

PERBANDINGAN PENGGUNAAN MIKROKONTROLER ESP8266 DAN ESP32 SECARA *REAL-TIME* UNTUK SISTEM MONITORING RUANG SERVER BERBASIS IOT

(STUDI KASUS: DINKOMINFO KAB. PURBALINGGA)

Rafli Maulana Fahrurrahman
Teknik Informatika
Telkom University Purwokerto
Purwokerto, Indonesia
raflimaulanaf@student.telkomuniversity.ac.id

Aulia Desy Nur Utomo, S.kom., M.Cs.
Teknik Informatika
Telkom University Purwokerto
Purwokerto, Indonesia
auliau@telkomuniversity.ac.id

Abstrak — Ruang *server* memiliki peranan yang sangat penting pada instansi ataupun lembaga yang menggunakan teknologi informasi dan komunikasi sebagai kegiatan sehari-hari, *server* tidak mungkin di nonaktifkan karena *server* harus selalu menyala untuk menunjang kebutuhan data. Akan tetapi *server* yang menyala secara terus menerus mungkin dapat menimbulkan terjadinya masalah seperti kenaikan suhu, kelembaban, dan masalah lainnya, serta jarak ruang *server* yang jauh dapat menjadi salah satu hambatan. Maka dari itu, dibutuhkan alat yang dapat memonitoring kondisi pada ruang *server*. Penelitian ini bertujuan membangun alat yang dapat memonitoring kondisi pada ruang *server* secara otomatis serta dapat diakses dimanapun dan kapanpun sehingga pengguna tidak selalu harus selalu berada di ruang *server* untuk melakukan pemantauan. Jenis metode yang digunakan pada penelitian ini yaitu *prototyping* dengan metode pengujian *blackbox*. Hasil dari pengujian yang telah dilakukan menunjukan bahwa kedua mikrokontroler berfungsi dengan baik dalam memantau kondisi ruang *server*. Namun, Esp8266 menunjukan performa yang lebih baik dalam hal *throughput* dan *delay data*, sehingga lebih efektif untuk aplikasi monitoring *real-time* dengan nilai *throughput esp8266* 3,166kbps dan nilai *delay Esp8266* yaitu 91ms serta nilai *throughput Esp32* 2,239kbps dan *delay* 148ms.

Kata kunci— Esp32, *Internet of Things*, Monitoring, NodeMCU Esp8266, *Server*

I. PENDAHULUAN

Ruang *server* merupakan ruangan yang memiliki fungsi sebagai tempat menyimpan *server*, perangkat jaringan, dan perangkat lainnya yang memiliki keterkaitan dengan sistem operasional sehari-hari. Kondisi *server* dapat dipengaruhi

oleh banyak faktor seperti suhu ruangan dan kelembaban harus berada dalam batas normal pada ruang *server*.

Penting untuk mengetahui apa yang terjadi di dalam ruang *server*. Masalah dapat terjadi ketika *server* bekerja keras atau tidak diawasi. Jika ada masalah dengan peralatan *server*, hal itu dapat menyebabkan hilangnya data dan bahkan mengakibatkan hilangnya uang. Selain itu, banyak administrator masih harus memeriksa ruang *server* secara berkala untuk memastikan suhunya sesuai agar *server* dapat berfungsi dengan baik. Masalah ini terjadi karena ruang *server* cukup jauh dan harus selalu dikunci demi alasan keamanan.

Berdasarkan masalah tersebut, maka dilakukan penelitian dengan judul “Perbandingan penggunaan mikrokontroler *esp8266* dan *esp32* secara real time untuk sistem monitoring ruang *server* berbasis iot (studi kasus: dinkominfo Kab. purbalingga)” dimana sistem yang dibuat dapat mengukur suhu, kelembaban, asap, percikan api, memonitoring dari jarak yang jauh dengan menggunakan platform *android* dan *firebase* sebagai *database* untuk menyimpan data dari sensor. Pada alat ini kita membutuhkan koneksi jaringan internet yang selalu aktif pada *smart phone* agar sistem dapat termonitoring dengan baik.

Berdasar pada latar belakang yang sudah diuraikan, maka rumusan masalah pada proyek akhir ini yaitu:

- a) Admin *server* belum bisa memonitoring secara berkala pada ruangan *server* yang harus dijaga setiap waktu, dikarenakan lokasi *server* yang berada terpisah sehingga admin terkendala dalam melakukan pengecekan ruangan *server* tersebut.

