksi dini kebakaran pada rumah menggunakan teknologi iot pada gambar 6.



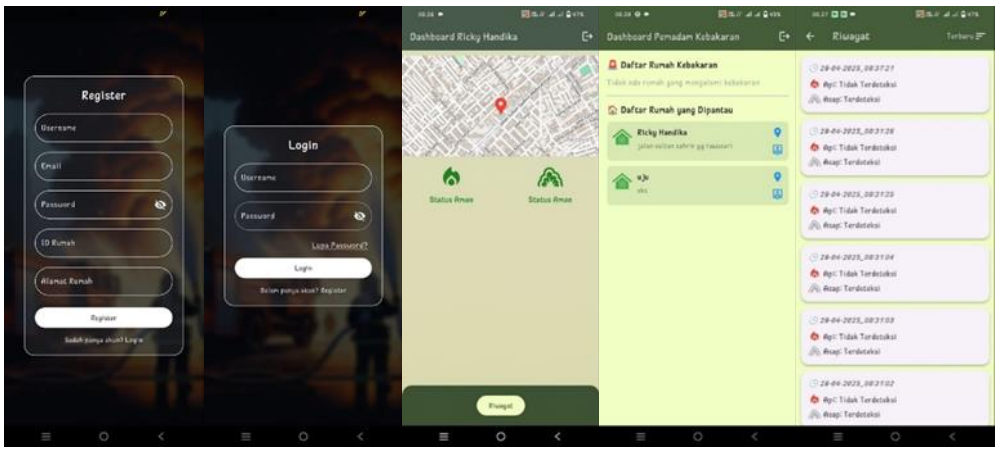
Gambar 6. Rangkaian Perangkat Keras

C. HASIL DAN PEMBAHASAN

Pada tahap implementasi, desain sistem yang telah disusun diterjemahkan ke dalam bentuk kode sumber. Setiap unit perangkat lunak dikembangkan dan diuji secara terpisah melalui uji unit untuk memastikan fungsionalitas internal komponen. Setelah semua unit selesai dan dinyatakan benar secara parsial, dilakukan integrasi komponen diikuti oleh uji integrasi guna memastikan interoperabilitas antar modul (*inter-component interaction*). Selanjutnya, dilaksanakan uji sistem secara menyeluruh untuk menilai kesesuaian sistem terhadap spesifikasi fungsional dan non-fungsional yang telah ditentukan, dan diakhiri dengan uji penerimaan oleh pengguna akhir sebagai validasi akhir terhadap kebutuhan bisnis. Setelah hasil pengujian memuaskan dan sistem mencapai stabilitas operasional, perangkat lunak dipindahkan ke lingkungan produksi (*deployment*) sehingga dapat digunakan oleh pengguna akhir. Tahap terakhir adalah pemeliharaan, yang mencakup perbaikan *bug*, pembaruan sistem, dan peningkatan fitur agar perangkat lunak tetap relevan dengan kebutuhan pengguna yang terus berubah.

1. *Implementation*

Hasil dari analisis kebutuhan dan perancangan sistem yang telah dilakukan pada tahap sebelumnya diimplementasikan menggunakan bahasa pemrograman Dart melalui pemanfaatan *framework Flutter*. Proses implementasi ini bertujuan untuk mengembangkan aplikasi Android yang berfungsi sebagai sistem pendeteksi dini kebakaran, dengan mengintegrasikan teknologi *Internet of Things* (IoT) sebagai bagian dari sistem pemantauan secara *realtime* [17]. Aplikasi ini terbagi menjadi lima halaman yang terdiri dari halaman *register*, halaman login, halaman *dashboard* pemilik rumah, halaman *dashboard* damkar, dan halaman riwayat. Berikut adalah tampilan terkait implementasi dari halaman-halaman tersebut dalam desain *UI/UX* yang di tunjukan pada gambar 7.

