

MONITORING CUACA PADA ANDROID MENGGUNAKAN NODE-RED, OPENWEATHER DAN SIGNAL4

Dewi Laksmiati
Universitas Bina Sarana Informatika
(Naskah diterima: 1 April 2024, disetujui: 25 April 2024)

Abstract

Weather forecasts are made by processing as much data as possible using the science of meteorology against data collected about the current state of the atmosphere (especially temperature, humidity and wind). This weather forecast is important because it can warn of future weather conditions so that people can plan and anticipate activities that will be carried out related to the weather. This paper describes the implementation of weather notifications on android devices. Where the data was originally retrieved by Node-RED from OpenWeather via API and JSON. Where weather data is specific data for a certain location and time span. Then by Node-RED the data is passed through SIGNAL4 to the Android device.

Keywords: *Weather, OpenWeather, Node-RED, SIGNAL4*

Abstrak

Prakiraan cuaca dibuat dengan cara melakukan pemrosesan sebanyak mungkin data menggunakan ilmu meteorologi terhadap data yang dikumpulkan tentang keadaan atmosfer saat ini (khususnya suhu, kelembaban, dan angin). Prakiraan cuaca ini penting dikarenakan dapat memperingatkan kondisi cuaca di masa depan agar orang-orang dapat melakukan perencanaan dan antisipasi terhadap kegiatan yang akan dilakukan terkait dengan cuaca. Pada penulisan ini dijabarkan implementasi notifikasi cuaca pada perangkat android. Dimana data yang awalnya diambil oleh Node-RED dari OpenWeather melalui API dan JSON. Dimana data cuaca merupakan data spesifik untuk lokasi dan rentang waktu tertentu. Kemudian oleh Node-RED data diteruskan melalui SIGNAL4 ke perangkat Android.

Kata Kunci: Cuaca, OpenWeather, Node-RED, SIGNAL4

I. PENDAHULUAN

Indonesia merupakan salah satu negara yang memiliki iklim tropis yang hanya memiliki 2(dua) musim, yaitu hujan dan kemarau. Di masa lalu cuaca sangat mudah

terprediksi, dimana 6 bulan merupakan musim kemarau, dan 6 bulan lainnya musim hujan. Namun semenjak terjadi pemanasan global, cuaca menjadi tidak pasti. Terkadang di saat musim kemarau marak turun hujan sampai

pada suatu masa timbul istilah “kemarau basah”, begitu juga sebaliknya saat musim hujan yang terjadi minim curah hujan dan panas. Terkadang juga terjadi perbedaan cuaca dratis, misalnya sebuah daerah panas terik, daerah berdekatan malah hujan deras Hal ini diakibatkan banyak faktor yang mempengaruhi secara global, mulai dari faktor angin, suhu, dan hal ini tak lepas juga faktor cuaca di daerah sekitarnya.

Ketidakpastian cuaca ini menimbulkan masalah bagi semua kalangan, mulai dari seorang ibu rumah tangga yang baru saja selesai mencuci, ternyata tidak bisa menjemur karena hujan deras. Hingga kalangan professional yang tadinya hendak naik sepeda motor untuk mempermudah perjalanan, ternyata kesulitan karena terguyur hujan deras.

Oleh karena itu, diperlukan sebuah monitoring cuaca yang seluruh orang dapat mengaksesnya. Bukan perkiraan cuaca. Karena prakiraan cuaca kita tahu sendiri bahwa kadang itu tidak tepat.

Dalam penulisan ini akan dirancang sebuah sistem monitoring cuaca secara real-time dengan mengolah data yang didapatkan dari OpenWeatherMap. Pengolahan data menggunakan Node-RED yang pada akhirnya akan mengirimkan data ke SIGNAL4 untuk

memudian ditampilkan sebagai notifikasi di perangkat Android.

II. KAJIAN TEORI

2.1 *Open Weather Map*

OpenWeatherMap merupakan layanan cuaca yang berdasar pada platform VANE *Geospatial Data Science* untuk mengumpulkan, memproses, dan mendistribusikan informasi tentang planet kita melalui *tools* dan API yang mudah digunakan.[1]

Platform VANE memungkinkan OpenWeatherMap untuk memproses dan mendistribusikan miliaran titik sumber data cuaca setiap detiknya. Jumlah server yang memproses lebih dari 10.000 panggilan data per detik yang diterima dari layanan OpenWeatherMap. Hal ini dimungkinkan karena teknologi *Big Data* mutakhir yang diterapkan pada platform VANE.



