

Perancangan Sistem Pengawas Pendeteksi Api Berbasis *Internet of Things*

1st Muhammad Annafi
Universitas Telkom

Fakultas Teknik Elektro
Bandung, Indonesia

aannafi@student.telkomuniversity.ac.id

2nd Ig. Prasetya Dwi Wibawa
Universitas Telkom

Fakultas Teknik Elektro
Bandung, Indonesia

Prasdwibawa@telkomuniversity.ac.id

3rd Achmad Rizal
Universitas Telkom

Fakultas Teknik Elektro
Bandung, Indonesia

achmadrizal@telkomuniversity.ac.id

Abstrak—Kamera *surveillance* pendeteksi api merupakan sistem keamanan atau pencegahan agar kebakaran tidak berdampak besar pada tempat tersebut. Api bisa muncul jika adanya pemicu dan dapat menyebar ke tempat yang mudah terbakar. Untuk mengurangi resiko kebakaran. Untuk mengurangi resiko kebakaran dapat dilakukan dengan memanfaatkan perkembangan teknologi saat ini, salah satunya memanfaatkan internet sebagai peringatan untuk keamanan suatu keadaan sekitar terutama tempat yang mudah terbakar.

Pada penelitian ini di buat sistem pendeteksi api menggunakan kamera sebagai pengawasnya. Kamera mendeteksi api yang kemudian data di proses *mikrokontroler*, selanjutnya data dari *mikrokontroler* di kirim ke *smartphone* dengan memanfaatkan aplikasi MIT APP *Inventor* berbasis *Internet of Things* dan mengirim data berupa notifikasi suhu ruangan dan video *live stream* yang dapat di pantau pengguna dari jarak jauh. Diharapkan sistem ini dapat membantu manusia untuk mengawasi tempat yang rawan terjadi kebakaran, agar dapat menjadi pencegahan dini bila terjadi kebakaran.

Tujuan dari pembuatan alat ini adalah agar dapat mencegah terjadi nya kebaran dan mengurangi resiko api menyebar ke tempat di sekitarnya. Menggunakan firebase data real-time, sensor DHT 22 didapatkan dari program Arduino dan deteksi objek menggunakan kamera dengan python yang kemudian data tersebut masuk ke firebase data real-time dan langsung terkirim ke Aplikasi dengan tingkat keberhasilan mengirim notifikasi 100%.

Kata Kunci: *Api, Kamera, Internet of Things, internet cloud*

I. PENDAHULUAN

Kebakaran merupakan musibah yang terjadi karena faktor kelalaian manusia atau faktor alam. Kasus kebakaran dapat terjadi di mana saja baik pada hutan, gedung, pusat perbelanjaan maupun stasiun pengisian bahan bakar umum. Api yang muncul biasanya dikarenakan faktor suhu, benda yang mudah terbakar, dan oksigen. Banyak kebakaran disebabkan oleh kelalaian manusia contohnya seperti membuang sesuatu yang mudah terbakar atau membakar sesuatu sembarangan. Kebakaran disebabkan karena ada nya pemicu api dan udara, besar kecilnya kadar oksigen mempengaruhi juga besar atau kecilnya api yang muncul. Musibah kebakaran ini dapat memicu kerugian yang cukup besar dari kerugian material, terhentinya kegiatan usaha, dan dapat menimbulkan ancaman terhadap keselamatan jiwa

manusia. Maka harus dilakukan pencegahan dan keamanan agar api cepat untuk dipadamkan sebelum menyebar ke benda yang mudah terbakar.

Pada instruksi menteri tenaga kerja No. : INS.11/M/BW/1997 yang berisi tentang mencegah kebakaran, mengurangi faktor yang menyebabkan kebakaran, dan memadamkan kebakaran. Pengurus atau pengusaha bertanggung jawab mencegah, mengurangi dan, memadamkan kebakaran, memberi latihan penanggulangan kebakaran pada tempat kerja sesuai dengan peraturan yang sudah ditentukan, agar kebakaran tidak terjadi atau tidak meluas ke daerah di sekitar tempat kebakaran awal muncul[1]. Kamera penjejak bisa menjadi pilihan untuk memonitoring tempat yang sudah dipasang kamera penjejak, Bertujuan untuk menjaga keamanan pada tempat tersebut. Kamera penjejak pendeteksi api mencegah jika terjadi kebakaran dan memberi peringatan untuk melakukan tindakan agar mencegah penyebaran api meluas dan menghindari bencana yang menyebabkan kerugian besar dan kehilangan nyawa manusia.

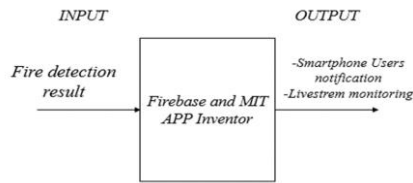
Pada penelitian sebelumnya ditulis oleh Fathoni, Eko Prayetno, dan Tonny Suhendra (2018) yang berjudul Sistem Pemantau Realtime Pada Ruang Server Kantor Badan Pengelola Pajak Dan Retribusi Daerah Kabupaten Bintan. Penelitian ini menggunakan Raspberry Pi dengan kamera webcam yang digunakan untuk melakukan monitoring keadaan sekitar [2]. Dalam pengembangan yang dilakukan penulis yaitu merancang kamera *surveillance* yang berfokus pada mendeteksi keberadaan api. Ketika kamera menemukan percikan api maka kamera bergerak menuju objek api tersebut, kemudian hasil deteksi dari kamera akan di olah menggunakan *fuzzy logic* untuk mengirimkan pemberitahuan berupa notifikasi suhu ruangan dan video ke *smartphone*, agar pemilik tempat tersebut dapat segera memberi tindakan agar tidak menyebar ke tempat disekitarnya. Pemilik atau pengguna dapat memantau dari jarak jauh dan melihat posisi api yang terdeteksi oleh kamera.