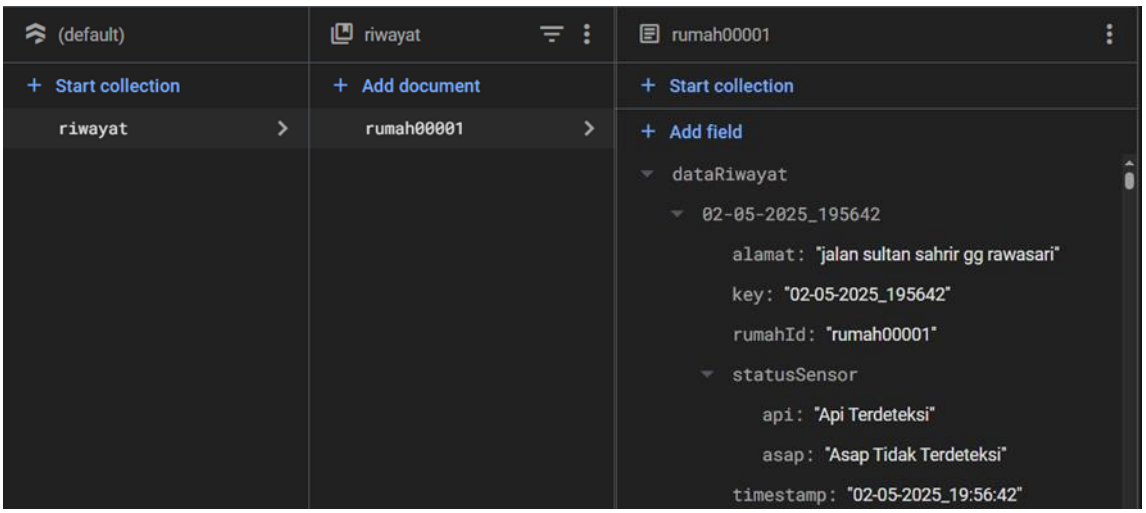


Gambar 7. Implementasi Aplikasi Sistem Pendeteksi Dini Kebakaran Pada Rumah

Aplikasi ini menggunakan layanan *Firebase Database* sebagai media untuk pengelolaan dan penyimpanan data terkait status sensor dalam sistem. Dua jenis basis data yang digunakan adalah *Firebase Realtime Database* dan *Cloud Firestore*. *Firebase Realtime Database* digunakan untuk menyimpan informasi terkait data *user* dan identitas rumah secara terstruktur dan sinkron secara *realtime*. Sementara itu, *Cloud Firestore* digunakan untuk mencatat data riwayat aktivitas sensor yang dikirim oleh perangkat NodeMCU, yang mencakup informasi waktu sensor aktif, identitas rumah yang terdeteksi mengalami kebakaran, serta alamat lokasi kejadian.

Gambar 8. *Konfigurasi Realtime Database*

Berdasarkan pada gambar 8, *Firebase Realtime Database* pada sistem ini dirancang menggunakan struktur data *JSON* yang terdiri dari dua *node* utama, yaitu rumah dan *users*. *Node* rumah menyimpan data setiap properti yang terdaftar dalam sistem, di mana masing-masing entri seperti rumah00001 memiliki sejumlah atribut penting, antara lain alamat yang merepresentasikan lokasi rumah secara tekstual (misalnya "jalan sultan sahir gg rawasan"), lokasiGPS yang ditujukan untuk menyimpan koordinat geografis rumah, pemilikrumahid yang mengidentifikasi pemilik rumah (contohnya "Ricky Handika"), serta statusSensor yang digunakan untuk mencatat kondisi terkini dari sensor kebakaran di rumah tersebut. Sementara itu, *node users* menyimpan informasi akun pengguna sistem berdasarkan *username* sebagai *key*, yang masing-masing berisi atribut seperti *email* untuk keperluan *autentikasi*, *role* untuk menentukan hak akses pengguna (seperti "pemilikrumah" atau "damkar"), uid sebagai identitas unik pengguna dari *Firebase Authentication*, serta *username* sebagai nama tampilan dalam aplikasi.