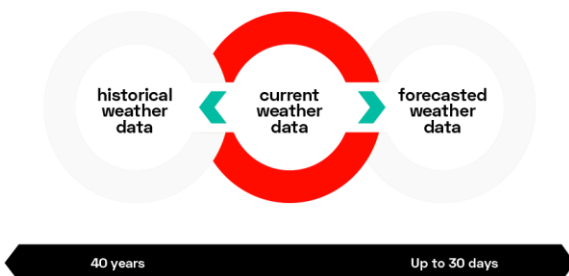
Gambar 1. Jangkauan OpenWeatherMap [2]

Perkembangan teknologi saat ini membuat biaya transaksi dan biaya penyimpanan data turun dengan cepat. Hal ini membuka peluang luar biasa bagi perusahaan IT kecil dan

menengah. Dimana beberapa tahun yang lalu biaya pengembangan sistem berskala besar yang mendukung beberapa terabyte dan memuat ribuan transaksi per detik benar-benar tidak terjangkau bagi sebagian pihak. Sekarang semuanya berubah. Dengan tersedianya *Big Data*, data cuaca OpenWeatherMap terus gratis dan terbuka.

OpenWeatherMap memiliki OpenAPI yang dapat digunakan pengembang aplikasi untuk menyematkan informasi cuaca ke dalam berbagai aplikasi termasuk aplikasi cuaca web dan seluler, solusi untuk periklanan, pertanian, olahraga, dan banyak lainnya. Bagi mereka yang hanya ingin mengetahui kondisi cuaca untuk lokasi mereka, kami menyediakan cuaca terkini, ramalan cuaca, peta, dan banyak grafik di OpenWeatherMap untuk umum.

OpenWeatherMap menyediakan rangkaian data yang lengkap, mulai 40 tahun sebelumnya, hingga prakiraan cuaca 30 hari ke depan

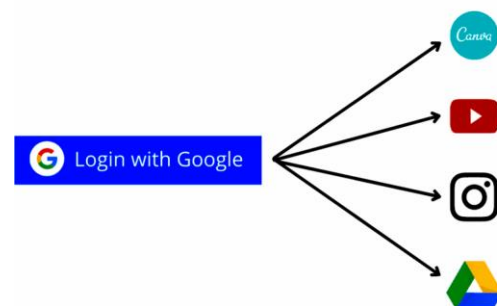


Gambar 2. Ketersediaan Data di OpenWeatherMap [2]

2.1 API (*Application Programming Interface*)

API kependekan dari *Application Programming Interface* merupakan sebuah perangkat lunak yang dapat menerima panggilan atau permintaan dari perangkat lunak lainnya seperti aplikasi dan website yang memberikan pelayanan. Dengan kata lain API merupakan sebuah perangkat lunak yang mengizinkan atau menghubungkan dua aplikasi.[3]

API juga menjadi teknologi yang memungkinkan untuk menggunakan fitur *one-click login/signup* (*Single Sign-On*) menggunakan akun Microsoft, Google, Facebook untuk login ke aplikasi lainnya



Gambar 3. *Single login* via API [4]

Ketika sebuah aplikasi digunakan di komputer atau HP, aplikasi tersebut akan menghubungkan ke internet dan mengirim data ke sebuah server. Server tersebut kemudian akan mengambil tindakan sesuai yang

diperlukan dan mengirim hasilnya kembali ke perangkatmu.

API sendiri juga dapat digunakan untuk berkomunikasi dengan berbagai bahasa pemrograman berbeda yang tentunya akan sangat memudahkan developer dalam mengembangkan aplikasi. Selain itu, developer juga dapat mengambil data yang sudah tersedia di platform lain tanpa perlu menyediakan seluruh data sendiri.

2.1.1 Jenis API

Terdapat empat jenis API yang dikategorikan berdasarkan hak aksesnya: [4]

1. *Private API*

Private API adalah sebuah interface pemrograman yang tidak terbuka untuk umum. Pada umumnya, jenis ini diciptakan untuk memenuhi kebutuhan pengembangan aplikasi secara internal. Jenis ini berperan sebagai interface bagian *front end* yang digunakan untuk mengakses data dan fungsi aplikasi di *back end*.

