

# IMPLEMENTASI PROTOKOL MQTT PADA GPS-NEO6M SEBAGAI RANCANG BANGUN MONITORING SEPEDA MOTOR

*by* Muhammad Fikri

---

**Submission date:** 18-Dec-2023 12:02PM (UTC+0300)

**Submission ID:** 2261782150

**File name:** jurnal\_malih\_123\_revisi\_senin.docx (1.44M)

**Word count:** 2218

**Character count:** 14206

# IMPLEMENTASI PROTOKOL MQTT PADA GPS-NEO6M SEBAGAI RANCANG BANGUN MONITORING SEPEDA MOTOR

Aldi Viantara<sup>1</sup>, Mochamad Alfian Rosid<sup>2</sup>, Azmuri Wahyu Azinar<sup>3</sup>

<sup>1,2,3</sup>. Informatika, Universitas Muhammadiyah Sidoarjo

**Abstract**—kendaraan menjadi kebutuhan esensial yang memfasilitasi mobilitas manusia. Namun, ancaman keamanan seperti pencurian kendaraan bermotor menjadi masalah yang signifikan. Teknologi keamanan IoT, seperti fitur darurat yang mematikan kendaraan dan pelacakan GPS menggunakan protokol HTTPS, menjadi solusi yang relevan. Penelitian sebelumnya menunjukkan keunggulan protokol MQTT dalam IoT berdasarkan selisih latency yang lebih rendah. Penelitian ini bertujuan memanfaatkan protokol MQTT dalam membangun sistem monitoring sepeda motor. Implementasi ini melibatkan penggunaan nodejs untuk menampilkan lokasi koordinat kendaraan, memungkinkan pemilik kendaraan melacak posisi kendaraan saat terjadi pencurian. Hasil analisis pengujian menunjukkan bahwa perangkat keras terkoneksi dengan WiFi dan broker MQTT berhasil, namun waktu akuisisi sinyal GPS bervariasi antara 9.07 detik, 4.73 detik, dan 11.03 detik. Protokol MQTT menggunakan broker EMQX pada VPS berhasil mengirimkan data dari modul GPS Neo6M ke web client dengan Express.js untuk menampilkan koordinat kendaraan. Pengujian tiga skenario durasi dan jarak menyoroti perbedaan kinerja antara MQTT dan HTTP: Latency rata-rata MQTT (1.58 ms - 1.645 ms) dan HTTP (0.455 ms - 0.46 ms), serta throughput rata-rata MQTT (621.48 bps - 676.385 bps) dan HTTP (2378.355 bps - 2409.175 bps). Meskipun HTTP memiliki latency yang lebih rendah, MQTT secara konsisten menunjukkan throughput yang lebih tinggi. MQTT mampu beroperasi dengan bandwidth yang lebih kecil namun tetap memberikan kinerja throughput yang unggul, meskipun dengan sedikit peningkatan latency.

**Keywords**— IoT Security, MQTT Protocol, Vehicle Security, Throughput and Latency Comparison, GPS Tracking, Node.js and Express.js Implementation, Wireless Connectivity (WiFi), dan Neo6M GPS Module.

## 1. Introduction

diera Kendaraan telah menjadi keharusan dalam kehidupan sehari-hari yang penuh aktivitas. Kendaraan memainkan peran penting dalam memfasilitasi mobilitas manusia, baik untuk perjalanan jauh maupun dekat. Kendaraan memungkinkan kita menjalani kehidupan dengan lebih efisien dan fleksibel. Yang memungkinkan kita untuk menghubungkan tempat-tempat penting tanpa terhalang oleh batasan geografis [1]. Tantangan yang perlu diatasi, bagaimanapun, terkait dengan kelanjutan penggunaan kendaraan. Salah satunya adalah masalah kejahatan seperti pencurian dan begal kendaraan bermotor menjadi masalah yang sangat menghantui masyarakat. Tingkat kejahatan yang tinggi ini tidak hanya mengancam harta benda, tetapi juga mengganggu rasa aman masyarakat [2]. Oleh karena itu, fitur keamanan pada sepeda motor menjadi alat yang dapat diandalkan untuk mencegah perilaku kriminal seperti begal atau perampasan kendaraan bermotor. Dalam sektor IoT,