Gambar 9. Konfigurasi *Cloud Firestore*

Berdasarkan gambar 9, Pada *Cloud Firestore* digunakan koleksi utama bernama “riwayat” yang menyimpan data histori sensor dari tiap rumah. Di dalam koleksi tersebut terdapat dokumen “rumah00001” yang merepresentasikan satu unit rumah. Dokumen ini memiliki field “dataRiwayat” yang mencatat riwayat status sensor berdasarkan waktu kejadian. Salah satu entri riwayat memiliki key "02-05-2025_195642" yang menunjukkan *timestamp* data dicatat. Setiap entri memuat data “alamat” rumah, “rumahId”, dan status sensor. Status sensor terdiri dari “api” seperti "Api Terdeteksi" dan “asap” seperti "Asap Tidak Terdeteksi".

Gambar 10. Implementasi Rangkaian Perangkat Keras

Berdasarkan gambar 10, maka diperoleh rincian rangkaian perangkat keras sistem pendeteksi dini kebakaran pada rumah menggunakan teknologi IoT sebagai berikut:

1. NodeMCU ESP8266 dihubungkan ke laptop menggunakan kabel microUSB untuk memantau proses *debugging*.
2. Sensor api dihubungkan ke NodeMCU melalui setiap pin yang digunakan, yaitu:
    a. Pin DO dihubungkan ke pin D1 pada NodeMCU menggunakan kabel jumper warna biru.
    b. Pin VCC dihubungkan ke pin 3V pada NodeMCU menggunakan kabel jumper warna ungu.
    c. Pin GND dihubungkan ke pin GND pada NodeMCU menggunakan kabel jumper warna putih.
3. Sensor asap dihubungkan ke NodeMCU melalui setiap pin yang digunakan, yaitu:
    a. Pin DO dihubungkan ke pin D2 pada NodeMCU menggunakan kabel jumper warna ungu.
    b. Pin VCC dihubungkan ke pin 3V pada NodeMCU menggunakan kabel jumper warna merah.
    c. Pin GND dihubungkan ke pin GND pada NodeMCU menggunakan kabel jumper warna coklat.
4. Modul GPS dihubungkan ke NodeMCU melalui setiap pin yang digunakan, yaitu:
    a. Pin RX dihubungkan ke pin D5 pada NodeMCU menggunakan kabel jumper warna ungu.
    b. Pin TX dihubungkan ke pin D6 pada NodeMCU menggunakan kabel jumper warna biru.
    c. Pin VCC dihubungkan ke pin 3V pada NodeMCU menggunakan kabel jumper warna oren.
    d. Pin GND dihubungkan ke pin GND pada NodeMCU menggunakan kabel jumper warna kuning.
5. LED dihubungkan ke NodeMCU melalui setiap pin yang digunakan, yaitu:
    a. Kabel positif (+) dihubungkan ke pin D7 pada NodeMCU menggunakan kabel jumper warna putih.
    b. Kabel negatif (-) dihubungkan ke pin GND pada NodeMCU menggunakan kabel jumper warna hitam.

6. Buzzer dihubungkan ke NodeMCU melalui setiap pin yang digunakan, yaitu:
   a. Kabel positif (+) dihubungkan ke pin D8 pada NodeMCU menggunakan kabel jumper warna kuning.
   b. Kabel negatif (-) dihubungkan ke pin GND pada NodeMCU menggunakan kabel jumper warna coklat.

2. *Testing*

Setelah seluruh komponen berhasil diimplementasikan, langkah selanjutnya adalah mengintegrasikan setiap komponen tersebut ke dalam satu kesatuan sistem yang utuh. Sistem hasil integrasi kemudian menjalani tahap pengujian secara menyeluruh (*system testing*) guna memastikan bahwa seluruh fungsi dan alur kerja sistem berjalan dengan baik serta sesuai dengan spesifikasi kebutuhan yang telah ditetapkan pada tahap perancangan sebelumnya [18].

1) Uji *Software Smartphone* Android

Pengujian perangkat lunak pada aplikasi Android dilakukan untuk memastikan bahwa seluruh fungsi dan fitur yang telah ditetapkan dalam dokumen kebutuhan fungsional dapat berjalan dengan baik. Proses pengujian ini menggunakan metode *black-box testing*, yang berfokus pada pengujian keluaran berdasarkan masukan tanpa memperhatikan struktur internal kode program, guna mengevaluasi kesesuaian aplikasi dengan spesifikasi fungsional yang telah dirancang sebelumnya. Berikut adalah paparan hasil pengujian *software smartphone android* aplikasi pendeteksi dini kebakaran pada rumah menggunakan teknologi IoT yang dirincikan pada tabel 2.