2. *Public API*

Public API, atau juga disebut *open API* adalah sebuah *programming interface* yang dapat diakses secara publik. *Developer* dapat mengakses sistem pemrograman dibalik sebuah aplikasi atau layanan web melalui public

API untuk mengembangkan aplikasi mereka sendiri dengan lebih cepat.

3. *Partner API*

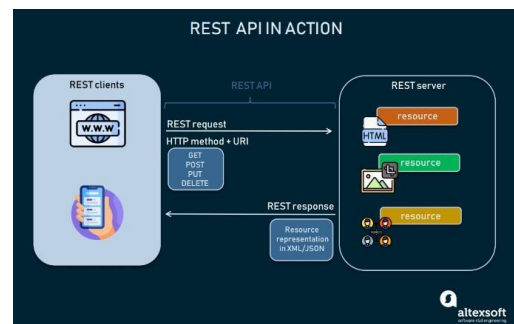
Partner API merupakan jenis *interface* yang dapat diakses oleh pihak-pihak tertentu yang telah ditunjuk sebagai rekanan bisnis dari pihak pemilik aplikasi atau *web service*. Jenis ini tidak tersedia untuk umum dan memerlukan kredensial tertentu untuk mengaksesnya.

4. *Composite API*

Composite API adalah jenis *interface* yang terdiri dari gabungan berbagai jenis data dari berbagai *server* dan *hosting* dalam satu tempat. Tipe API satu ini sangat berguna bagi *developer* karena mereka dapat mengakses banyak informasi dalam satu tempat saja.

2.1.2 Cara Kerja API

Application Programming Interface (API) memiliki fungsi utama sebagai perantara yang menghubungkan aplikasi *client side* dengan aplikasi *server side*. Di bawah ini ilustrasi mengenai cara kerja API.



Gambar 4. Koneksi API

1. Aplikasi Mengakses API

Saat sebuah informasi diakses dalam sebuah aplikasi, maka aplikasi yang sedang diakses tersebut akan mengakses data pada layanan terkait. Contohnya disaat melakukan pemesanan pulsa melalui aplikasi *marketplace*, maka aplikasi tersebut akan mengakses API yang menjadi perantara dengan operator seluler yang telah tersambung

2. API Membuat *Request* ke *Server*

Kemudian, sehabis aplikasi berhasil mengakses API, maka melalui API permintaan (*request*) akan diteruskan ke *server*. Sehingga pembelian pulsa melalui aplikasi *marketplace* tersebut tidak mengakses data di system lokal mereka, namun dari *server* operator seluler

3. *Server* Merespon *Request* dari API

Setelah *server* menerima *request* dari API dan menerjemahkannya, *server* kemudian mencari data yang diminta oleh *client*. Ketika data sudah ditemukan, *server* akan mengembalikan informasi seperti status nomor ponsel yang akan diisi pulsa. Nominal dan harga pulsa yang tersedia ke API untuk diteruskan ke aplikasi.

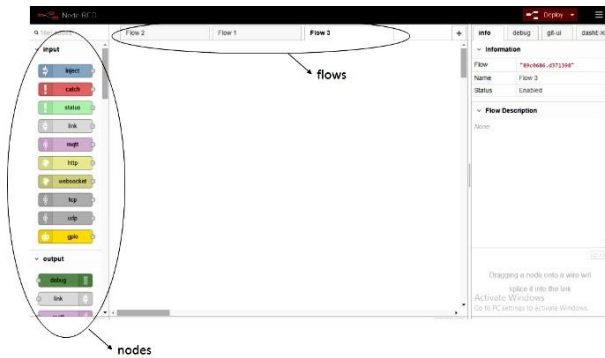
4. API Meneruskan Respon ke Aplikasi

Setelah API menerima data dari *server*, kemudian selanjutnya adalah meneruskan informasi tersebut ke aplikasi pengguna.

2.3 Node-RED

Node-RED merupakan sebuah *tool* berbasis *browser* untuk membuat aplikasi secara visual. Node-RED cukup dikenal dalam implementasi *Internet of Things* (IoT) dimana lingkungan pemrograman visualnya mempermudah penggunaannya untuk membuat aplikasi sebagai “*flow*”.

Node-RED adalah bahasa pemrograman berbasis visual. Dimana cara membuat aplikasi sebagai tidak menggunakan barisan koding seperti pemrograman pada umumnya. Node-RED memiliki fokus pada penyederhanaan ‘*wiring together*’ dari blok kode untuk melakukan tugas-tugas yang diinginkan. Node-RED menggunakan pendekatan pemrograman visual yang memungkinkan pengembang untuk menghubungkan blok kode yang telah ditetapkan, yang dikenal sebagai ‘*node*’, bersama-sama untuk melakukan tugas. *Node* yang terhubung, biasanya kombinasi dari *node* input, *node* pemrosesan dan output *node*, ketika dihubungkan bersama, membentuk ‘*flow*’.