II. KAJIAN TEORI

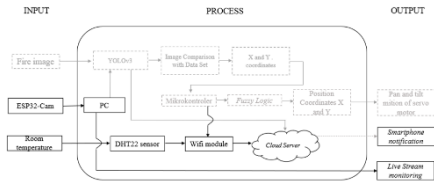
A. Prinsip Kerja Solusi

Pada perancangan kamera pendeteksi api dengan menggunakan metode pengolahan citra dan notifikasi berbasis *Internet of Things* (IoT). Sistem ini menggunakan Laptop, kamera, dan motor servo. Kamera penjejak ini bertujuan ketika alat tersebut mendeteksi api maka kamera penjejak akan mengirimkan sebuah notifikasi berupa suhu ruangan dan video *real time* ke *smartphone* pengguna jika

adanya api yang terdeteksi oleh kamera, berikut diagram fungsi dari alat tersebut.



Gambar II-1 Diagram Konsep Sistem



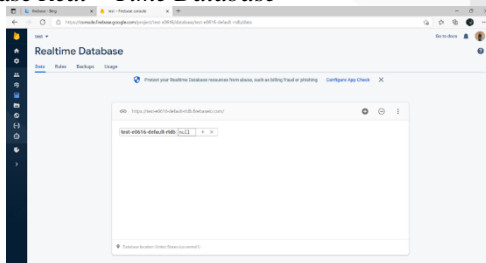
Gambar II-1 Diagram Konsep Sistem

Dari proses sebelum nya, ketika *image processing* menangkap objek api dan sensor suhu mendeteksi suhu pada ruangan, selanjutnya akan dikirim data tersebut ke *firebase* data *real time*, setelah diolah *firebase* data selanjut nya akan dikirim kan ke aplikasi yang sudah dibuat melalui MIT APP Inventor. Jika api terdeteksi oleh kamera aplikasi MIT APP Inventor akan memanggil sebuah notifikasi ke *smartphone*, agar api segera di padamkan.

B. Internet of Things (IoT)

Internet of Things atau dikenal juga dengan singkatan IoT, merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terusmenerus yang memungkinkan pengguna 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. *Internet of Things* atau sering disebut IoT adalah sebuah gagasan dimana semua benda di dunia nyata dapat berkomunikasi satu dengan yang lain sebagai bagian dari satu kesatuan sistem terpadu menggunakan jaringan internet sebagai penghubung. [7]

C. Firebase Real – Time Database



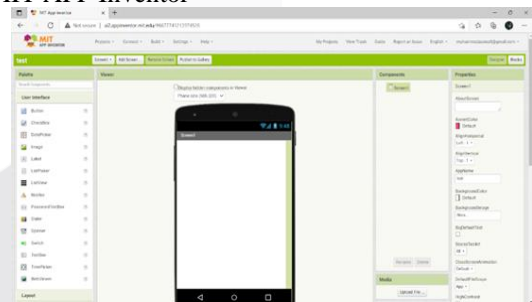
Gambar II-3 Firebase Realtime Database

Firestore Realtime Database adalah database yang ada pada cloud. Data disimpan sebagai JSON (*JavaScript Object Notation*) dan disinkronkan secara real-time dengan setiap klien atau data yang alat yang terhubung. Saat membuat aplikasi lintas-platform menggunakan android semua klien

akan berbagi contoh database real-time dan secara otomatis akan menerima pembaruan dari data terbaru [8]. Firebase Realtime Database memungkinkan Anda untuk membuat aplikasi kolaboratif dan kaya fitur dengan menyediakan akses yang aman ke database, langsung dari kode sisi klien. Data disimpan di *drive* lokal. Bahkan saat offline sekalipun, peristiwa realtime terus berlangsung, sehingga pengguna akhir akan merasakan pengalaman yang responsif. Ketika koneksi perangkat pulih kembali, Realtime Database akan menyinkronkan perubahan data lokal dengan update jarak jauh yang terjadi selama klien offline, sehingga setiap perbedaan akan otomatis digabungkan. API *Real-time* Database dirancang agar hanya mengizinkan operasi yang dapat dijalankan dengan cepat. [9]

Firestore memiliki produk utama, yaitu menyediakan database realtime dan backend sebagai layanan (Backend as a Service). Layanan ini menyediakan pengembang aplikasi API yang memungkinkan aplikasi data yang akan disinkronisasi di klien dan disimpan di *cloud* Firestore. Firestore menyediakan library untuk berbagai clientplatform yang memungkinkan integrasi dengan Android, iOS, JavaScript, Java, Objective-C dan Node aplikasi Js dan dapat juga disebut sebagai layanan DbaaS (Database as a Service) dengan konsep realtime. Firestore digunakan untuk mempermudah dalam penambahan fitur-fitur yang akan dibangun oleh developer. DatabaseRealtime merupakan basis data data dalam firestore yang berbasis cloud dan tidak memerlukan query berbasis SQL untuk menyimpan dan mengambil data. Basis data ini terkenal sangat handal dan supercepat dalam proses update data dan sinkronisasi sehingga data tetap dipertahankan bahkan ketika user tidak terhubung dengan internet sekalipun data tetap dipertahankan [10].

