

PENGEMBANGAN APLIKASI ANDROID PADA SISTEM PENDETEKSI DINI KEBAKARAN LAHAN

Nada Araminta¹

Program Studi Broadband Multimedia, Jurusan Teknik Elektro, Politeknik Negeri Jakarta
Email: nadaaraminta03@gmail.com

Agus Wagyana¹

Program Studi Broadband Multimedia, Jurusan Teknik Elektro, Politeknik Negeri Jakarta
Email: agus.wagyana@elektro.pnj.ac.id

ABSTRACT: Land fires are a tragedy that is difficult to control if the fire is large and can result in considerable losses. To be able to reduce the impact of losses due to land fires, an early warning is needed, therefore an application system is made that can detect land fires based on fire, smoke and temperature parameters. The methods used in the process of making this application are application system planning, application feature planning, layout design and java programming for each feature using Android Studio, configuration and programming on firebase, and finally the application can be released. This application uses firebase as a database platform that will store sensor data, user data, and cloud functions. The way this application works is that the application will display data on land conditions in real time, and when there is a change in status, the firebase will execute a command to provide a fire status warning via push notification in the form of a "safe", "alert", "fire level 1" status, or "fire level 2" to the application so that landowners can act on them immediately. The quality of this application is assessed through testing based on the ISO/EIC 25010 standard, namely testing database accuracy, compatibility, usability, and performance efficiency aspects. With the test results of 100% database accuracy, 100% compatibility, and 86% usability, the application is categorized as very feasible and with 3,213% CPU usage efficiency and 159,213 KB memory that meets little eye standards, it shows that the "Siaga Api" application has very good quality.

Keyword: Land Fire, Android Application, ISO/EIC 25010, Realtime Database, Monitoring Application.

ABSTRAK: Kebakaran lahan merupakan tragedi yang sulit untuk dikendalikan jika api sudah besar dan dapat mengakibatkan kerugian yang cukup besar. Untuk dapat mengurangi dampak kerugian akibat kebakaran lahan, diperlukan adanya sebuah peringatan dini, maka dari itu dibuatlah sebuah sistem aplikasi yang dapat mendeteksi kebakaran lahan berdasarkan parameter api, asap dan suhu. Metode yang digunakan pada proses pembuatan aplikasi ini adalah perencanaan sistem aplikasi, perencanaan fitur aplikasi, perancangan layout serta pemrograman java untuk setiap fitur menggunakan Android Studio, konfigurasi dan pemrograman pada firebase, dan akhirnya aplikasi dapat di release. Aplikasi ini menggunakan firebase sebagai platform database yang akan menyimpan data sensor, data pengguna, serta cloud function. Cara kerja dari aplikasi ini adalah aplikasi akan menampilkan data kondisi lahan secara realtime, dan ketika terjadi perubahan status maka firebase akan mengeksekusi perintah untuk memberikan peringatan status kebakaran melalui push notification berupa status "aman", "waspada", "kebakaran level 1", atau "kebakaran level 2" ke aplikasi sehingga pemilik lahan dapat langsung bertindak menanganinya. Kualitas aplikasi ini dinilai melalui pengujian yang berdasarkan standar ISO/EIC 25010 yaitu pengujian akurasi database, aspek compatibility, usability, dan performance efficiency. Dengan hasil uji akurasi database 100%, compatibility 100%, dan usability 86% aplikasi dikategorikan sangat layak serta dengan performance efficiency CPU usage 3,213% dan memory 159,213 KB yang memenuhi standar little eye menunjukkan bahwa aplikasi "Siaga Api" memiliki kualitas yang sangat baik.

Kata Kunci: Kebakaran Lahan, Aplikasi Android, ISO/EIC 25010, Realtime Database, Aplikasi Pemantauan.

PENDAHULUAN

Sekitar 98% kebakaran hutan dan lahan gambut di Indonesia terjadi akibat ulah manusia, baik karena kesengajaan maupun kelalaian. Pengelolaan lahan gambut yang tidak tepat, seperti praktik pembukaan lahan untuk pertanian dan perkebunan, adalah beberapa contohnya. Pembukaan lahan ini biasanya diikuti pengeringan serta pembersihan lahan dengan api. Padahal, tanah gambut yang kering akan mudah sekali untuk terbakar jika terkena api.

Pada tahun 2019 lalu, kebakaran hutan dan lahan di Indonesia setidaknya menghancurkan 1,6 juta hektar lahan [1], dan sekitar 500 ribu hektarnya terjadi di lahan gambut. Sepanjang Januari hingga Desember 2019, terdapat 55.006 peringatan kebakaran di tujuh provinsi prioritas restorasi gambut, yaitu Jambi, Kalimantan Barat, Kalimantan Selatan, Kalimantan Tengah, Papua, Riau, dan Sumatera Selatan. Puncak peringatan kebakaran terjadi di bulan September [2].

Kebakaran lahan merupakan salah satu tragedi yang datangnya tidak dapat diprediksi, disamping tidak diinginkan oleh masyarakat juga sering tidak terkendalikan apabila api sudah besar. Kejadian kebakaran sangat membahayakan dan mengganggu kehidupan dan penghidupan masyarakat. Kebakaran dikategorikan sebagai salah satu bentuk bencana. Bencana adalah peristiwa atau rangkaian

¹ Program Studi Broadband Multimedia, Jurusan Teknik Elektro, Politeknik Negeri Jakarta

peristiwa yang mengancam dan mengganggu kehidupan dan penghidupan masyarakat yang disebabkan oleh faktor alam dan/atau faktor non alam maupun faktor manusia sehingga mengakibatkan timbulnya korban jiwa, kerusakan lingkungan, kerugian harta benda dan dampak psikologis [3].