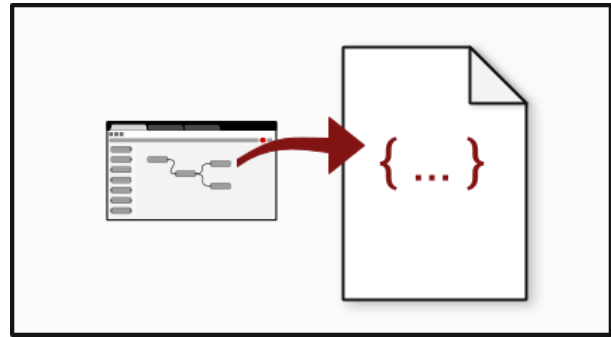


Gambar 5. Komponen Dalam Antarmuka Node-RED[6]

Node-RED dapat digunakan untuk menghubungkan perangkat keras, API, dan layanan *online* dengan cara baru dan menarik. Ia menyediakan editor berbasis *browser* yang memudahkan untuk menyatukan aliran menggunakan berbagai node di palet yang dapat digunakan untuk *runtime* dalam satu klik.[8]

Pada awalnya Node-RED dikembangkan sebagai proyek *open source* di IBM pada akhir tahun 2013, Node-RED dikembangkan untuk memenuhi kebutuhan akan kecepatan dalam menghubungkan *hardware* dan *device* ke layanan web dan *software* lainnya, sebagai semacam perekat untuk IoT. Node-RED kemudian dengan cepat berkembang menjadi tujuan umum *tool* pemrograman IoT. Node-RED dengan cepat telah mengembangkan basis pengguna yang signifikan dan terus berkembang serta komunitas pengembang aktif yang menyumbangkan simpul baru yang di-

simpan dalam bentuk JSON dimana memungkinkan digunakan kembali oleh *programmer* lain



Gambar 6. Konversi *Flow* to JSON [7]

Meskipun Node-RED pada awalnya dirancang untuk bekerja dengan perangkat *Internet of Things*, saat ini Node-RED telah menjadi berguna untuk berbagai aplikasi.

Dalam penulisan ini, Node-RED digunakan dalam memproses data yang masuk kemudian dikirim ke aplikasi SIGNAL4

Node-RED menyediakan *flow editor* berbasis browser yang dapat digunakan untuk merangkai *device hardware*, API dan layanan online menggunakan representasi *nodes* dalam satu *palette*. *Flow* dapat dibangun ke dalam *runtime* hanya dengan satu klik saja.

Adapun Fungsi Javascript dapat dibuat dengan editor menggunakan *rich text editor* dan pustaka bawaan yang dapat digunakan untuk menyimpan fungsi, *template*, *flow* agar bisa digunakan kembali.[9]

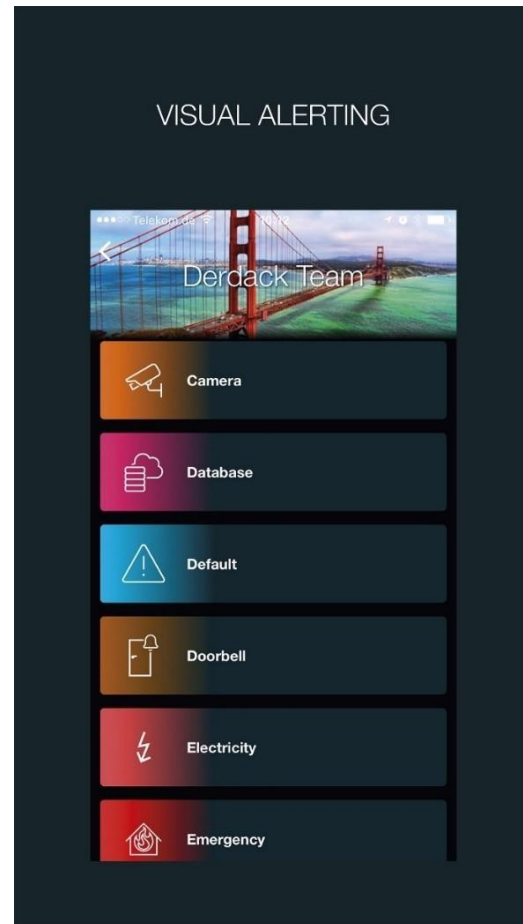
2.4 SIGNAL4

SIGNAL4 merupakan adalah SaaS (*Software as a Service*) yang melayani pemberitahuan peringatan melalui perangkat seluler untuk menginformasikan tim operasi dalam peringatan, insiden, dan permintaan layanan yang mendesak.