D. MIT APP Inventor



Gambar II-4 Tampilan MIT App Inventor

MIT App Inventor 2 adalah Integrated Development Environment (IDE) yang didapat bagi semua kalangan untuk mengembangkan aplikasi Android tanpa harus berpengalaman dalam dunia pemrograman. Pada App Inventor pengguna tidak harus bersentuhan dengan dunia pemrograman komputer yang kompleks. Pengguna tidak perlu menuliskan kode-kode pemrograman untuk membuat aplikasi. Yang uniknya dari App Inventor 2 adalah App Inventor 2 berbasis Visual Block Programming yang mampu mentransformasikan pengkodean bahasa pemrograman kedalam bentuk visual dalam bentuk kode-kode program. Pengkodean program dilakukan dengan cara Drag – Drop blok-blok kode program kedalam Block Editor yang kemudian menyusun block-block tersebut seperti

halnya pengguna menyusun puzzle sesuai dengan alur program yang pengguna kehendaki [11].

Untuk tampilan MIT App Inventor itu sendiri terdapat 2 halaman utama, yaitu halaman designer dan halaman blocks. Untuk Halaman Designer digunakan untuk mendesain tampilan aplikasi dengan berbagai komponen serta layout yang sudah disediakan sesuai dengan keinginan pengguna, sedangkan untuk halaman block berfungsi untuk memprogram jalannya aplikasi android sesuai dengan tujuan yang diinginkan [12].

MIT App Inventor dapat di hubungkan dengan firebase dengan menyalin firebase URL dan Firebase Token yang sudah di dapatkan pada console firebase. Data yang dikirim bisa secara real-time masuk aplikasi MIT App Inventor. Kemudian data yang sudah masuk ke firebase dapat diterima oleh aplikasi yang sudah di buat dari MIT App Inventor [12].

E. Quality of Service (QoS)

Quality of Service (QoS) merupakan mekanisme pada jaringan yang menentukan bahwa aplikasi-aplikasi atau layanan yang dapat beroperasi sesuai dengan standart kualitas layanan yang telah diterapkan. Ada Standart Quality of Services (QoS) salah satunya adalah THIPON (Telecommunications and Internet Protocol Harmonization Over Network) [13]. Adapun parameter QoS yang akan dihitung yaitu:

1. Throughput

Throughput merupakan kecepatan transfer data. Throughput adalah jumlah total kedatangan paket yang sukses diamati pada tujuan selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut.

$$\text{Throughput} = \frac{\text{Paket data yang diterima}}{\text{waktu pengamatan}} \quad (2.1)$$

2. Delay

Delay adalah waktu yang dibutuhkan sebuah data untuk menempuh jarak dari asal ke tujuan. Delay dapat dipengaruhi oleh jarak media fisik dan waktu proses yang lama.

$$\text{Rata - rata delay} = \text{Time2} - \text{Time 1} \quad (2.2)$$

3. Jitter

Jitter adalah variasi delay yang disebabkan oleh variasi-variasi panjang antrian dalam waktu mengolah data. Delay antrian pada router dan switch dapat menyebabkan jitter.

$$\text{Jitter} = \frac{\text{Total Variasi Delay}}{\text{Total paket data yang diterima} - 1} \quad (2.3)$$

4. Packet Loss

Packet loss adalah banyaknya paket yang gagal mencapai tempat tujuan paket tersebut dikirim.

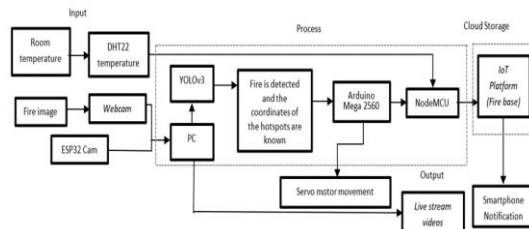
$$\text{Packet loss} = \frac{(\text{Paket data dikirim} - \text{paket data di terima})}{\text{paket datang yang dikirim}} \times 100\% \quad (2.4)$$

III. METODE

A. DESAIN ALAT

Dalam mengimplementasikan sistem ini maka akan dirancang perangkat keras dari sistem tersebut. Sistem ini bekerja pada kamera yang sudah dirancang dengan image processing dan menggunakan sensor DHT22 untuk mendeteksi api pada keadaan sekitar. Sistem ini menggunakan laptop sebagai controller, webcam untuk menampilkan keadaan sekitar tempat, motor servo sebagai aktuator untuk menggerakkan kamera, ESP8266 untuk mengirim hasil sensor DHT22 ke firebase, dan Arduino Mega 2560 sebagai microcontroller untuk mengendalikan motor servo. Alat ini bekerja ketika image processing sudah menemukan api yang terdeteksi oleh kamera dan sensor DHT22 mendapatkan suhu, kemudian akan dikirim ke firebase dan data nya akan diolah agar langsung dikirim berupa notifikasi ke aplikasi yang sudah dipasang pada smartphone pengguna. Tujuan tugas akhir ini agar dapat mencegah terjadinya kebakaran yang dapat menyebar luas ke area sekitar.