contohnya, teknologi keamanan membuat pengguna kendaraan bermotor merasa aman saat terjadi keadaan darurat, dengan hanya menekan tombol darurat dan motor akan mati dalam waktu 10 detik[3]. Selain itu, salah satu solusi untuk masalah ini adalah menggunakan protokol HTTPS secara berkala untuk melacak posisi kendaraan dengan GPS. Data koordinat lokasi dapat dikirim ke web server dan ditampilkan dalam bentuk peta menggunakan Google Maps [4]. Berbicara tentang Internet of Things, atau Internet of Things, tidak bisa lepas dari kata "internet" protokol yang memungkinkan perangkat berkomunikasi satu sama lain. Beberapa protokol yang paling umum untuk komunikasi dan pemantauan Internet of Things adalah NFC, Bluetooth Low Energy, MQTT, HTTP, dan HTTPS[5]. Penelitian terkait IoT dan penerapan protokol MQTT serta HTTP dalam berbagai aplikasi telah dilakukan oleh beberapa peneliti, mulai dari aspek keamanan pada sepeda motor dengan SIM GSM, perbandingan performa latency antara protokol

4  
komunikasi HTTP dan MQTT pada Internet of Things, hingga implementasi MQTT dalam aplikasi perkebunan pintar dan greenhouse. Meskipun penelitian ini menunjukkan kelebihan protokol MQTT, seperti keunggulan dalam keamanan dan performa latency, setiap studi juga mencerminkan keterbatasan, seperti biaya tambahan dalam penggunaan GSM/SMS, keterbatasan uji coba yang tidak memperhitungkan jarak antara server MQTT dan perangkat IoT, serta kebutuhan untuk lebih mengevaluasi efisiensi latency dan bandwidth saat menggunakan MQTT. Secara keseluruhan, penelitian-penelitian ini menyuguhkan wawasan yang mendalam mengenai potensi dan tantangan dalam mengadopsi protokol MQTT dalam konteks aplikasi Internet of Things. Berdasarkan latar belakang diatas, penelitian ini bertujuan untuk memanfaatkan protokol MQTT dalam membangun sistem monitoring sepeda motor. Penelitian ini juga menerapkan nodejs sebagai Bahasa pemrograman pengembangan sistem monitoring sepeda motor sebagai media untk menampilkan lokasi koordinat kendaraan sehingga ketika terjadi pencurian bisa mengetahui lokasi kendaraan yang dicuri . Dengan demikian diharapkan penelitian ini dapat membantu pemilik kendaraan mengamankan dari tindakan pencurian

2. Metode

Alur penelitian yang dirancang dari tahap awal hingga selesai. Setiap titik dalam alur tersebut menjelaskan proses perancangan sistem dan alat yang bertujuan membentuk prototipe. Penjelasan pada setiap tahap dalam alur penelitian menggunakan metode SDLC Prototipe (System Development Life Cycle).

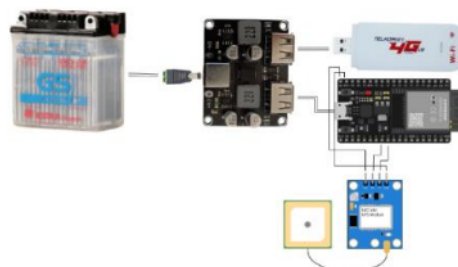


Gambar 2 alur penelitian

2.1. Perumusan metode studi literatur secara teoritis

Dalam tahap awal penelitian, fokusnya adalah menganalisis protokol komunikasi umum dalam Internet of Things (IoT), seperti MQTT, CoAP, dan HTTP. melibatkan pengumpulan informasi mendalam tentang berbagai protokol tersebut, menilai keunggulan, kelemahan, serta aplikasi khusus dalam konteks sistem pemantauan sepeda motor [6]. Kemudian, dalam studi literatur mengenai pemilihan modul dan sensor, diputuskan untuk menggunakan modul GPS-Neo6M

yang akan terkoneksi dengan ESP32. Modul ini akan mengirimkan data koordinat longitude dan latitude dalam format JSON. Dengan kombinasi ini, diharapkan mendapatkan informasi lokasi yang akurat dari sepeda motor yang sedang dimonitor. Langkah berikutnya adalah pengembangan antarmuka web menggunakan Node.js untuk responsifitas pembuatan halaman website, MySQL sebagai basis data penyimpanan lokasi dari GPS, dan library Leaflet maps untuk visualisasi data lokasi. Harapannya, antarmuka ini memberikan pengalaman yang intuitif bagi pengguna.