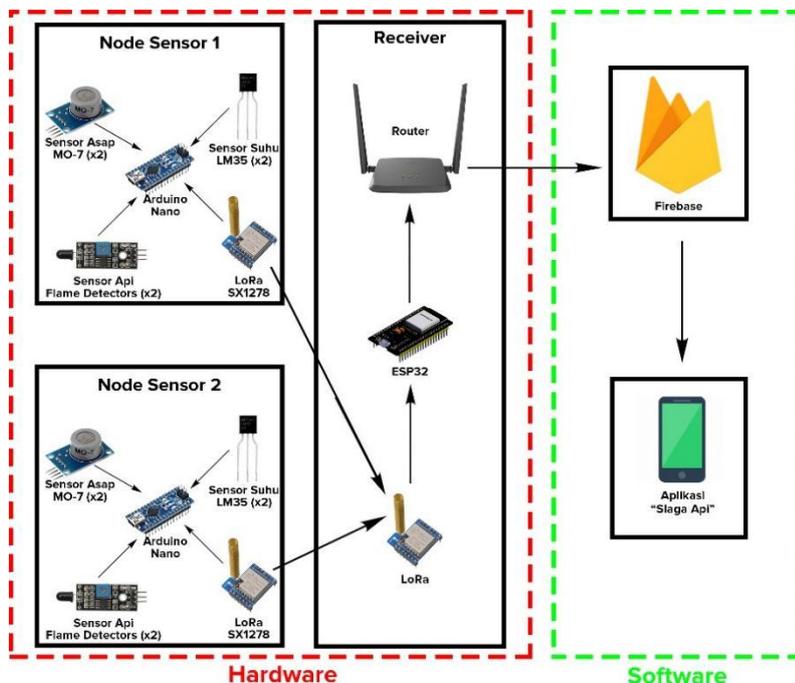
Dalam penanggulangan masalah kebakaran, banyak sekali ditemukan kesulitan-kesulitan, seperti sukarnya ditemukan sumber api yang menyala, sehingga api akan terus menjalar ke tempat lain dan kerugian pun akan menjadi semakin besar. Untuk itu perlu adanya sebuah sistem yang memudahkan untuk memperoleh peringatan informasi potensi kebakaran di suatu lahan. Informasi potensi kebakaran tersebut dapat diketahui salah satunya dari asap seperti dalam penelitian Dani Sasmoko dan Arie Mahendra (2017) yang membuat sistem pendeteksi kebakaran berbasis IoT dan SMS gateway menggunakan arduino dengan tujuan untuk mengurangi resiko dan tingkat penyakit yang disebabkan oleh asap kebakaran hutan [4].

Informasi potensi kebakaran tersebut harus dapat dikirimkan secara *realtime* kepada pemilik lahan. Siswanto *et al* (2020) dalam penelitiannya membuat sebuah *prototype Wireless Sensor Network* (WSN) sistem pendeteksi dini kebakaran yang dapat mengirimkan informasi dari sensor ke penjaga hutan melalui internet [5]. Joniwarta *et al* (2019) dalam penelitiannya mengenai sistem *monitoring* kebakaran berbasis SMS gateway, menggunakan SMS dan alarm untuk dapat memberikan peringatan secara otomatis ke *user* ketika terjadi kebakaran [6].

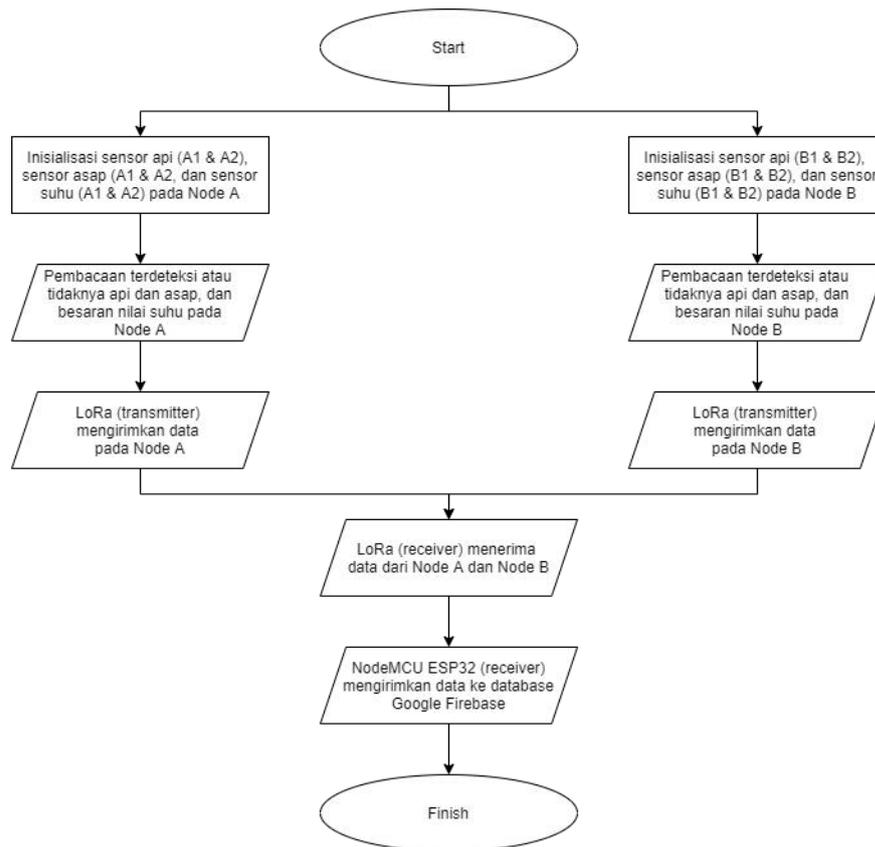
Penelitian ini mengembangkan dari penelitian-penelitian sebelumnya yang telah disebutkan diatas, yakni sistem peringatan dini kebakaran yang akan menggunakan sensor api *flame detectors*, sensor asap MQ-7, dan sensor suhu LM35 untuk dapat memprediksi potensi kebakaran menggunakan Arduino Nano yang nantinya data akan diterima oleh NodeMCU ESP32 melalui LoRa RA-02 SX1278. Informasi yang diperoleh dari sensor tersebut dikirimkan secara *realtime* ke *cloud* dan akan memberikan peringatan langsung secara personal kepada pemilik lahan melalui *push notification* sebuah aplikasi yang didesain khusus.

METODOLOGI

Pengembangan aplikasi android pada sistem pendeteksi dini kebakaran lahan bekerja berdasarkan *hardware* dari sistem pendeteksi dini kebakaran lahan yang diinstalasi pada sebuah maket lahan. Diagram blok dari sistem pendeteksi dini kebakaran lahan yang terintegrasi dengan aplikasi android ditunjukkan pada Gambar 1.