Tabel 2. *Blackbox Testing Software Smartphone* Android

No	Kasus Pengujian	Hasil yang diharapkan	Hasil Aktual	Status
1	Monitoring setatus sensor api	Setatus sensor api ditampilkan	Setatus sensor api saat ini berhasil ditampilkan	Berhasil
2	Monitoring setatus sensor asap	Setatus sensor asap ditampilkan	Setatus sensor asap saat ini	Berhasil

No	Kasus Pengujian	Hasil yang diharapkan	Hasil Aktual	Status
			berhasil ditampilkan	
3	Sensor api kondisi aman	Setatus sensor api kondisi aman ditampilkan	Setatus sensor api kondisi aman berhasil ditampilkan	Berhasil
4	Sensor asap kondisi aman	Setatus sensor asap kondisi aman ditampilkan	Setatus sensor asap kondisi aman berhasil ditampilkan	Berhasil
5	Sensor api mendeteksi	Setatus sensor api terdeteksi bahaya ditampilkan	Setatus sensor api terdeteksi bahaya berhsasil ditampilkan	Berhasil
6	Sensor asap mendeteksi	Setatus sensor asap terdeteksi bahaya ditampilkan	Setatus sensor asap terdeteksi bahaya berhsasil ditampilkan	Berhasil
7	Mengirim lokasi GPS	Titik lokasi ditampilkan	Titik lokasi berhasil ditampilkan	Berhasil
8	Riwayat setatus sensor api/asap	Riwayat setatus sensor api/asap ditampilkan	Riwayat setatus sensor api/asap berhasil ditampilkan	Berhasil
9	Notifikasi kebakaran	Aplikasi mengirim notifikasi kebakaran	Notifikasi kebakaran berhasil dikirim	Berhasil

No	Kasus Pengujian	Hasil yang diharapkan	Hasil Aktual	Status
10	Integrasi *firebase realtime database*	Data setatus Sensor api/asap berubah ketika terdeteksi apa/asap	Data setatus Sensor api/asap berhasil berubah ketika terdeteksi api/asap	Berhasil

2) Uji Perangkat Keras

Pengujian perangkat keras dilakukan untuk memastikan bahwa seluruh fungsi dan fitur yang telah ditentukan dalam dokumen kebutuhan fungsional dapat berjalan sebagaimana mestinya. Proses pengujian ini menggunakan pendekatan *black-box testing*, yang difokuskan pada evaluasi keluaran perangkat berdasarkan masukan tertentu tanpa memperhatikan struktur internal dari perangkat keras tersebut. Pendekatan ini bertujuan untuk menilai kesesuaian kinerja perangkat keras terhadap spesifikasi fungsional yang telah dirancang sebelumnya. Berikut adalah paparan hasil pengujian perangkat keras aplikasi pendeteksi dini kebakaran pada rumah menggunakan teknologi IoT yang dirincikan pada tabel 3.

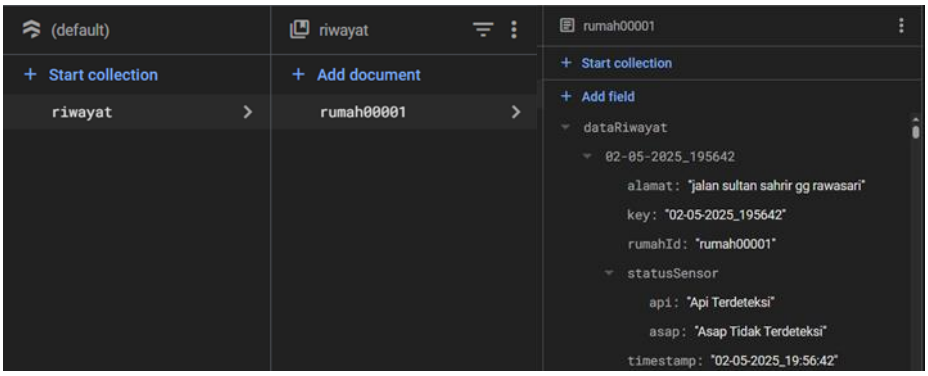
Tabel 3. Blackbox Testing Perangkat Keras