Gambar 1 rangkaian perangkat keras

2.2. Node Desain dan Perancangan Perangkat keras

Beberapa komponen yang diperlukan untuk penelitian ini meliputi ESP32, Modul GPS-Neo6M, Modul USB Fast Charger 5V 3.4A QC3.0 24W, dan Modem Stik. Pada Gambar 2, ESP32 berperan sebagai Mikrokontroler yang akan terkoneksi dengan Modul GPS Neo-6M untuk menangkap titik koordinat lokasi. Mikrokontroler ESP32 akan bertindak sebagai pengirim data ke server menggunakan koneksi WiFi yang diakses melalui modem stik sebagai sumber internet. Kedua perangkat, ESP32 dan modem stik, akan ditenagai oleh Modul USB Fast Charger 5V 3.4A QC3.0 24W Charging. Modul ini akan memperoleh daya listrik dari aki motor. Pada table 1 adalah rangkain kabel jumper dari esp32 ke modul GPS-Neo6M

2.3. Desain Perancangan Sistem

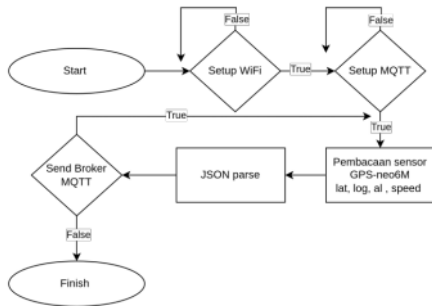
Langkah-langkah dalam pengembangan sistem dimulai dengan pemahaman mendalam akan kebutuhan pengguna. melibatkan analisis yang cermat untuk menentukan fungsionalitas esensial tanpa memasukkan elemen yang tidak relevan. Proses ini kemudian dimodelkan dalam diagram atau catatan langkah demi langkah yang jelas, menggambarkan proses dari input sampai output. Dalam tahap ini, algoritma sistem dirancang dengan fokus pada proses inti, mengeliminasi yang tidak relevan, dan menyederhanakan proses secara keseluruhan.

2.4. Inisialisai Program ESP32

Program pada ESP32 bertanggung jawab untuk mengatur koneksi WiFi, mengelola klien MQTT, dan menginisialisasi komunikasi serial untuk modul GPS. Dalam prosesnya, program terus memantau ketersediaan koneksi WiFi, koneksi broker MQTT, dan data sinyal GPS. Program akan mengambil informasi lintang, bujur, ketinggian, dan kecepatan jika data tersedia, lalu mengirimnya ke broker MQTT dalam format JSON. Selain itu, program juga terus memantau koneksi ke broker MQTT mencoba untuk terhubung kembali jika terputus, sambil memberi informasi tentang status koneksi. Semua proses ini berlangsung dalam loop utama program hingga program selesai atau dihentikan.

2.5. Pembuatan Prototype Web

Data output dari MQTT broker akan disubscribe oleh web menggunakan library dari Node.js, yaitu MQTT.js, dan Leaflet.js sebagai maps. Desain web yang interaktif akan meningkatkan pengalaman pengguna secara lebih mendalam. Antarmuka yang dirancang merupakan representasi dari hasil pengumpulan data yang akan diimplementasikan ke dalam suatu sistem. Desain antarmuka diperlukan agar mempermudah interaksi antara pengguna dan sistem.



Gambar 3 alur program pada ESP32

2.6. Integrasi Perangkat Keras dengan Sistem Monitoring

Integrasi perangkat keras dengan sistem monitoring melibatkan tiga komponen utama. Pertama, NodeMCU



Gambar 4 alur Integrasi Perangkat Keras dengan Sistem Monitoring

ESP32 dan GPS Neo-6M berperan dalam pengambilan data dari modul GPS dan koneksi ke internet. Data seperti longitude, latitude, dan kecepatan akan dikirim ke Broker MQTT dengan pesan terjadwal setiap 5 detik dalam format JSON. Kedua, Broker MQTT menggunakan EMQX di VPS untuk menerima dan menyimpan data tersebut. Terakhir, klien web yang telah direncanakan akan berlangganan (subscribe) ke topik GPS/malih/data di

MQTT Broker untuk menampilkan informasi tersebut dalam antarmuka web.