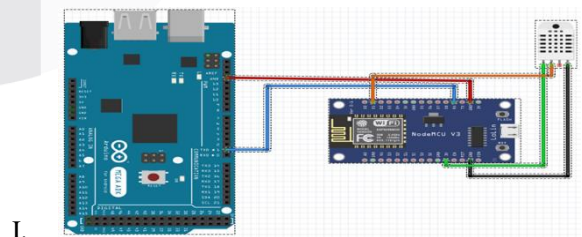
B. PERANGKAT KERAS



Gambar III-1 Diagram Blok Sistem

Pada Gambar III-1 telah dibuat diagram blok sistem kamera pendeteksi api dengan inputan sensor suhu dan penangkapan berupa cita dari kamera webcam saat memonitoring keadaan disekitar kemudian data tersebut akan masuk ke firebase agar dapat dikirim ke smartphone pengguna berupa notifikasi dan dapat live stream dari smartphone.

C. RANCANGAN PERANGKAT KERAS

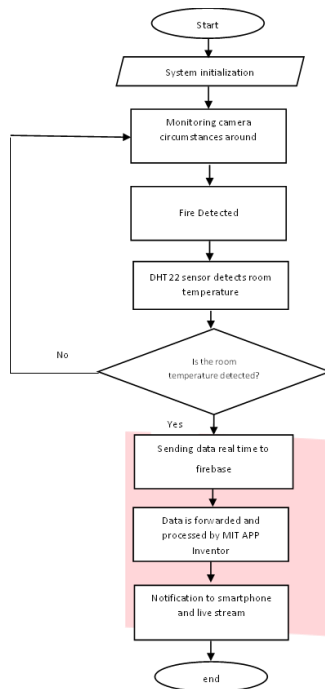


I.

Gambar III-3 Serial Arduino – NodeMcu

Pada Gambar III-3 dilakukan serial pada Arduino Mega dengan NodeMCU dan sensor DHT 22 disambungkan ke NodeMCU. Pada serial ini pin Tx pada arduino disambungkan ke pin Rx NodeMCU yang bertujuan untuk transfer data dari Arduino Mega yang kemudian diterima oleh NodeMCU.

D. Digram Aliran Sistem



Gambar III-12 Flowchar Individu

Pada Gambar III-12 adalah alur sistem yang dibuat, yang dimulai dari:

- a. Sistem akan melakukan inisialisasi.
- b. Kamera akan *memonitoring* keadaan pada sekitar yang sudah dipasang kamera
- c. Kamera akan menangkap objek, objek yang ditangkap kamera akan diolah dengan *image processing*.
- d. Sensor DHT22 akan mendeteksi suhu untuk mengetahui berapa suhu pada ruangan tersebut
- e. Ketika api terdeteksi oleh kamera dan DHT22 mendapatkan suhu ruangan maka data tersebut akan dikirim ke *firebase*
- f. Data yang sudah masuk pada *firebase* data *real-time* akan langsung dikirim ke aplikasi yang sudah dibuat dengan MIT App Inventor
- g. Pada *smartphone* pengguna akan muncul notifikasi dari aplikasi tersebut dan jika aplikasi dibuka akan muncul suhu yang terdeteksi dan dapat melihat video secara *realtime* untuk mengetahui titik api yang terdeteksi.

IV. HASIL PENGUJIAN DAN ANALISIS

A. Pengujian Alat

Agar mendapatkan nilai akurasi dengan nilai error dari setiap sensor yang digunakan penulis untuk tugas akhir, maka penulis menggunakan persamaan sebagai berikut:

$$Akurasi = 100\% - \%error \quad (4.1)$$

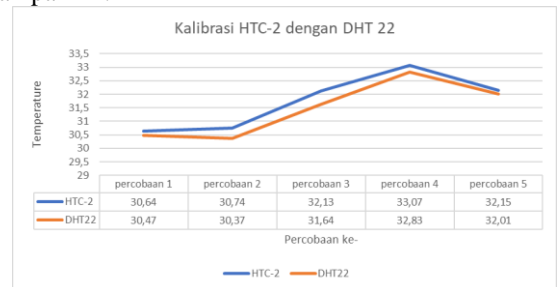
Dengan

$$\%error = \left| \frac{\text{measurement value} - \text{actual value}}{\text{actual value}} \right| \times 100$$

(4.2)

4.1.1. Kalibrasi Sensor DHT22

Kalibrasi sensor DHT22 dilakukan dengan cara mengambil data suhu sebanyak 5 kali percobaan, masing masing percobaan dilakukan 10 kali dengan jarak awal 10cm hingga 100cm dan menghidupkan 5 buah lilin menjadi 1, pada rentang waktu 3 menit secara berkala dengan jarak 10cm sampai 1m.