Mengingat luasnya penelitian ini, maka dilakukan batasan masalah yaitu:

- a) Menggunakan Nodemcu *esp8266* dan *esp32* sebagai mikrokontroler.
- b) Menggunakan platform *android* sebagai media monitoring sistem keamanan pada ruang *server*.
- c) Hanya melakukan pengujian *Qos* dengan parameter *throughput* dan *delay* pada *Esp32* dan *Esp8266*.

- d) Hanya menggunakan *database firebase* untuk menyimpan data yang akan dikirimkan ke platform *android*.
- e) Hanya melakukan monitoring pada ruangan *server* menggunakan sensor *DHT11*, *Acs712*, *mq-2* dan sensor flame api.

Tujuan penulis untuk melakukan penelitian ini yaitu:

- a) Tujuan dari tugas akhir ini adalah merancang sistem monitoring pada ruang *server* dimana dapat memonitoring secara jarak jauh menggunakan platform *android* dan juga membantu memenuhi standar keamanan pada ruang *server* apabila terjadi insiden yang terjadi pada ruang *server* dan melakukan perbandingan *QoS* dengan parameter *throughput* dan *delay* pada mikrokontroler *Esp32* dan *Esp8266* untuk mengetahui

II. KAJIAN TEORI

Internet of Things (IoT) adalah salah satu tren terbaru dalam perkembangan teknologi yang diprediksi akan menjadi kecenderungan di masa depan. Teknologi ini menawarkan sebuah konsep yang dapat meningkatkan penggunaan jaringan internet. Sederhananya, IoT digunakan untuk menghubungkan objek fisik di dunia nyata secara virtual melalui jaringan. Selain itu, data juga dapat diperoleh dari sensor yang dimanfaatkan teknologi tersebut. Inti dari IoT adalah akses internet yang luas karena perangkat harus selalu terkoneksi dengan jaringan agar dapat selalu diakses kapanpun dan dimanapun.

A. Tinjauan Alat



GAMBAR 1
(NODEMCU ESP32)

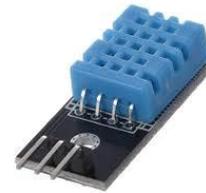
ESP32 adalah jenis mikrokontroler buatan Espressif Systems. Mikrokontroler ini memiliki jaringan Wi-Fi bawaan. ESP32 menggunakan prosesor dual-core berbasis Xtensa LX16. Secara umum, ESP32 merupakan mikrokontroler serbaguna yang dapat digunakan dalam berbagai proyek elektronik, seperti Internet of Things (IoT), robotika, sistem otomatis, dan lainnya. Fitur nirkabelnya menjadikannya ideal untuk proyek yang membutuhkan koneksi internet atau untuk berkomunikasi secara nirkabel dengan perangkat lain[1].



GAMBAR 2
(NODEMCU ESP8266)

NodeMCU adalah platform IoT sumber terbuka. Platform ini menggunakan chip ESP 8266 dari seri ESP

buatan Espressif Systems, dan firmware yang digunakan ditulis dalam bahasa skrip Lua. Istilah NodeMCU biasanya merujuk pada firmware, bukan perangkat keras fisik. NodeMCU juga dapat dianggap sebagai papan Arduino berbasis ESP 8266. Selain Lua, NodeMCU juga dapat digunakan dengan Arduino IDE[2].



GAMBAR 3
(Sensor DHT11)

DHT-11 adalah sensor yang mengukur suhu dan kelembapan. Sensor ini bekerja dengan andal dalam jangka waktu lama dan tetap stabil. Sensor ini mengirimkan informasi suhu sebagai sinyal digital ke pin data, sehingga Anda tidak memerlukan input sinyal lain[3].



GAMBAR 4
(Sensor MQ-2)

Secara umum Sensor bisa didefinisikan menjadi alat yang bisa mendeteksi sebuah benda atau fenomena atau kimia lalu mengubahnya menjadi frekuensi elektrik ataupun menjadi sebuah tegangan dan bisa membuat sinyal elektrik seperti tekanan, gaya, pergerakan, temperature dan sebagainya. Gas sensor digunakan untuk mendeteksi berbagai jenis gas atau asap di sekitarnya. Layaknya hidung manusia, sensor ini dapat membedakan antara gas normal dan gas berbahaya[4].



GAMBAR 5
(Sensor Flame Api)

Sensor api ini dapat mendeteksi api dengan mendeteksi gelombang cahaya antara 760 nm dan 1100 nm. Sensor ini memiliki sudut pandang 60 derajat dan bekerja dalam rentang suhu 25 hingga 85 derajat Celsius. Perlu diperhatikan juga bahwa jarak antara sensor dan objek yang dideteksi tidak boleh terlalu dekat untuk mencegah kerusakan sensor[5].