3. Result Hasil dan Pengujian

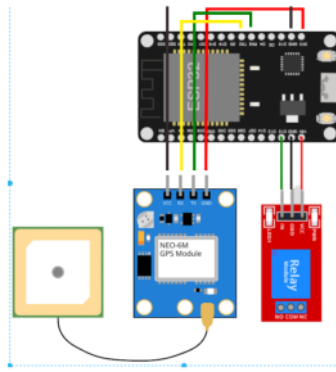
Bab ini merangkum hasil dari serangkaian eksperimen dan pengujian yang telah dilakukan dalam konteks implementasi dan evaluasi sistem yang telah dirancang. Hasil-hasil yang diuraikan dalam bab ini mencakup:

1. Pengujian Perangkat Keras

Proses pertama melibatkan unggah sketch pada ESP32 melalui Arduino IDE. Rangkaian alat yang telah dirancang. Tahap pertama melibatkan kompilasi sketch dan menunggu hingga modul GPS mendapatkan sinyal yang terbukti dengan lampu indikator pada modul yang Akan berkedip. Selanjutnya data akan ditampilkan



Gambar 5 pengujian serial monitor

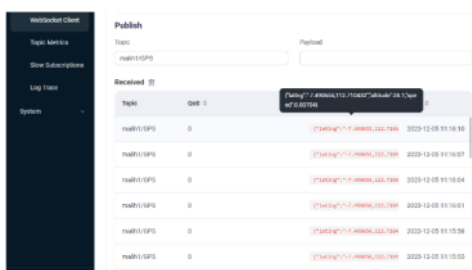


Gambar 6 pengujian serial monitor



2 Implementasi Protokol MQTT

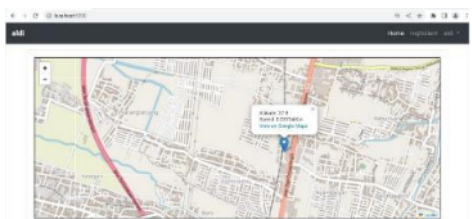
Langkah pertama adalah menginstal broker pada VPS. Langkah ini dimulai dengan mengunduh installer dari broker.emqx.io, kemudian menjalankan proses instalasi. Jika instalasi berhasil, akseslah web dashboard yang telah disediakan melalui web browser menggunakan alamat `http://alamatip:18083`. Setelah itu, periksa bagian klien pada dashboard dan subscribe topik yang sesuai untuk memastikan bahwa data dari modul perangkat keras, telah berhasil mengirimkan data berupa longitude, latitude, dan kecepatan



Gambar 7 hasil dari dashbord EMQX

3. Pengujian Web Dashboard

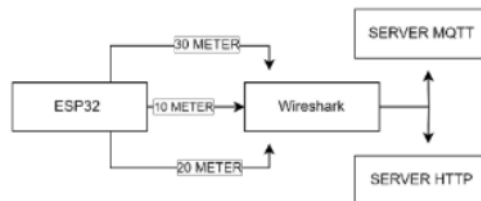
Pengujian ini dilakukan untuk melihat apakah data sensor yang dikirim oleh ESP32 pada server MQTT bisa ditampilkan dan data modul GPS-Neo6M bisa ditampilkan pada web yang telah dirancang



Gambar 8 hasil dari web Dashboard

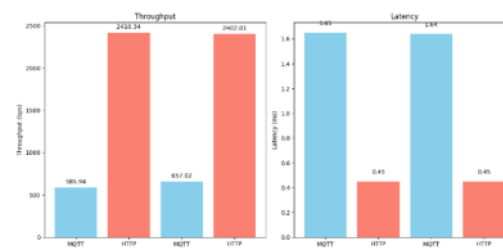
4. Pengujian Perbandingan Protokol MQTT dan HTTP

Mekanisme pengujian menggunakan MQTT dan HTTP sebagai protokol pengiriman data. Wireshark akan mengumpulkan informasi tentang throughput (bps), jitter (ms), dan latency (ms) dari data yang dikirim. Pengujian dilakukan dari jarak 20m, 50m, dan 70m dari WiFi ke ESP32. Ini memungkinkan evaluasi menyeluruh kinerja kedua protokol pada jarak yang berbeda, yang berdampak pada konektivitas skenario pengujian.