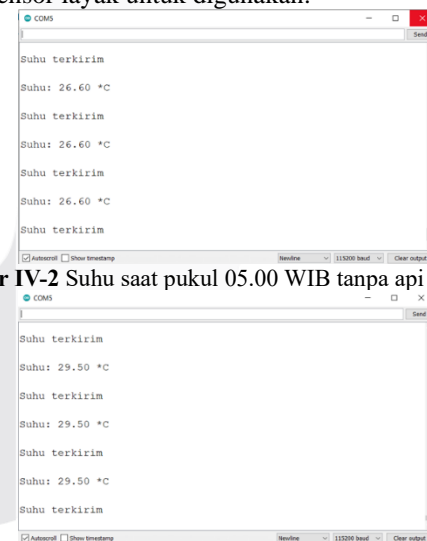


Gambar IV-11 Kalibrasi DHT22 dengan Sensor HTC-2

$$S = \sqrt{\frac{\sum(X - \bar{X})^2}{n - 1}} \quad (4,3)$$

$$S = \sqrt{\frac{4,38}{4}} = 1,04^{\circ}\text{C}$$

Kalibrasi ini dilakukan di kamar kos penulis. Dengan rata – rata didapatkan sensor DHT 22 dari lima percobaan yaitu 31,46°C dengan standar deviasi 1,04°C. Hasil pengujian tersebut didapatkan nilai rata – rata akurasi dari sensor suhu DHT22 yang telah dihitung dengan lima kali percobaan menggunakan persamaan (4.1), didapatkan nilai error rata – rata 0,88% dengan tingkat akurasi rata – rata 99,12%, karena akurasi yang cukup tinggi saat pengujian maka sensor layak untuk digunakan.



Gambar IV-2 Suhu saat pukul 05.00 WIB tanpa api

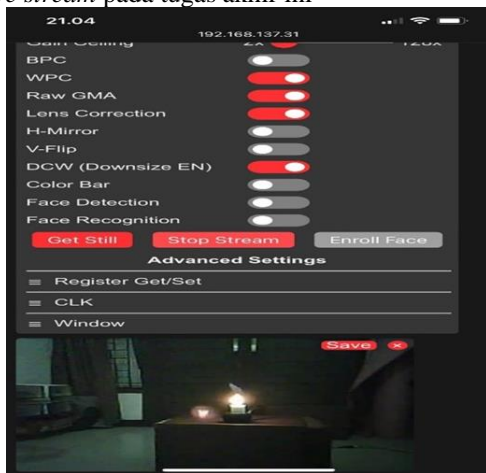
Gambar IV-3 Suhu saat pukul 05.00 WIB ada api

Pada Gambar IV-2 pengujian DHT22 pada jam 05.00 WIB sampai 05.30 WIB didapatkan suhu seperti pada gambar. Ketika ada api suhu berubah menjadi seperti pada gambar Gambar IV-3 naik tiga derajat celcius dapat dilihat perbedaan suhu ruangan ketika api tidak ada dan api ada dengan sensor DHT22.

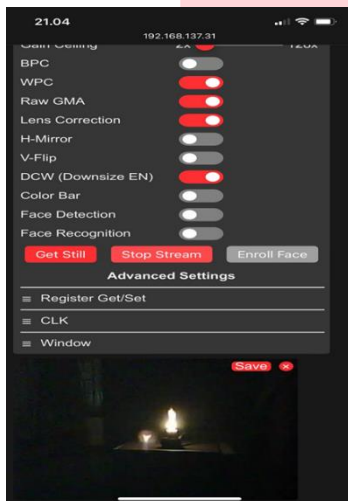
B. Pengujian Kamera

Pada gambar Gambar IV-2 dan Gambar IV-3 dilakukan pengujian kamera menggunakan ESP32-Cam yang bertujuan untuk *live stream* ketika api terdeteksi. Pengujian

ini dilakukan dengan menghidupkan cahaya lampu pada ruangan dan mematikan lampu pada ruangan. Terlihat api pada kamera cukup jelas dengan menggunakan ESP32-Cam untuk *live stream* pada tugas akhir ini



Gambar IV-5 Pengujian ESP32-Cam tanpa cahaya lampu



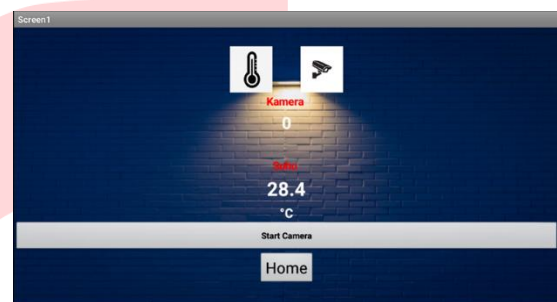
Gambar IV-4 Pengujian ESP32-Cam dengan cahaya lampu

yang terdapat pada *Firebase* dihubungkan dengan program di *Arduino* dan *Python* agar *Firebase* dapat menerima data yang dikirimkan dari *NodeMCU*. Pada gambar terdapat hasil pembacaan dari kamera yang diambil dari *image processing*, jika *image processing* pada kamera mendeteksi ada nya api maka nilai yang akan dikirim ke *Firebase* adalah 1, ketika kamera tidak mendeteksi api maka nilai nya akan 0.

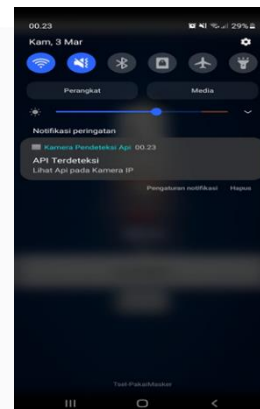
4.3.2. Tampilan Aplikasi MIT APP Inventor Pada Android



Gambar IV-7 Icon Aplikasi



Gambar IV-8 Tampilan Aplikasi



Gambar IV-9 Notifikasi Smart Phone

Pada gambar di atas memperlihatkan tampilan awal dari *interface* aplikasi untuk *Android* yang sudah didesain dan dirancang menggunakan website *MIT APP Inventor*. Ketika aplikasi sudah didesain dan dirancang oleh penulis, selanjutnya aplikasi dibuat dan didownload, setelah aplikasi didownload maka aplikasi sudah bisa di buka melalui *smartphone*, jika kamera mendeteksi nilai satu maka notifikasi akan muncul pada *smartphone*, sebaliknya ketika nilai pada kamera nol maka notifikasi tidak muncul. Suhu pada aplikasi akan update dari data yang dikirim oleh *firebase* jika *Arduino* masih terhubung pada port *USB*.