No	Kasus Pengujian	Input	Hasil yang diharapkan	Hasil Aktual	Setatus
1	Pembacaan setatus sensor api/asap	Setatus data sensor di kirim ke *firebase*	Setatus sensor api/asap berubah di *firebase*	Setatus sensor api/asap berhasil berubah	Berhasil
2	Pembacaan setatus GPS	Data lokasi di kirim ke *firebase*	Data GPS terbaca di *firebase*	Data GPS berhasil terbaca di *firebase*	Berhasil

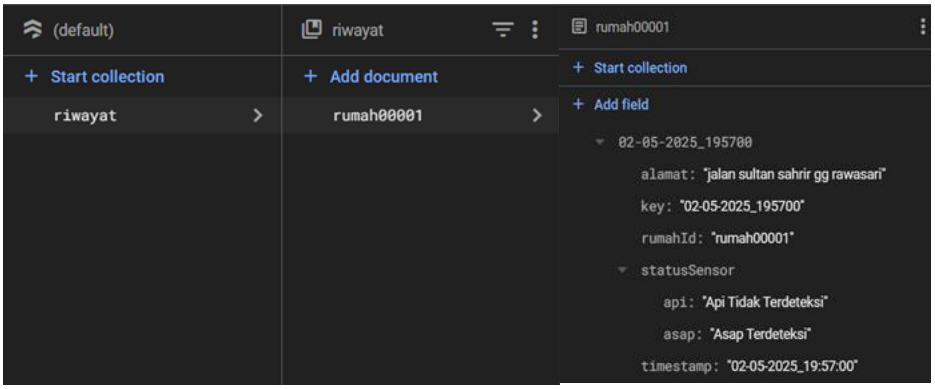
No	Kasus Pengujian	Input	Hasil yang diharapkan	Hasil Aktual	Setatus
3	Kontrol Led	-	Lampu menyala ketika NodeMCU terhubung ke wifi	Lampu berhasil menyala ketika NodeMCU terhubung ke wifi	Berhasil
4	Kontrol Buzzer	-	Buzzer aktif ketika Sensor api/asap mendeteksi	Buzzer berhasil aktif ketika Sensor api/asap mendeteksi	Berhasil
5	Koneksi *Wifi*	-	NodeMCU Esp8266 terhubung ke jaringan *wifi*	NodeMCU Esp8266 berhasil terhubung ke jaringan *wifi*	Berhasil
6	Mendapatkan data waktu	Sensor api/asap mendeteksi	NodeMCU Esp8266 mengirim data ke *firebase*	NodeMCU Esp8266 berhasil mengirim data ke *firebase*	Berhasil

3) Hasil Simulasi Kebakaran

Hasil simulasi kebakaran dari keseluruhan data sensor dilakukan untuk mengetahui apakah aplikasi dan sensor sudah berkerja sesuai dengan yang diinginkan. Proses pengambilan data ini dilakukan secara *realtime* dalam rentang waktu 28 April 2025- 2 Mei 2025. Data yang diperoleh berupa data tanggal, bulan, tahun, jam, menit, detik dan setatus sensor diambil dari *Firebase* yang hasil dari kiriman data riwayat oleh *NodeMCU*. Data tersebut disimpan di *Cloud Firestore* dalam bentuk file dengan format *json*. Berikut adalah gambarl hasil *output* dari data sensor pada tanggal 2 Mei 2025 pukul 19:56:42 WIB dalam bentuk format json di *firestore.*



Gambar 10. Data Simulasi Kebakaran Sensor Api Aktif



Gambar 11. Data Simulasi Kebakaran Sensor Asap Aktif

Pada gambar 10 dan 11 hasil yang diperoleh pada pembacaan sensor api dan asap yang dibuat dalam tabel 4 hasil simulasi kebakaran pada tanggal 2 mei 2025 pukul 19:56:42 WIB.