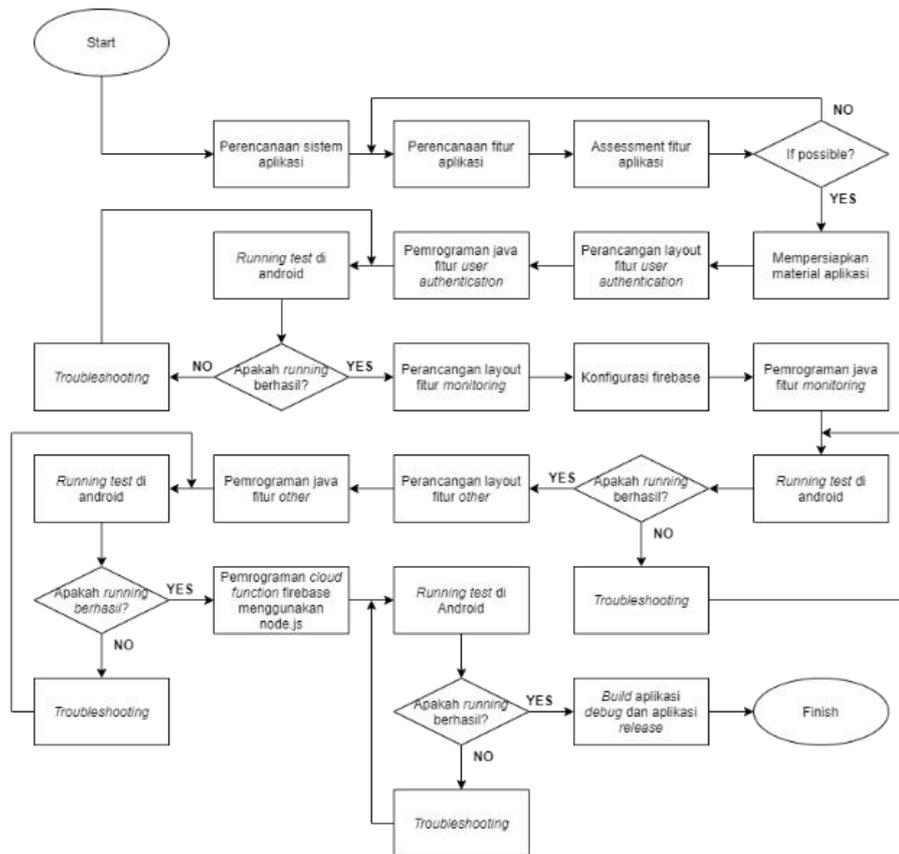
■ Gambar 1. Diagram Blok Sistem Pendeteksi Dini Kebakaran Lahan yang Terintegrasi dengan Aplikasi Android



■ **Gambar 4.** *Flowchart Cara Kerja Hardware*

Berdasarkan *flowchart* pada Gambar 4, dapat diketahui proses pembuatan aplikasi dari sistem sebagai berikut :

1. Pada sistem pendeteksi dini kebakaran lahan, terbagi menjadi Node A dan Node B yang masing masing dari node tersebut memiliki komponen 1 buah Arduino Nano, 1 buah LoRa RA-02 SX1278, 2 buah sensor api *flame detectors*, 2 buah sensor asap MQ-7, dan sensor suhu LM35. Tahap awal dari alur kerja sistem ini adalah proses inisialisasi pin dari setiap komponen yang digunakan pada Arduino Nano. Sensor api *flame detectors* memerlukan 2 buah pin digital, sensor asap MQ-7 memerlukan 2 buah pin analog, sensor suhu LM35 memerlukan 2 buah pin analog, dan LoRa RA-02 SX1278 memerlukan 6 pin digital. Sehingga terdapat total 8 pin digital dan 4 pin analog. Terdapat juga pin 3.3V sebanyak 1 buah, pin VCC sebanyak 6 buah, dan pin GND sebanyak 7 buah.
2. Setelah tiap pin yang digunakan di inisialisasi oleh Arduino Nano, selanjutnya sensor api *flame detectors*, sensor asap MQ-7, dan sensor suhu LM35 akan bekerja. Sensor api *flame detectors* akan mendeteksi ada atau tidak adanya api, sensor asap MQ-7 akan mendeteksi besaran nilai asap pada lahan, dan sensor suhu LM35 akan mendeteksi besaran nilai suhu pada lahan.
3. Setelah data dari sensor didapatkan, LoRa RA-02 SX1278 pada masing-masing node akan mengirimkan data kepada LoRa RA-02 SX1278 yang terdapat pada sisi receiver.
4. Selanjutnya NodeMCU ESP32 akan mengirimkan data ke Google Firebase untuk dikirimkan langsung ke aplikasi yang digunakan sebagai monitoring lahan.



■ Gambar 5. Flowchart Pembuatan Aplikasi

Berdasarkan *flowchart* pada Gambar 5, dapat diketahui proses pembuatan aplikasi dari sistem sebagai berikut :

1. Langkah pertama yang harus dilakukan dalam proses pembuatan aplikasi ada merencanakan sistem aplikasi seperti apa yang ingin dibuat. Pada aplikasi ini sistem aplikasi yang akan dibuat aplikasi ini dapat digunakan pada sistem operasi Android 5.0 (Lollipop) sampai dengan Android 11.0 (R) dan dapat digunakan pada API level 21 sampai dengan 30. Sistem aplikasi ini juga bekerja dengan cara menampilkan hasil *monitoring* dari data yang didapatkan dari sensor dan memberikan notifikasi peringatan juga ada perubahan status.
2. Jika sudah menentukan sistem aplikasi, maka langkah selanjutnya adalah perencanaan fitur aplikasi. Pada aplikasi ini, fitur yang digunakan dibagi menjadi 3, yaitu fitur *user authentication*, fitur *monitoring*, dan fitur *other*. Lalu dilakukan *assessment* fitur untuk mengetahui apakah fitur- fitur yang ada bisa dibuat di Android Studio atau tidak? Jika tidak, maka fitur yang ada harus disesuaikan kembali. Jika iya, maka bisa dilanjutkan dengan menentukan material yang akan digunakan untuk proses pembuatan aplikasi. Pada aplikasi ini, fitur-fitur yang direncanakan bisa dibuat menggunakan Android Studio, maka dapat berlanjut ke proses mempersiapkan material aplikasi.
3. Karena fitur-fitur yang direncanakan bisa dibuat menggunakan Android Studio, maka langkah selanjutnya adalah mempersiapkan segala material yang dibutuhkan. Adapun material yang dibutuhkan, yaitu logo aplikasi, *icon/symbol* yang digunakan untuk aplikasi, kode warna yang digunakan untuk aplikasi, dan lain-lain.
4. Fitur pertama yang dibuat adalah fitur *user authentication*. Didalam fitur *user authentication* terdapat beberapa tampilan halaman aplikasi dan pemrograman javanya, yaitu halaman *login*, halaman registrasi, halaman lupa *password*, dan halaman verifikasi. Tahap awal yang harus dilakukan adalah merancang layout untuk masing-masing halaman, selanjutnya melakukan pemrograman java