C. Pengujian Aplikasi

Pengujian sistem *monitoring* berbasis *internet of things (IoT)* bertujuan untuk membantu pengguna dalam menampilkan suhu pada ruangan dan video secara *real-time* yang kemudian ditampilkan pada *mobile* dari *MIT App Inventor*.

4.3.1. Tampilan Data Terkirim Pada *Firebase* Data *Real-time*



Gambar IV-6 Tampilan *firebase* data *real-time*

Pada gambar di atas dapat dilihat tampilan *Firebase* sebagai *Internet of Things Platform* yang digunakan penulis. Setelah membuka website dan membuka data *real-time* URL

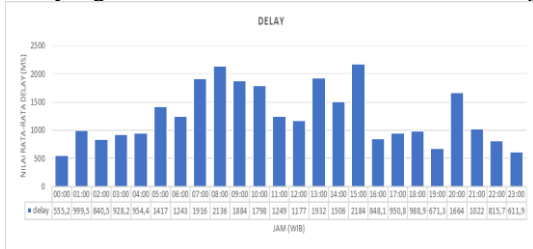
D. Pengujian Sistem Komunikasi

Pengujian sistem komunikasi atau *Quality of Service (QoS)* dilakukan menggunakan *software Wireshark* untuk mendapatkan parameter nilai *delay time*, *jitter*, *throughput*, dan *packet loss* untuk mengetahui kualitas pada *Internet of Things (IoT)*. Penulis menggunakan standar dari *THIPON* untuk menentukan kualitas dari setiap parameter tersebut.

Pengujian dilakukan pada daerah di tempatkannya alat *monitoring* yaitu di kosan Permata Buah Batu No. 40 blok G, Lekong, kecamatan Bojongsong, kabupaten Bandung, Jawa Barat. Berikut adalah hasil dari *Quality of Service* (QoS).

E. Pengujian Delay Time Terhadap Pengiriman Data dari NodeMCU ke Firebase

Pada pengujian ini dilakukan pengujian selama 24 jam dikarenakan alat yang penulis rancang aktif selama 1 hari penuh. Pengujian menghasilkan nilai rata – rata delay dari 20 paket data yang dikirim dalam interval waktu selama 1 jam.

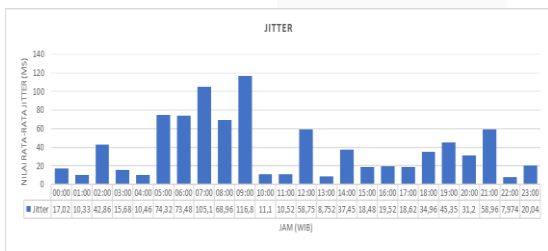


Gambar IV-10 Hasil Pengujian delay

Hasil dari pengujian *delay* dengan mengirim 20 paket data yang dikirim dalam interval waktu setiap 1 jam, didapatkan pengukuran rata – rata *delay* sebesar 1262 ms dengan standar deviasi 499,62 ms. Pengujian *delay* ini termasuk dalam kategori jelek **Table II-4**, karena faktor padatnya penghuni kos dan untuk kamar sendiri jaringan sedikit terhambat.

F. Pengujian Jitter Terhadap Pengiriman Data dari NodeMCU ke Firebase

Pada pengujian ini dilakukan pengujian selama 24 jam dikarenakan alat yang penulis rancang aktif selama 1 hari penuh. Pengujian menghasilkan nilai rata – rata delay dari 20 paket data yang dikirim dalam interval waktu selama 1 jam.

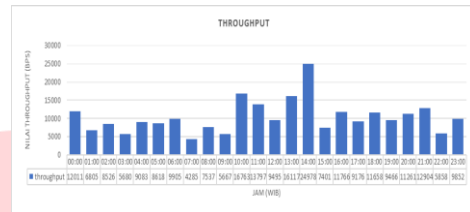


Gambar IV-11 Hasil Pengujian Jitter

Hasil dari pengujian jitter sebanyak 20 paket data yang dikirim dalam Interval waktu setiap 1 jam, didapatkan pengukuran rata-rata variasi *jitter* sebesar 38,19ms dengan standar deviasi 31,19ms. Pengujian jitter ini tergolong dalam kategori baik dan tergolong pada indeks 3. Dikarenakan kecilnya angka variasi delay antar data, hal ini dapat diabaikan dan sekaligus menandakan bahwa data yang dikirim tepat sasaran dan kapasitas jaringan yang ada sudah cukup sehingga kemungkinan adanya kerusakan informasi dalam pengiriman paket data dapat dihindari.