Gambar 7. Interaksi Perangkat yang
Dihubungkan SIGNAL4 [10]

Ketika terjadi kegagalan sistem, atau satu hal yang dianggap darurat. SIGNAL4 akan menjadi ujung tombak dalam menjembatani sistem TI, perangkat, dan sensor IoT ke sisi pengguna yaitu teknisi, staf TI, dan pekerja lapangan. Layanan ini menambahkan peringatan seluler waktu nyata ke layanan, sistem, dan proses dalam waktu singkat.[11]



Gambar 8. Antarmuka Aplikasi SIGNAL4 [12]

SIGNAL4 memberikan notifikasi melalui *mobile push*, teks, dan suara yang berkelanjutan dengan *acknowledgement*, pelacakan, dan eskalasi. Penjadwalan tugas terintegrasi memastikan orang yang tepat diberi tahu pada waktu yang tepat.

III. METODE PENELITIAN

3.1 Metode Observasi

Melakukan pengumpulan data-data dengan cara mengamati serta mencatat secara sistematis tentang perangkat dan aplikasi yang

digunakan dalam konfigurasi dalam prak-tek langsung.

3.2 Metode Studi Pustaka

Yaitu menggunakan literatur baik dalam bentuk media online, artikel atau buku bacaan yang berkaitan dengan penyusunan artikel ini.

3.3 Metode Pengembangan Jaringan

1. Analisa Kebutuhan

Analisa akan dilakukan melalui beberapa tahapan, yaitu

- a. Observasi langsung
- b. Memahami semua kondisi kebutuhan di lapangan terkait kebutuhan *monitoring* cuaca hujan.
- c. Analisa hasil observasi.

2. Desain

Perancangan dilakukan melalui beberapa tahapan, yaitu:

- a. Pemilihan sumber data untuk cuaca
- b. Pemilihan aplikasi untuk pengolahan API
- c. Pemilihan OS untuk instalasi *tools* aplikasi pengolahan API
- d. Pemilihan aplikasi untuk memberikan notifikasi

3. Testing

Melakukan *monitoring* setiap perubahan cuaca menjadi hujan, dengan hasil akhir muncul notifikasi di perangkat Android

4. Implementasi

Untuk menjalankan *monitoring* cuaca yang bersumber dari OpenWeatherMap ke SIGNAL4 diperlukan langkah berikut.

a. Registrasi OpenWeatherMap untuk Mendapatkan API Key

- b. Registrasi di OpenWeatherMap.com untuk mendapatkan API-key. Dimana API-Key dibutuhkan untuk mendapatkan konten cuaca yang diinginkan di OpenWeatherMap

c. Registrasi SIGNAL4

Registrasi akun SIGNAL4 untuk dapat menerima masukan data dan meneruskannya sebagai notifikasi di aplikasi Android

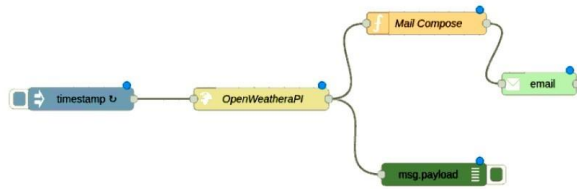
d. Instalasi Node-RED di Linux Ubuntu

Instalasi Node-RED pada Linux Ubuntu di mesin virtual (VM) pada sistem lokal. Linux Ubuntu di-*install* dengan spesifikasi:

Processor : 4 core
RAM : 8 GB
Hard Disk : 100 GB

e. Pembuatan Flow di Node-RED

Membuat *flow* untuk menarik data dari OpenWeatherMap menggunakan API, kemudian diolah untuk kemudian diteruskan ke alamat email khusus yang didapatkan dari SIGNAL4. Oleh system SIGNAL4 kemudian akan diteruskan ke aplikasi SIGNAL4