5. untuk masing-masing halaman, lalu lakukan *running test* di *handphone* Android. Jika gagal, lakukan *troubleshooting* agar aplikasi bisa di *run* kembali. Jika berhasil, maka lanjut mengerjakan fitur berikutnya.
6. Fitur kedua yang dibuat adalah fitur *monitoring*. Didalam fitur *monitoring* terdapat beberapa tampilan halaman aplikasi dan pemrograman javanya, yaitu halaman “status”, halaman “api”, halaman “asap”, dan halaman “suhu”. Tahap awal yang harus dilakukan adalah merancang layout untuk masing-masing halaman, selanjutnya melakukan pemrograman java untuk masing- masing halaman, lalu lakukan *running test* di *handphone* Android. Jika gagal, lakukan *troubleshooting* agar aplikasi bisa di *run* kembali. Jika berhasil, maka lanjut mengerjakan fitur berikutnya.
7. Fitur ketiga yang dibuat adalah fitur *other*. Didalam fitur *other* terdapat beberapa tampilan halaman aplikasi dan pemrograman javanya, yaitu halaman “tentang”, halaman “tentang aplikasi”, dan halaman “tentang kami”, dan halaman verifikasi. Tahap awal yang harus dilakukan adalah merancang layout untuk masing-masing halaman, selanjutnya melakukan pemrograman java untuk masing-masing halaman, lalu lakukan *running test* di *handphone* Android. Jika gagal, lakukan *troubleshooting* agar aplikasi bisa di *run* kembali. Jika berhasil, maka lanjut mengerjakan langkah berikutnya.
8. Langkah selanjutnya adalah melakukan pemrograman *cloud function* firebase menggunakan node.js. Pemrograman ini berfungsi untuk mentrigger *push notification* jika ada perubahan status pada *database* firebase. Apabila sudah dilakukan pemrograman, selanjutnya adalah *running test* di *handphone* Android. Jika gagal, lakukan *troubleshooting* agar aplikasi bisa di *run* kembali. Jika berhasil, maka lanjut mengerjakan langkah berikutnya.
9. Jika *running test* dari pemrograman *cloud function* firebase sudah berhasil dan tidak ada kendala lagi, maka aplikasi dapat di *release* dan digunakan oleh *user*.



Gambar 6. Flowchart Cara Kerja Aplikasi

Berdasarkan *flowchart* pada Gambar 6, dapat diketahui bahwa aplikasi sistem pendeteksi dini kebakaran lahan berbasis Android memiliki alur kerja sebagai berikut :

1. Ketika memasuki aplikasi, halaman awal yang akan muncul adalah halaman *loading screen*. Halaman *loading screen* merupakan layar pembuka setiap kali pengguna menjalankan aplikasi.
2. Untuk dapat mengakses aplikasi, pengguna harus melakukan *login* akun terlebih dahulu, jika pengguna belum memiliki akun maka pengguna harus mendaftar atau melakukan registrasi terlebih dahulu. Pembuatan akun dilakukan dengan mendaftarkan nama pengguna, alamat email, *password*, dan konfirmasi *password*.
3. Setelah pengguna memiliki akun, pengguna harus melakukan verifikasi akun melalui *link* yang dikirim oleh Google Firebase melalui email yang didaftarkan. Jika pengguna sudah melakukan verifikasi akun maka pengguna bisa melakukan aktivitas *login*. Namun jika pengguna belum melakukan verifikasi akun maka pengguna tidak bisa melakukan aktivitas *login*.
4. Jika akun sudah terverifikasi, maka pengguna dapat melakukan aktivitas *login* dan pengguna dapat mengakses halaman utama dari aplikasi. Pada halaman utama ini pengguna dapat me- *monitoring* terdeteksi atau tidaknya api, besaran nilai asap, besaran nilai suhu, dan status pada maket.
5. Ketika ada perubahan data pada *database* status, pengguna akan menerima *push notification* meskipun aplikasi sedang tidak dijalankan atau dalam posisi *background activity*.

Spesifikasi aplikasi yang dirancang untuk sistem pendeteksi dini kebakaran lahan ditunjukkan pada Tabel 1 berikut ini.

Tabel 1. Spesifikasi *Mobile Application* Pada Sistem Pendeteksi Dini Kebakaran Lahan Berbasis Android

Nama	Keterangan
Sistem Operasi	Android 5.0 (Lollipop) – Android 11.0 (R)
API Level	21 – 30

Aplikasi “Siaga Api” ini juga memiliki 4 perubahan status, yaitu :

1. Status “AMAN” yang ditandai dengan tidak ada api, tidak ada asap, dan suhu $<35^{\circ}\text{C}$ pada masing-masing node dan masing-masing titik.
2. Status “WASPADA” yang ditandai dengan tidak ada api, ada asap, dan suhu $\geq 35^{\circ}\text{C}$ pada salah satu titik yang sama di salah satu node, atau bahkan pada kedua titik di salah satu node.
3. Status “KEBAKARAN LEVEL 1” yang ditandai dengan ada api pada salah satu titik di salah satu node, ada asap, dan suhu $\geq 35^{\circ}\text{C}$.
4. Status “KEBAKARAN LEVEL 2” yang ditandai dengan ada api pada kedua titik di salah satu node, ada asap, dan suhu $\geq 35^{\circ}\text{C}$.

Realisasi *mobile application* pada sistem pendeteksi dini kebakaran lahan berbasis Android dapat dilihat pada Gambar 7 berikut ini.



Gambar 7. Realisasi *Mobile Application* Pada Sistem Pendeteksi Dini Kebakaran Lahan Berbasis Android

(a) Tampilan Halaman “Status”, (b) Tampilan Halaman “Api”, (c) Tampilan Halaman “Asap”, (d) Tampilan Halaman “Suhu”, (e) Tampilan Halaman “Tentang”

Langkah-langkah pengujian yang dilakukan adalah sebagai berikut :

1. Pengujian akurasi *database* bertujuan untuk menguji ketepatan *reading realtime database* Firebase.
2. Pengujian kualitas aplikasi menggunakan standar ISO/EIC 25010 pada karakteristik aspek *compatibility*, aspek *usability*, dan aspek *performance efficiency* yang dijelaskan sebagai berikut:
 - a. Karakteristik aspek *compatibility* bertujuan untuk mengetahui kemampuan aplikasi untuk berjalan pada *environment* yang berbeda. Setelah dilakukan pengujian, selanjutnya dilakukan perhitungan persentase keberhasilan

- b. menggunakan rumus pada Persamaan (1) sebagai berikut :

$$\text{Persentase keberhasilan (\%)} = \frac{\text{Skor yang didapat}}{\dots} \times 100\%$$

..... (1)

Setelah dilakukan perhitungan, nilai persentase yang didapatkan akan diklasifikasikan berdasarkan Tabel 2.