G. Pengujian nilai Throughput terhadap pengiriman data dari NodeMCU ke firebase

Pada pengujian kali ini dilakukan pengujian selama waktu 1 hari atau 24 jam. Pengujian menghasilkan nilai throughput dari 20 paket data dalam interval waktu setiap 1 jam untuk melihat proses komunikasi dan pengiriman data dari NodeMCU menuju Firebase data *real – time*. Nilai throughput dapat dihitung dengan persamaan (2.1). Berikut hasil pengujian *throughput* dalam pengiriman data ke Firebase.

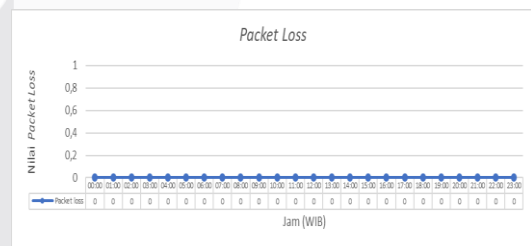


Gambar IV-12 Hasil Pengujian Throughput

Hasil dari pengujian *throughput* pada pengiriman data dari ESP32 ke Firebase dengan pengukuran rata – rata nilai kecepatan transfer sebesar 10358 bps dengan standar deviasi 4448,87 bps. Pengujian *throughput* ini termasuk dalam kategori sangat bagus dan tergolong pada indeks ke 4 seperti pada tabel (II-3). Hal ini menandakan bahwa data yang dikirim dalam selang waktu tertentu berhasil dengan kecepatan transfer yang efektif.

H. Pengujian nilai Packet loss terhadap pengiriman data dari NodeMCU ke firebase

Pada pengujian ini dilakukan pengujian selama 1 hari atau 24 jam. Pengujian mendapatkan hasil *packet loss* dari 20 paket data dalam interval 1 jam untuk melihat proses komunikasi dan pengiriman data dari NodeMCU menuju *firebase* data *real – time*. Berikut hasil pengujian *packet loss* dalam pengiriman data ke Firebase.



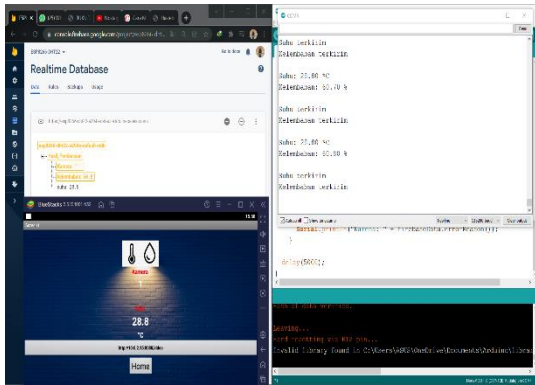
Gambar IV-13 Hasil Pengujian Packet Loss

Hasil dari pengujian packet loss pada pengiriman data dari NodeMCU ke Firebase didapatkan tidak adanya data paket yang tidak terkirim karna semua data berhasil diterima oleh Firebase sehingga penggunaan modul NodeMCU dalam komunikasi sudah berjalan dengan baik. Pengujian packet loss ini termasuk dalam kategori sangat bagus dan termasuk indeks ke 4 seperti pada Table II-6.

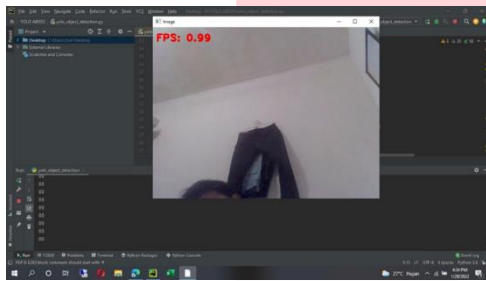
I. Pengujian Sistem Keseluruhan

Pada pengujian sistem secara keseluruhan, sistem diuji mulai dari kamera mendeteksi api dengan menggunakan

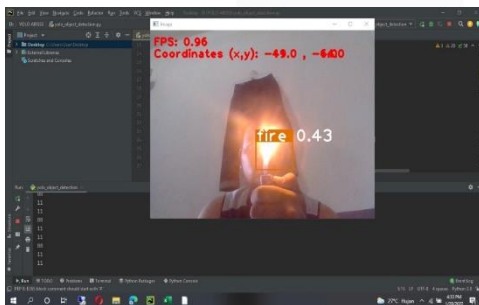
Python dan sensor DHT 22 menggunakan Arduino IDE lalu ke dua data tersebut dikirimkan ke Firebase data real-time. Kemudian Firebase data real-time mengirim data ke aplikasi yang sudah dibuat.



Gambar IV-14 Percobaan Sistem Keseluruhan 1



Gambar IV-15 Percobaan Sistem Keseluruhan 2



Gambar IV-16 Percobaan Sistem Keseluruhan 3

Pada Gambar IV-14 merupakan percobaan dilakukan pada Arduino IDE, Firebase, dan Aplikasi. Pada gambar tersebut diketahui bahwa Arduino mengirim data yang didapat pada sensor DHT 22 lalu dikirimkan pada Firebase data real time, lalu pada Firebase data teruskan ke Aplikasi. Dapat diketahui juga data suhu yang terdeteksi dapat dilihat pada serial monitor, data real time, dan aplikasi yaitu sama sebear 28,8. Pada Gambar IV-15 dapat dilihat bahwa jika tidak ada api maka python bernilai 0 (tidak terdeteksi). Pada Gambar IV-16 dapat dilihat bahwa jika ada api maka python akan mendeteksi api dan mengeluarkan nilai 1 (Terdeteksi).