Gambar 9. Flow Node-RED Untuk Monitoring Cuaca

IV. HASIL PENELITIAN

Pada bab ini akan dijelaskan mengenai hasil pengujian.

a.1. Monitoring

Melakukan *monitoring* setiap perubahan cuaca menjadi hujan, API di setting untuk memperoleh data terkait hujan dari OpenWeatherMap. Apabila hujan maka Node-RED akan meneruskan data ke SIGNAL4 dan muncul notifikasi di perangkat Android

a.2. Hasil Pengujian

Dari hasil monitoring menggunakan agent selama 2 hari pada didapatkan hasil cuaca hujan sebagai berikut.

| Hari | Jam | Status |
|----------|-------|---------------|
| Hari – 1 | 14:30 | Mulai Hujan |
| Hari – 1 | 16:10 | Selesai Hujan |
| Hari – 1 | 20:23 | Mulai Hujan |
| Hari – 1 | 23:32 | Selesai Hujan |
| Hari – 2 | 17:40 | Mulai Hujan |
| Hari – 2 | 19:45 | Selesai Hujan |

V. KESIMPULAN

Dari pengujian monitoring cuaca diperoleh kesimpulan sebagai berikut :

1. Setiap perubahan cuaca terkait hujan didapatkan *update*-nya dari OpenWeatherMap
2. Node-RED memproses data yang didapatkan dari API, untuk kemudian diteruskan ke SIGNAL4 menggunakan email khusus
3. Aplikasi SIGNAL4 memberikan notifikasi setiap mulai dan berakhirnya hujan.
4. Aplikasi dapat digunakan untuk memonitor cuaca terkait hujan

DAFTAR PUSTAKA

- Anonim, 2020, “OpenWeatherMap”, AgriTeach 4.0.
<https://www.agriteach.hu/en/content/openweathermap>
 (diakses pada 21 November 2021 09:21 WIB)
- Anonim, 2020, “Current Weather and Forecast - OpenWeatherMap”, Open Weather.
<https://openweathermap.org/>
 (diakses pada 19 November 2021 19:21 WIB)
- Anonim, 2019, “Apa itu API?”, NetMonk.
<https://netmonk.id/apa-itu-api>
 (diakses pada 19 November 2021 16:21 WIB)
- Yonata, Jefri, 2021, “API: Penjelasan Lengkap, Jenis, dan Cara Kerjanya”, Dewa Web Blog,
<https://www.dewaweb.com/blog/api-adalah/>
 (diakses pada 22 November 2021 15:11 WIB)

Anonim, 2021 “REST API: Key Concepts, Best Practices, and Benefits”, Altex Soft Blog,
<https://www.altexsoft.com/blog/rest-api-design/>
(diakses pada 22 November 2021 17:46 WIB)

Anonim, 2021 “Node-RED - Pengiriman dan Penerimaan Data”, Antares,
<https://antares.id/id/node-red.html>
(diakses pada 23 November 2021 14:25 WIB)

Anonim, 2019 “Sekilas Tentang Node-RED”, IoTQu
<https://iotqu.blogspot.com/2019/02/sekilas-tentang-Node-RED.html>
(diakses pada 23 November 2021 17:16 WIB)

Anonim, 2021, “ Node-RED Low-code programming for event-driven applications”, Node-RED,
<https://www.nodered.org/>
(diakses pada 24 November 2021 16:42 WIB)

Permana, Yana, “Belajar Membangun Flow Aplikasi Internet of Things dengan IBM

Bluemix Node-RED”, Codepolitan
<https://www.codepolitan.com/belajar-membangun-flow-aplikasi-internet-of-things-dengan-ibm-bluemix-Node-RED-5857df6755f59-14>
(diakses pada 24 November 2021 18:29 WIB)

Anonim, 2021,” Mobile Alerting and Anywhere Incident Response App”, SIGNAL4 Feature Overview,
<https://www.SIGNAL4.com/mobile-alert-notification-app/>
(diakses pada 24 November 2021 19:15 WIB)

Anonim, 2020, “SIGNAL4 Reviews & Product Details”, G2 Product Reviews,
<https://www.g2.com/products/SIGNAL4-derdack/reviews>.
(diakses pada 25 November 2021 10:17 WIB)

Anonim, 2021, SIGNAL4 “Mobile alerting via app, text and voice”, SIGNAL4
<https://www.derdack.com/SIGNAL4-mobile-alerting-app/>
(diakses pada 25 November 2021 13:17 WIB).