■ **Tabel 2.** Standar Kelayakan Aplikasi

Persentase Skor	Interpretasi
81 % - 100%	Sangat layak
61 % - 80 %	Layak
41 % - 60 %	Cukup layak
21 % - 40 %	Tidak layak
0 % - 20 %	Sangat tidak layak

- c. Karakteristik aspek *usability* bertujuan untuk mengetahui sistem aplikasi dapat bekerja dengan baik apabila dipergunakan oleh pengguna, sehingga semua kemampuan sistem dapat dapat bermanfaat secara maksimal. Pengujian ini dilakukan dengan menyebarkan kuesioner yang berisikan pertanyaan mengenai aplikasi berdasarkan ada metode *System Usability Scale* (SUS). Pada metode SUS terdapat nilai skala aspek *usability*. Nilai skala aspek *usability* ditunjukkan pada Tabel 3.

■ **Tabel 3.** Nilai Skala Aspek *Usability*

PK	ST	TS	R	S	SS
Nilai	1	2	3	4	5

Keterangan :

- *PK: Pertanyaan Kuesioner*
- *ST: Sangat Tidak Setuju*
- *TS: Tidak Setuju*
- *R: Ragu-Ragu*
- *S: Setuju*
- *SS: Sangat Setuju*

- d. Karakteristik aspek *performance efficiency* bertujuan untuk mengetahui dan menguji kinerja aplikasi untuk sub-karakteristik *resource utilization* agar dapat memenuhi persyaratan ketika menjalankannya. Aspek *performance efficiency* dilakukan untuk mengetahui dan menguji kinerja aplikasi yaitu jumlah dan jenis sumber daya yang digunakan oleh sistem dengan menggunakan *tools* pengujian Firebase Test Lab dengan cara uji coba *RoboTest*. Pengujian yang dilakukan adalah melihat performansi pada CPU *usage* dan *memory*.
3. Pengujian sistem notifikasi peringatan bertujuan untuk mengetahui apakah sistem pemberian *warning* melalui *push notification* ke aplikasi dapat berjalan dengan baik, dan bagaimana performansinya. *Response time* pada *database* merupakan waktu yang dibutuhkan oleh *client* untuk mengirimkan *request* atau permintaan ke server *database* dan menerima respon dari server *database*. *Request* dapat berupa *writing, reading, connecting, disconnecting*, maupun *broadcast*. *Writing* merupakan sebuah *request* untuk melakukan penulisan data pada *database*. *Reading* merupakan *request* untuk membaca data yang ada pada *database*. *Connect-disconnect* merupakan *request* untuk berkomunikasi dengan data, dan *broadcast* merupakan *request* untuk mengirimkan informasi pembaruan data pada *database* ke semua *client*. *Response time database* dapat dikategorikan berdasarkan standar *Query Response Time index* (QRTi) MySQL Enterprise ditunjukkan pada Tabel 4.

Tabel 4. Kategori *Response Time* Berdasarkan Standar QRTi MySQL Enterprise
(Sumber: QRTi MySQL Enterprise)

Kategori <i>Response Time</i>	<i>Response Time</i> (ms)
<i>Optimum</i>	100
<i>Acceptable</i>	100 – 400
<i>Unacceptable</i>	>400

PENGUJIAN PENELITIAN

Terdapat 5 pengujian yang dilakukan, yaitu pengujian akurasi *database*, pengujian *compatibility* aplikasi, pengujian *usability* aplikasi, pengujian *performance efficiency* aplikasi, dan pengujian sistem notifikasi peringatan.

a) Pengujian Akurasi Database

Pengujian akurasi *database* bertujuan untuk mengetahui ketepatan antara *realtime database* pada Firebase dengan aplikasi “Siaga Api” agar mengetahui apakah sistem aplikasi dapat bekerja dengan baik dan dapat memenuhi tujuan penelitian. Tingkat akurasi ini ditinjau dari masing-masing data sensor yang digunakan, yaitu data ada atau tidak adanya api, ada atau tidak adanya asap, dan besaran nilai suhu. Pengujian ini akan dilakukan pada *realtime database* Firebase dan tampilan pada aplikasi “Siaga Api”.

b) Pengujian Compatibility Aplikasi

Pengujian *compatibility* aplikasi bertujuan untuk mengetahui kemampuan aplikasi untuk berjalan pada *environment* yang berbeda. Pengujian ini dilakukan dengan melakukan instalasi dan *menjalankan* aplikasi pada 5 perangkat *handphone* dengan spesifikasi yang berbeda. Pengujian *compatibility* aplikasi akan membuktikan apakah sistem aplikasi sudah mampu untuk diinstal dan dijalankan pada *environment* yang berbeda. Perangkat yang digunakan dalam proses pengujian *compatibility* aplikasi ini terdapat pada Tabel 5.

Tabel 5. Perangkat *Handphone* yang Digunakan dalam Proses Pengujian *Compatibility* Aplikasi

No.	Perangkat	Spesifikasi
1	Samsung Galaxy A32	Versi Android : 11 API Level : 33
2	Oppo A57	Versi Android : 6 API Level : 23
3	Xiaomi Redmi Note 10	Versi Android : 11 API Level : 30
4	Samsung Galaxy A20	Versi Android : 10 API Level : 29
5	Asus Zenfone 5	Versi Android : 9 API Level : 28

c) Pengujian Usability Aplikasi

Pengujian *usability* aplikasi bertujuan untuk mengetahui sistem aplikasi dapat bekerja dengan baik apabila dipergunakan oleh pengguna, sehingga semua kemampuan sistem dapat bermanfaat secara maksimal. Pengujian ini dilakukan dengan menggunakan metode *System Usability Scale* (SUS). Pengujian ini akan dilakukan dengan cara menyebarkan kuesioner yang berisi pertanyaan mengenai aplikasi berdasarkan pada

metode SUS. *Platform* yang digunakan untuk melakukan pengujian ini adalah *google form* dan menggunakan skala *likert* sebagai parameternya untuk mengukur penilaian dari reponden. Skala yang digunakan terdiri dari 1 sampai dengan 5. Pertanyaan yang digunakan pada kuesioner ini berdasarkan pada metode SUS sebagai berikut yang tertera pada Tabel 6.