Tabel 4. Hasil Simulasi Kebakaran

Tanggal dan Waktu	Sensor Api	Sensor Asap	Setatus
2 Mei 2025 Jam 19:56:42	Api Tereteksi	Asap Tidak Terdeteksi	Kebakaran
2 Mei 2025 Jam 19:57:00	Api Tidak Terdeteksi	Asap Terdeteksi	Kebakaran

Bedasarkan hasil semulasi kebakaran pada tabel 4 setatus sensor api mendeteksi adanya keberadaan api dan setatus sensor asap tidak mendeteksi keberadaan asap pada pukul 19:56:42 WIB. Kemudian setatus sensor api tidak mendeteksi keberadaan api dan setatus sensor asap mendeteksi adanya keberadaan asap pada pukul 19:57:00 WIB. Dari pembacaan sensor pada tanggal 2 Mei 2025 menunjukan setatus rumah mengalami kebakaran

D. KESIMPULAN

Berdasarkan hasil implementasi dan pengujian, sistem pendeteksi dini kebakaran berbasis *Internet of Things* (IoT) yang dirancang dalam penelitian ini menunjukkan kinerja yang efektif dalam mendeteksi keberadaan api dan asap secara otomatis, serta mengirimkan notifikasi peringatan beserta lokasi kejadian secara *realtime* kepada pemilik rumah dan pihak Dinas Pemadam Kebakaran (DAMKAR) melalui aplikasi Android, dengan dukungan *Firebase Realtime Database* dan pendekatan pengembangan sistem menggunakan metode Waterfall. Sistem ini memiliki beberapa keunggulan, antara lain kemampuan mengirimkan peringatan dini secara *realtime*, fitur berbagi lokasi yang mempermudah pihak DAMKAR dalam merespons kejadian, antarmuka aplikasi yang sederhana dan mudah dipahami, serta sinkronisasi data antar perangkat yang efisien melalui *Firebase*. Selain itu, sistem tetap mengaktifkan alarm fisik (buzzer) meskipun tidak terhubung ke jaringan, dan suara alarm dalam aplikasi tidak dapat disenyapkan meskipun perangkat dalam mode senyap, sehingga meningkatkan keandalan sistem dalam memberikan peringatan. Namun demikian, sistem ini masih memiliki beberapa keterbatasan, seperti ketergantungan terhadap koneksi internet yang dapat mengganggu kinerja saat jaringan tidak stabil, cakupan deteksi yang terbatas hanya pada api dan asap, belum mendukung fitur *multi-akun* baik untuk satu rumah maupun satu pengguna

dengan banyak rumah, serta audio alarm yang dapat dihentikan setelah keluar dari akun. Selain itu, pengguna diwajibkan tetap masuk ke dalam aplikasi agar dapat menerima notifikasi kebakaran secara efektif.

DAFTAR PUSTAKA

CNNindonesia, "17.768 Kebakaran di 2021, 5.274 di Antaranya Akibat Korsleting," cnnindoensia. Accessed: Jul. 20, 2024. [Online]. Available: https://www.cnnindonesia.com/nasional/20220301134907-20-765357/17768-kebakaran-di-2021-5274-di-antaranya-akibat-korsleting

T. Laudia, "8 Penyebab Kebakaran di Rumah, Lengkap Penjelasannya," liputan6. Accessed: Jul. 20, 2024. [Online]. Available: https://www.liputan6.com/hot/read/5308206/8-penyebab-kebakaran-di-rumah-lengkap-penjelasannya?page=2

U. I. Abdullahi, W. Zhang, Y. Cao, and G. Irankunda, "Integrating IoT Technology for Fire Risk Monitoring and Assessment in Residential Building Design," *Buildings*, vol. 15, no. 8, 2025, doi: 10.3390/buildings15081346.

S. Suhartini, M. Peslinof, and M. F. Afrianto, "Rancang Bangun Sistem Deteksi Kebakaran pada Ruangan Berbasis Internet of Things (IoT)," *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 7, no. 3, p. 329, 2023, doi: 10.30998/string.v7i3.15493.