V. KESIMPULAN

Berdasarkan pengerjaan dan pengujian yang dilakukan oleh penulis pada Tugas Akhir ini, maka dapat di simpulkan:

1. Sistem dapat *monitoring* suhu sudah dibuat menggunakan sensor DHT 22 dengan akurasi 99,12% dan kamera mampu mengirim video *live stream* ke *smartphone* android menggunakan aplikasi IoT.
2. Dalam aplikasi terdapat suhu ruangan dan ESP32-Cam tujuannya agar dapat melihat api pada ruangan yang sudah dipasang kamera dari *smartphone*, dan dapat mencegah api menyebar lebih luas lagi.
3. Sistem berbasis *Internet of Things* menggunakan firebase data *real-time*, sensor DHT 22 didapatkan dari program Arduino dan deteksi objek menggunakan kamera dengan python yang kemudian data tersebut masuk ke firebase data *real-time* dan langsung terkirim ke Aplikasi dengan tingkat keberhasilan mengirim notifikasi di atas 100%.

A. SARAN

Berdasarkan pengerjaan dan pengujian yang dilakukan pada tugas akhir ini, penulis menyarankan beberapa hal yang tujuannya untuk melakukan pengembangan selanjutnya pada tugas akhir ini. Berikut beberapa saran dari penulis:

1. Disarankan untuk menggunakan sensor yang lebih akurat agar sensor dapat dipasang lebih jauh dan mendapatkan hasil yang lebih baik lagi.
2. Menggunakan kamera dengan resolusi yang lebih tinggi agar api yang terdeteksi lebih jelas dan api dapat dilihat posisi nya.
3. Aplikasi yang digunakan hanya dapat diakses oleh android tidak dapat diakses pada *IoS*, disarankan untuk menggunakan aplikasi yang dapat diakses oleh *IoS*

REFERENSI

- [1] Instruksi Menteri Tenaga Kerja no. : ins.11/m/bw/1997 *Tentang Pengawasan Khusus K3 Penanggulangan Kebakaran*
- [2] Fathoni, Prayeno. Eko, dan Suhendra. Tonny, "SISTEM PEMANTAUREALTIME PADA RUANG SERVER KANTOR BADANPENGELOLA PAJAK DAN RETRIBUSI DAERAH KABUPATEN BINTAN", Teknik Elektro Universitas Maritim Raja Ali Haji, 2018.

- [3] Rifandi. Rifki, Sutarti, dan Anharudin, "RANCANG BANGUN KAMERA PENGAWAS MENGGUNAKAN RASPBERRY DENGAN APLIKASI TELEGRAM BERBASIS *INTERNET OF THINGS*". Jurnal PROSISKO. Vol.8 No.1. Rekayasa Sistem Komputer Universitas Serang Raya. 2021.
- [4] H. Kristomson, H. Subrata. Rosalia, Gozali. Ferrianto "Sistem Keamanan Ruang Berbasis *Internet Of Things* Dengan Menggunakan Aplikasi Android" jurnal Teknik Elektro Vol.20 No.2, 2018.
- [5] R. Terok. Filisia, F. Sangkop. Ivan, Santa. Kristofel "Sistem Pendeteksi Gerakan Berbasis *Internet Of Things*" Artikel, Vol.01, No. 1, 2018
- [6] Efendi Yoyon "INTERNET OF THINGS (IOT) SISTEM PENGENDALIAN LAMPU MENGGUNAKAN RASPBERRY PI BERBASIS MOBILE". Jurnal Ilmiah Ilmu Komputer. Vol.4, No. 1, Teknik Informatika STMIK Amik Riau. 2018.
- [7] Rafi'. Muhammad. Ramadhani, Tri. Fetty. Anggraeny, Prakarsa Eka Mandyartha "RANCANG BANGUN SISTEM KAMERA PENDETEKSI API SEDERHANA MENGGUNAKAN RASPBERRY PI" Jurnal Informatika dan Sistem Informasi. Vol.2, No. 2, 2 Juli 2021, Ilmu Komputer, UPN "Veteran" Jawa Timmur
- [8] Asfihan Akbar, "Fire Base Adalah : Cara Kerja, Sejarah, Manfaat, Kelebihan, dan Kekurangan" artikel, <https://adalah.co.id/>, 1 Januari 2022.
- [9] Ahadiansyah Dyan "Fitur Luar Biasa Dari Google Firebase : Firebase Realtime Database" artikel, www.fit.labs.telkomuniversity.ac.id/, 1 Agustus 2019
- [10] Prasetyo. Anjani "Membuat Aplikasi Sederhana Berbasis Android Menggunakan MIT App Inventor 2" artikel, www.predatech.org, 15 Mei 2020
- [11] Richard George Payara, Tanone Radius "Penerapan *Firestore Realtime Database* Pada *Prototype* Aplikasi Pemesanan Makanan Berbasis Android" Jurnal Teknik Informatika dan Sistem Informasi Vol. 4. No. 3, Desember 2018
- [12] Arya. Wahyu pambudi "CRUD Firebase App Inventor Kodular" artikel, www.kubahas.com, 18 Mei 2020
- [13] Wisnu Sumogo Pamungkas, Kusri, Pramono Eka "Analisis *Quality of Service (QoS)* Pada Jaringan Hotspot SMA Negri XYZ" Jurnal Sistem Informasi dan Teknologi Informasi Vol. 7, No. 2, Oktober 2018