■ **Tabel 6.** Pertanyaan Metode SUS

Kode	Pertanyaan
Q1	Saya akan menggunakan aplikasi ini lagi.
Q2	Saya merasa aplikasi ini rumit untuk digunakan.
Q3	Saya merasa aplikasi ini mudah digunakan.
Q4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan aplikasi ini.
Q5	Saya merasa fitur-fitur aplikasi ini berjalan dengan semestinya.
Q6	Saya merasa ada banyak hal yang tidak konsisten pada aplikasi ini.
Q7	Saya merasa orang lain akan memahami cara menggunakan aplikasi ini dengan cepat.
Q8	Saya merasa aplikasi ini membingungkan.
Q9	Saya merasa tidak ada hambatan dalam menggunakan aplikasi ini.
Q10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan aplikasi ini.

d) Pengujian *Performance Efficiency* Aplikasi

Pengujian *performance efficiency* aplikasi bertujuan untuk mengetahui dan menguji kinerja aplikasi untuk sub-karakteristik *resource utilization* agar dapat memenuhi persyaratan ketika menjalankannya. *Resource utilization* adalah jumlah dan jenis sumber daya yang digunakan oleh sistem atau produk agar dapat memenuhi persyaratan ketika menjalankannya. Pengujian ini dilakukan dengan menggunakan fitur Firebase Test Lab, dengan uji coba *RoboTest*, untuk mengetahui penggunaan CPU *usage*, dan *memory* pada perangkat saat menjalankan aplikasi “Siaga Api” sehingga dapat diketahui performansinya. Pada Firebase Test Lab, *simulation device* yang digunakan adalah Pixel 3, API Level 28.

e) Pengujian Sistem Notifikasi Peringatan

Pengujian sistem notifikasi peringatan bertujuan untuk mengetahui apakah sistem pemberian *warning* melalui *push notification* ke aplikasi dapat berjalan dengan baik, dan bagaimana performansinya. Pengujian performansi dilakukan dengan Firebase *function log* melalui console Firebase. Pengujian ini juga bertujuan untuk mengukur waktu yang dibutuhkan untuk melakukan proses pengiriman *warning* kepada pengguna.

HASIL PENELITIAN

Dari 5 pengujian penelitian yang telah dijelaskan diatas, maka berikut ini merupakan hasil yang didapatkan dari pengujian penelitian tersebut.

a. Hasil Pengujian Akurasi Database

Pada hasil pengujian akurasi sensor akan dilihat bagaimana keakuratan dari masing-masing data pada database *Firebase* dengan data yang ditampilkan pada aplikasi “Siaga Api”. Pengujian dilakukan pada sensor api *flame detectors*, sensor asap MQ-7, dan sensor suhu LM35. Hasil pengujian ini ditunjukkan pada Tabel 7, Tabel 8, dan Tabel 9.

■ **Tabel 7.** Hasil Pengujian Database Sensor Api *Flame Detectors*

No.	Data	Data Realtime Database	Data Aplikasi	Sesuai	Tidak Sesuai
1	Api A1	Ada	Ada	✓	
2	Api A2	Tidak Ada	Tidak Ada	✓	

3	Api B1	Tidak Ada	Tidak Ada	✓
4	Api B2	Ada	Ada	✓

Berdasarkan data pada tabel diatas dapat diketahui bahwa semua data sensor api yang ditampilkan di aplikasi sesuai dengan data yang disimpan di *realtime database* sehingga dapat disimpulkan bahwa *realtime database* sensor api *flame detectors* memiliki *accuracy rate* sebesar 100%.

Tabel 8. Hasil Pengujian *Database* Sensor Asap MQ-7

No	Data	Data Realtime Database	Data Aplikasi	Sesuai	Tidak Sesuai
1	Asap A1	Ada	Ada	✓	
2	Asap A2	Tidak Ada	Tidak Ada	✓	
3	Asap B1	Tidak Ada	Tidak Ada	✓	
4	Asap B2	Ada	Ada	✓	

Berdasarkan data pada tabel diatas dapat diketahui bahwa semua data sensor asap yang ditampilkan di aplikasi sesuai dengan data yang disimpan di *realtime database* sehingga dapat disimpulkan bahwa *realtime database* sensor asap MQ-7 yang digunakan memiliki *accuracy rate* sebesar 100%.

■ Tabel 9. Hasil Pengujian *Database* Sensor Suhu LM35

No.	Data	Data Realtime Database	Data Aplikasi	Sesuai	Tidak Sesuai
1	Suhu A1	40°C	40°C	✓	
2	Suhu A2	34°C	34°C	✓	
3	Suhu B1	34°C	34°C	✓	
4	Suhu B2	39°C	39°C	✓	

Berdasarkan data pada tabel diatas dapat diketahui bahwa semua data sensor suhu yang ditampilkan di aplikasi sesuai dengan data yang disimpan di *realtime database* sehingga dapat disimpulkan bahwa *realtime database* sensor suhu LM35 yang digunakan memiliki *accuracy rate* sebesar 100%.

b. Pengujian *Compatibility* Aplikasi

Pada hasil pengujian ini akan dilihat apakah aplikasi “Siaga Api” dapat berjalan pada berbagai *environment* yang berbeda. Hasil dari pengujian *compatibility* aplikasi ditunjukkan pada Tabel 10, dan Tabel 11.

■ Tabel 10. Hasil Pengujian Instalasi Aplikasi “Siaga Api”

No.	Perangkat	Proses Instalasi	
		Berhasil	Tidak Berhasil
1	Samsung Galaxy A32	✓	
2	Oppo A57	✓	
3	Xiaomi Redmi Note 10	✓	
4	Samsung Galaxy A20	✓	
5	Asus Zenfone 5	✓	

Berdasarkan data pada tabel diatas dapat diketahui bahwa aplikasi “Siaga Api” dapat diinstal pada 5 perangkat handphone berbeda yang diuji cobakan. Hal ini menunjukkan bahwa aplikasi ini kompatibel untuk diinstal di berbagai perangkat dengan *environment* berbeda. Pada pengujian ini maka skor yang didapat adalah 5 poin.