M. N. Adlini, A. H. Dinda, S. Yulinda, O. Chotimah, and S. J. Merliyana, "Metode Penelitian Kualitatif Studi Pustaka," *Edumaspul J. Pendidik.*, vol. 6, no. 1, pp. 974–980, 2022, doi: 10.33487/edumaspul.v6i1.3394.

Khairul Abidin, Fenty Kurnia Oktorina, and Slamet Triyanto, "Penerapan Metode SDLC Waterfal Dalam Pembuatan Sistem Pendataan Peternakan Kabupaten Kampar Berbasis Web Menggunakan Framework Codeigniter," *J. Elektron. dan Tek. Inform. Ter. JENTIK)*, vol. 1, no. 3, pp. 241–248, 2023, doi: 10.59061/jentik.v1i3.396.

A. A. Wahid, "'Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi,' ," *J. Ilmu-ilmu Inform. dan Manaj. STMIK*, vol. 1, no. November, 2020.

H. Pangestu, H. Alianto, and S. F. Wijaya, "Hasil Rancang Bangun Sistem Erp Dengan Sdlc Model Waterfall :," *J. Comtech*, vol. 3, no. 2, pp. 1036–1042, 2020.

R. A. Setyawan, "Penerapan Firebase Realtime Database Pada Aplikasi Catatan Harian Diabetes Melitus," *J. Inform. Komputer, Bisnis dan Manaj.*, vol. 22, no. 1, pp. 1–9, 2024, doi: 10.61805/fahma.v22i1.102.

H. J. Christanto and Y. A. Singgalen, "Analysis and Design of Student Guidance Information System through Software Development Life Cycle (SDLC) dan Waterfall Model," *J. Inf. Syst. Informatics*, vol. 5, no. 1, pp. 259–270, 2023, doi: 10.51519/journalisi.v5i1.443.

F. vazquez Penaloza and C. R. J. Gonzalez, "Towards a web application to create flowcharts for supporting the teaching-learning process of structured programming courses.," *Am. J. Educ. Res.*, vol. 7, no. 12, pp. 976–982, 2020, doi: 10.12691/education-7-12-12.

M. T. Chin, "V M o e la m t o i r le y bugejado System," vol. 2, 2022.

K. Nadiyah and G. S. Dewi, "Quality Control Analysis Using Flowchart, Check Sheet, P-Chart, Pareto Diagram and Fishbone Diagram," *Opsi*, vol. 15, no. 2, p. 183, 2022, doi: 10.31315/opsi.v15i2.7445.

V. Vranić, J. Lang, M. L. Nores, J. J. P. Arias, J. Solano, and G. Laseca, "Use case modeling in a research setting of developing an innovative pilgrimage support system," *Univers. Access Inf. Soc.*, vol. 23, no. 4, pp. 1543–1560, 2024, doi: 10.1007/s10209-023-01047-1.

M. M. I. Molla, J. Ahmad, and W. M. N. Wan Kadir, "A Comparison of Transforming the User Stories and Functional Requirements into UML Use Case Diagram," *Int. J. Innov. Comput.*, vol. 14, no. 1, pp. 29–36, 2024, doi: 10.11113/ijic.v14n1.463.

L. Lavazza, A. Locoro, and R. Meli, "Software Development and Maintenance Effort Estimation Using Function Points and Simpler Functional Measures," *Software*, vol. 3, no. 4, pp. 442–472, 2024, doi: 10.3390/software3040022.

M. Trinath Basu, R. Karthik, J. Mahitha, and V. Lokesh Reddy, "IoT based forest fire detection system," *Int. J. Eng. Technol.*, vol. 7, no. 2, pp. 124–126, 2018, doi: 10.14419/ijet.v7i2.7.10277.

T. S. Gbli, "Software Testing Techniques and Levels in Software Development," *Int. J. Adv. Comput. Technol.*, vol. 2, no. 1, pp. 10–19, 2024, doi: 10.56472/25838628/IJACT-V2I1P102.