Gambar 9 skenario pengujian HTTP dan MQTT

- a. Pengujian throughput(bps),Jitter(ms),Latency(ms) selama 10 menit 20 meter

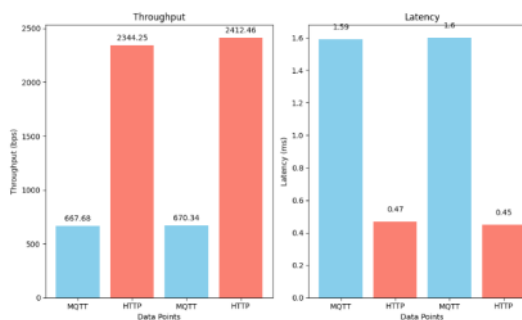


Gambar 10 hasil pengujian HTTP dan MQTT

No	Scenario	Protokol	Throughput (bps)	jitter (ms)	Latency (ms)
1	10d-20M	MQTT	585.94	1658.03	1.65
		HTTP	2416.34	448.12	0.45
		MQTT	657.02	1642.96	1.64
		HTTP	2402.01	451.93	0.45

Table 1 hasil pengujian HTTP dan MQTT

- b. Pengujian throughput(bps),Jitter(ms),Latency(ms) selama 10 menit 20 meter

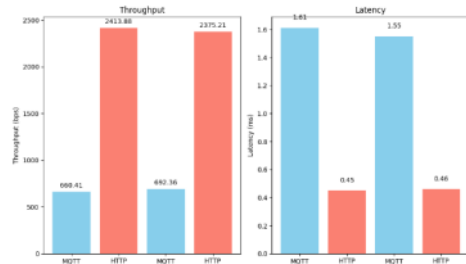


Gambar 11 hasil pengujian HTTP dan MQTT

No	Scenario	Protokol	Throughput (bps)	jitter (ms)	Latency (ms)
1	10d-20M	MQTT	585.94	1658.03	1.65
		HTTP	2416.34	448.12	0.45
		MQTT	657.02	1642.96	1.64
		HTTP	2402.01	451.93	0.45

Table 1 hasil pengujian HTTP dan MQTT

- c. Pengujian throughput(bps),Jitter(ms),Latency(ms) selama 10 menit 20 meter



Gambar 12 hasil pengujian HTTP dan MQTT

No	Scenario	Protokol	Throughput (bps)	jitter (ms)	Latency (ms)
1	10d-20M	MQTT	585.94	1658.03	1.65
		HTTP	2416.34	448.12	0.45
		MQTT	657.02	1642.96	1.64
		HTTP	2402.01	451.93	0.45

Table 1 hasil pengujian HTTP dan MQTT

5. Analisa hasil Pengujian

Dari paparan dan penjelasan mendapatkan hasil yang terbaik pada implemtasi protokol MQTT pada rancang bangun monitroing sepeda motor ada bebrapa analisa diantaranya sebagai berikut :

- Perangkat keras berhasil berfungsi dengan baik mampu terkoneksi dengan WiFi dan terhubung ke broker MQTT. Dalam pengujian akuisisi sinyal GPS dengan tiga kali percobaan, waktu yang dihasilkan adalah 9.07 detik, 4.73 detik, dan 11.03 detik. Terdapat beberapa perbedaan dalam proses akuisisi sinyal yang bisa dipengaruhi oleh faktor-faktor seperti lokasi berada di dalam ruangan, kondisi cuaca, dan lain sebagainya. Semua hasil pengujian dimulai dari koneksi ke WiFi, koneksi ke broker MQTT, dan pengiriman data koordinat lokasi ke broker. Informasi ini dapat dilihat melalui serial monitor pada Arduino IDE
- Protokol MQTT menggunakan broker EMQX yang diinstal pada VPS berhasil berfungsi dengan baik untuk menerima dan mengirimkan data dari modul GPS Neo6M kepada web client
- Web client yang menggunakan Express.js berhasil berfungsi dengan baik, mampu menampilkan titik koordinat yang diterima dari broker MQTT
- Pada hasil pengujian tiga skenario yang berbeda, dengan durasi dan jarak yang beragam, menyoroti perbedaan kinerja antara protokol MQTT dan HTTP. Dalam pengukuran throughput (bps) dan latency (ms), protokol HTTP menunjukkan

performa latency yang lebih baik dibandingkan MQTT, dengan nilai latency rata-rata 0.455 ms - 0.46 ms dibandingkan dengan 1.58 ms - 1.645 ms pada MQTT. Namun, nilai throughput rata-rata pada MQTT secara konsisten lebih tinggi daripada HTTP, dengan nilai rata-rata throughput MQTT antara 621.48 bps - 676.385 bps dan HTTP antara 2378.355 bps - 2409.175 bps. Dari hasil ini, terlihat bahwa MQTT mampu beroperasi dengan menggunakan bandwidth jaringan yang lebih kecil dibandingkan HTTP, sementara tetap memberikan kinerja throughput yang unggul, meskipun dengan nilai latency yang sedikit lebih tinggi