■ **Tabel 11.** Perangkat *Handphone* yang Digunakan

No	Halaman Aplikasi	Perangkat				
		Samsung Galaxy A32	Oppo A57	Xiaomi Redmi Note 10	Samsung Galaxy A20	Asus Zenfone 5
1	Splash Screen	✓	✓	✓	✓	✓
2	Login	✓	✓	✓	✓	✓
3	Registrasi	✓	✓	✓	✓	✓
4	Lupa Password	✓	✓	✓	✓	✓
5	Verifikasi	✓	✓	✓	✓	✓
6	Status	✓	✓	✓	✓	✓
7	Api	✓	✓	✓	✓	✓
8	Asap	✓	✓	✓	✓	✓
9	Suhu	✓	✓	✓	✓	✓
10	Tentang	✓	✓	✓	✓	✓
11	Tentang Aplikasi	✓	✓	✓	✓	✓
12	Tentang Kami	✓	✓	✓	✓	✓

Berdasarkan data pada tabel diatas dapat diketahui bahwa semua perangkat yang diuji cobakan berhasil menjalankan aplikasi dengan lancar dan dapat membuka semua halaman aplikasi. Hal ini menunjukkan bahwa aplikasi ini kompatibel untuk dijalankan di berbagai perangkat dengan *environment* berbeda. Pada pengujian ini maka skor yang didapat adalah 60 poin.

Berdasarkan data hasil pengujian pada Tabel 10 dan Tabel 11 diatas, maka selanjutnya perlu menghitung persentase kelayakan aplikasi dengan menggunakan perhitungan sebagai berikut.

$$\begin{aligned} \text{Persentase keberhasilan (\%)} &= \frac{5+60}{65} \times 100\% \\ &= 100\% \end{aligned}$$

Berdasarkan perhitungan tersebut maka hasil persentase kelayakan yang didapatkan hasilnya adalah 100% maka aplikasi “Siaga Api” memenuhi standar aspek *compatibility* dengan kategori “sangat layak”.

c. Pengujian *Usability* Aplikasi

Pada pengujian ini didapatkan hasil responden sebanyak 15 responden. Dari hasil jawaban responden terhadap kuesioner kemudian akan dirangkum dan akan dihitung skor untuk masing-masing pertanyaan. Hasil dari pengujian *usability* aplikasi ditunjukkan pada Tabel 12, dan Tabel 13.

■ **Tabel 12.** Rangkuman Hasil Jawaban Responden Terhadap Kuesioner

Responden	Skor Asli									
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	4	2	5	2	4	2	5	1	5	1
2	4	1	4	1	4	1	5	1	5	1
3	5	1	5	1	5	1	5	1	5	1
4	5	2	5	3	5	2	5	3	5	2
5	4	1	3	1	4	4	5	1	2	1
6	4	3	4	2	4	3	4	3	3	2
7	5	1	4	1	5	1	5	1	5	1
8	4	1	5	1	5	1	5	1	5	2
9	5	1	5	1	5	1	5	1	5	1
10	5	1	5	1	5	1	5	1	5	1
11	5	2	5	2	5	2	5	2	5	2
12	5	2	5	2	5	2	5	2	5	2
13	4	3	4	4	5	1	3	4	3	1

■ **Tabel 13.** Perhitungan Persentase Hasil Kuesioner

Responden	Skor Hasil Hitung										Jumlah	Skor (%)
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10		
1	3	3	4	3	3	3	4	4	4	4	35	88
2	3	4	3	4	3	4	4	4	4	4	37	93
3	4	4	4	4	4	4	4	4	4	4	40	100
4	4	3	4	2	4	3	4	2	4	3	33	83
5	3	4	2	4	3	1	4	4	1	4	30	75
6	3	2	3	3	3	2	3	2	2	3	26	65
7	4	4	3	4	4	4	4	4	4	4	39	98
8	3	4	4	4	4	4	4	4	4	3	38	95
9	4	4	4	4	4	4	4	4	4	4	40	100
10	4	4	4	4	4	4	4	4	4	4	40	100
11	4	3	4	3	4	3	4	3	4	3	35	88
12	4	3	4	3	4	3	4	3	4	3	35	88
13	3	2	3	1	4	4	2	1	2	4	26	65
14	3	3	3	3	4	3	4	3	4	3	33	83
15	3	3	3	3	3	3	3	3	4	3	31	78
Skor Rata-Rata (%)												86

Berdasarkan data pada tabel diatas maka didapatkan bahwa skor tertinggi adalah 100 berdasarkan jawaban dari responden 3, 9, dan 10. Sedangkan skor terendah adalah 65 berdasarkan jawaban dari responden 6 dan 13. Dan rata-rata persentase skor keseluruhan adalah 86. Rata-rata persentase yang telah didapatkan selanjutnya dikonversikan sesuai standar ISO/EIC 25010. Dengan nilai standar kelayakan sebesar 86% maka aplikasi “Siaga Api” memenuhi standar aspek *usability* dengan kategori “sangat layak”.

d. Pengujian *Performance Efficiency* Aplikasi

Pada pengujian ini didapatkan persentase pemakaian CPU *usage* serta *memory device* pada saat menjalankan aplikasi “Siaga Api”. Pengujian ini dilakukan selama 1 menit 4 detik melalui *simulation device* Pixel 3, API Level 28. Grafik hasil performansi CPU *usage* dan *memory* saat menjalankan aplikasi “Siaga Api” ditunjukkan pada Gambar 8.



■ **Gambar 8.** Grafik Hasil Performansi CPU Usage dan Memory

Hasil dari pengujian *performance efficiency* aplikasi ditunjukkan pada Tabel 14, dan Tabel 15.

■ **Tabel 14.** Hasil Pengujian CPU Usage

No.	Menit Ke-	Penggunaan CPU (%)
1	0.05	0,625
2	0.1	1,911
3	0.15	5,66
4	0.2	14,103
5	0.25	8,075
6	0.3	0,637
7	0.35	0,621
8	0.4	4,403
9	0.45	0,625
10	0.5	0,629
11	0.55	0,633
12	1	0,629
Rata-Rata		3,213

Berdasarkan data pada tabel diatas rata-rata *performance CPU usage* dari aplikasi “Siaga Api” adalah sebesar 3,213%. Nilai rata-rata tersebut masih berada dibawah batas aman yang ditetapkan oleh standar *little eye tools* untuk CPU usage, yaitu sebesar 15%.

■ **Tabel 15.** Hasil Penggunaan Memory

No.	Menit Ke-	Penggunaan Memory (KB)
1	0.05	74,912
2	0.1	1,45,816
3	0.15	1,36,608
4	0.2	1,89,448
5	0.25	2,02,060
6	0.3	1,63,760
7	0.35	1,61,928
8	0.4	1,65,632
9	0.45	1,70,056
10	0.5	1,64,360

11	0.55	1,68,536
12	1	1,67,436
Rata-Rata		1,59,213

Berdasarkan data pada tabel diatas dapat diketahui bahwa rata-rata penggunaan *memory* pada aplikasi “Siaga Api” yaitu sebesar 159,213 KB. Nilai tersebut dapat dikatakan cukup kecil untuk penggunaan *memory* sebuah aplikasi. Dalam pengujianya pun aplikasi tidak mengalami *memory leaks*, yaitu suatu keadaan dimana *resource* dari aplikasi yang dialokasikan pada *memory* tidak mampu dibebaskan sehingga dapat menyebabkan sistem *crash*.

e. Pengujian Sistem Notifikasi Peringatan

Pengiriman notifikasi peringatan dilakukan secara otomatis oleh *cloud function firebase*, terdapat 2 fungsi yang di *deploy* untuk menjalankan perintah otomatis pengiriman notifikasi peringatan kepada pengguna. Kedua fungsi tersebut tertera pada Tabel 16 dibawah ini.

■ **Tabel 16.** *Cloud Function* Sistem Notifikasi Peringatan

No.	Nama Fungsi	Kegunaan
1	sendNotif	Mengirimkan <i>push notification</i> kepada pengguna jika terjadi perubahan status di Node A
2	sendNotifB	Mengirimkan <i>push notification</i> kepada pengguna jika terjadi perubahan status di Node B

Kedua fungsi tersebut akan menjalankan tugasnya ketika ada trigger yaitu perubahan data status pada *realtime* database. Pada pengujian ini kedua fungsi tersebut diukur waktu eksekusi fungsinya dengan menggunakan *function logs* ketika ada fungsi yang dieksekusi. Berdasarkan nilai yang terukur pada *function logs Firebase* maka didapatkan nilai waktu pengeksekusian setiap fungsi pada saat pengujian berlangsung. Nilai tersebut ditunjukkan pada Tabel 17 dibawah ini.

■ **Tabel 17.** Waktu Eksekusi Fungsi Pada *Cloud Function* Firebase

No.	Fungsi	Waktu Eksekusi				Rata-Rata Waktu Eksekusi
		Status		Kebakaran Level 1	Kebakaran Level 2	
Aman	Waspada					
1	sendNotif	415ms	456ms	320ms	329ms	380ms
2	sendNotifB	348ms	332ms	320ms	365ms	341ms

Waktu eksekusi yang tertera pada tabel diatas merupakan waktu *Firebase* untuk memproses atau menjalankan perintah yang tertera dalam *cloud function*. Fungsi pengiriman *push notification* untuk Node A (*sendNotif*) memiliki rata-rata waktu eksekusi sebesar 380ms, dan fungsi pengiriman *push notification* untuk Node B (*sendNotifB*) memiliki rata-rata waktu eksekusi sebesar 341ms. Rata-rata waktu eksekusi berada pada kisaran 100ms – 400ms, sehingga dapat dikategorikan *acceptable* berdasarkan standar QRTi MySQL *Enterprise* yang artinya nilai tersebut dapat diterima.

KESIMPULAN

1. *Mobile application* pada sistem pendeteksi dini kebakaran lahan berbasis Android merupakan sebuah aplikasi yang didesain untuk pada pengguna Android dan berfungsi untuk melakukan pemantauan lahan yang menampilkan data-data ada atau tidak adanya api dari sensor api *flame detectors*, ada atau tidak adanya asap dari

2. sensor asap MQ-7, dan besaran nilai suhu dari sensor suhu LM35, yang jika ada perubahan status maka *user* akan menerima *push notification* berupa perubahan status “aman”, “waspada”, “kebakaran level 1”, atau “kebakaran level 2”.
3. Kualitas aplikasi “Siaga Api” dinilai berdasarkan aspek *compatibility*, *usability*, dan *performance efficiency*. Pada aspek *compatibility* aplikasi memiliki persentase skor kelayakan sebesar 100%, sehingga berdasarkan standar ISO/EIC 25010 termasuk kedalam kategori sangat layak. Dari aspek *usability* memiliki persentase skor kelayakan sebesar 86% sehingga juga termasuk kedalam kategori sangat layak, dan dari *performance efficiency* rata-rata persentase CPU *usage* sebesar 3,213% nilai ini masih berada dalam batas aman standar yg ditetapkan *little eye tools* yaitu 15% dan penggunaan *memory* saat menjalankan aplikasi pun terbilang cukup kecil yaitu hanya sebesar 159,213 KB.
4. Performansi layanan database yang digunakan pada sistem ini memiliki kemampuan yang cukup baik dalam pengiriman *push notification*. Pada Node A rata-rata waktu eksekusinya adalah 380ms, dan pada Node B rata-rata waktu eksekusinya adalah 341ms. Rata-rata waktu eksekusi tersebut berada pada range 100ms – 400ms sehingga dapat dikatakan *acceptable* berdasarkan standar QRTi MySQL *Enterprise* karena pada sistem ini data yang tersimpan secara *realtime* dapat langsung diketahui oleh *client*.

DAFTAR PUSTAKA

- [1] SiPongi. “Rekapitulasi Luas Kebakaran Hutan dan Lahan (Ha) Per Provinsi Di Indonesia Tahun 2016- 2021”. Internet : http://sipongi.menlhk.go.id/hotspot/luas_kebakaran. [4 Juli 2021]
- [2] PRIMs. “Perencanaan Restorasi di 7 Provinsi Prioritas”, 2020. Internet : <https://prims.brg.go.id/dasbor/restorasi>. [4 Juli 2021]
- [3] BNPB (Badan Nasional Penanggulangan Bencana). “Definisi Bencana”. Internet : <https://bnpb.go.id/definisi-bencana>. [4 Juli 2021]
- [4] Sasmoko, D. & Mahendra, A.: Rancang Bangun Sistem Pendeteksi Kebakaran Berbasis IoT dan SMS Gateway Menggunakan Arduino. Indonesia: Semarang. 2017.
- [5] Siswanto, dkk.: *Prototype Wireless Sensor Network (WSN) Sistem Pendeteksi Dini Kebakaran Hutan*. Indonesia: Serang. 2020.
- [6] Joniwarta, dkk.: *Sistem Monitoring Kebakaran Berbasis SMS Gateway*. Indonesia: Jakarta. 2019.