4. Conclusion

Analisis dari hasil pengujian menunjukkan bahwa protokol MQTT sangat cocok untuk penggunaan dalam rancang bangun monitoring sepeda motor ini. Meskipun menunjukkan throughput yang lebih rendah dibandingkan HTTP, MQTT memberikan keunggulan dalam penggunaan bandwidth yang lebih efisien, menghasilkan penghematan dalam paket data yang dikirimkan melalui jaringan. Meskipun latency-nya sedikit lebih tinggi, kelebihan dalam efisiensi penggunaan bandwidth dapat menjadi faktor penting dalam mempertimbangkan penggunaan protokol ini untuk aplikasi monitoring sepeda motor

## References

- [1] Patel and R. Goyena, "UPAYA KEPOLISIAN DALAM MENANGGULANGI TINDAK PIDANA PEMBEGALAN DI WILAYAH KOTA PASURUAN," *J. Chem. Inf. Model.*, vol. 15, no. 2, pp. 9–25, 2019.
- [2] M. Arifin, D. P. D. selengkapnya <https://www.detik.com/jatim/berita/-6030059/3-begal-sadis-di-pasuruan-diringkus-dua-pelaku-ditembak>. Baca artikel detikjatim, "3 Begal Sadis di Pasuruan Diringkus, and D. A. D. S. <https://apps.detik.com/detik/>, "3 Begal Sadis di Pasuruan Diringkus, Dua Pelaku Ditembak." <https://www.detik.com/jatim/berita/d-6030059/3-begal-sadis-di-pasuruan-diringkus-dua-pelaku-ditembak> (accessed Aug. 28, 2023).
- [3] S. Samsugi and W. Wajiran, "IoT: EMERGENCY BUTTON SEBAGAI PENGAMAN UNTUK MENGHINDARI PERAMPASAN SEPEDA MOTOR," *J. Teknoinfo*, vol. 14, no. 2, p. 99, 2020, doi: 10.33365/jti.v14i2.653.
- [4] Y. Pratama, D. N. Ramadan, S. Pd, and T. N. Damayanti, "Perancangan GPS Tracking Untuk Penyewaan Kendaraan Bermotor Design of GPS Tracking on Lending Motor Vehicle," *e-Proceeding Appl. Sci.*, vol. 6, no. 2, pp. 1–15, 2020.
- [5] S. B. Bhaskoro, H. Supriyanto, B. B. Aji, and B. Pamungkas, "Perbandingan Performansi Latency Protokol Komunikasi Http Dan Mqtt Pada Internet of Things," *JIT (Jurnal Teknol. Ter.*, vol. 8, no. 2, p. 82, 2022, doi: 10.31884/jit.v8i2.309.
- [6] S. Beta, R. Reynaldi, and G. F. Avisyiah, "PROTOTYPE AIR CONDITIONING MONITORING AND CONTROL SYSTEM FOR SMART CLASSROOM BASED ON THE INTERNET OF THINGS," vol. 8, no. 1, pp. 207–216, 2023.

# IMPLEMENTASI PROTOKOL MQTT PADA GPS-NEO6M SEBAGAI RANCANG BANGUN MONITORING SEPEDA MOTOR

## ORIGINALITY REPORT

7%

SIMILARITY INDEX

6%

INTERNET SOURCES

3%

PUBLICATIONS

3%

STUDENT PAPERS

## PRIMARY SOURCES

1	<a href="http://jurnal.polines.ac.id">jurnal.polines.ac.id</a> Internet Source	2%
2	<a href="http://ejournal.itn.ac.id">ejournal.itn.ac.id</a> Internet Source	1%
3	<a href="http://www.bukalapak.com">www.bukalapak.com</a> Internet Source	1%
4	<a href="http://jurnal.polindra.ac.id">jurnal.polindra.ac.id</a> Internet Source	1%
5	Diana Cindy Agustin, Mochamad Alfian Rosid, Novia Ariyanti. "IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK UNTUK DETEKSI KESEGERAN PADA APEL", JURNAL FASILKOM, 2023 Publication	<1%
6	<a href="http://news.unair.ac.id">news.unair.ac.id</a> Internet Source	<1%
7	<a href="http://id.123dok.com">id.123dok.com</a> Internet Source	<1%



---

Exclude quotes      On

Exclude matches      Off

Exclude bibliography      On