GAMBAR 6
(Sensor ACS712)

Sensor arus ACS712 memiliki akurasi tinggi dan dapat digunakan untuk mengukur arus AC maupun DC. Prinsip kerja sensor ini adalah ketika arus mengalir melalui konduktor, medan magnet akan tercipta, yang dideteksi oleh IC efek Hall yang terintegrasi di dalam perangkat. Keluaran sensor menghasilkan tegangan linier yang berubah seiring dengan perubahan arus yang terdeteksi. Nilai keluaran yang dapat dideteksi oleh sensor kemudian masuk sebagai masukan ke mikrokontroler[6].

III. METODE

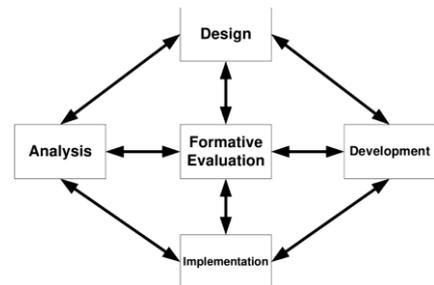
Teknis pengujian akan dilakukan dalam satu waktu dimana pengujian tersebut akan melakukan pengambilan data terhadap sensor yang digunakan seperti suhu, kelembaban, percikan api, asap, tegangan arus listrik. Untuk mengujian sensor flame api akan digunakan korek api sebagai bahan pengambilan data, dan sensor MQ-2 akan menggunakan kertas yang dibakar untuk pengambilan data, serta akan dilakukan perbandingan *Qos* dengan parameter *throughput* dan *delay* pada *esp2866* dan *esp32* untuk mencari tahu alat mana yang lebih efektif dalam melakukan pengiriman datanya. Pengujian dilakukan sebanyak 5 kali pada setiap mikrokontroler untuk melakukan pengambilan data dan menggunakan jaringan yang sama untuk mengetahui selisih nilai *throughput* dan *delay* dari kedua mikrokontroler, kemudian data tersebut akan dimasukkan kedalam tabel yang sudah disertakan.



GAMBAR 7

(Diagram Alir Penelitian)

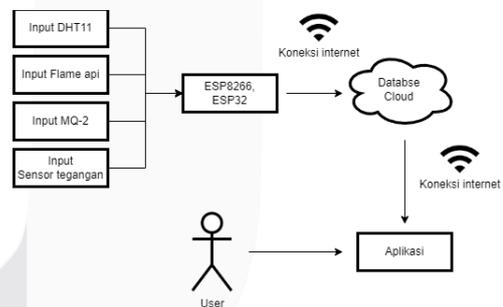
Metodologi Penelitian ialah langkah-langkah yang memiliki tujuan untuk menyelesaikan permasalahan sehingga permasalahan tersebut dapat diselesaikan dengan terstruktur. Berikut adalah *flowchart* penelitian sebagai dasar penelitian sehingga dapat terlaksanakan dengan terarah.



GAMBAR 8

(Alur metode *rapid prototype*)

Pada tahap ini yaitu menentukan metode penelitian yang akan digunakan dalam pembuatan sistem. Pada penelitian ini menggunakan metode *rapid prototype*, dikarenakan pada metode ini sesuai dalam melakukan perancangan aplikasi sederhana dan perancangan alat. Metode *rapid prototype* digambarkan saling terhubung dan dimulai dari tahap '*analysis*', pada tahap ini dilakukan penentuan komponen apa saja yang dibutuhkan untuk membangun sebuah sistem. Selanjutnya tahap '*design*', pada tahap kedua ini dilakukan perancangan sistem *input*, *output*, serta proses yang digunakan dan dihasilkan pada sistem. Kemudian tahap '*development*' yaitu tahap yang dilakukan ketika sistem yang sudah terdesain tersebut sudah sesuai dengan apa yang diinginkan sehingga proses dapat dilakukan. Tahap selanjutnya yaitu '*implementation*' tahap ini dilakukan ketika sistem sudah berjalan dengan baik. Tahap terakhir yaitu '*formative evaluation*' pada tahap ini bertujuan menentukan apa yang harus ditingkatkan dari sistem yang telah dibangun agar sistem lebih efektif.



GAMBAR 9

(BLOK DIAGRAM SISTEM)

Diagram Blok diatas terlihat bahwa hasil dari semua *input* sensor akan masuk ke *esp8266* dan *esp32* yang kemudian data tersebut akan dikirimkan ke cloud setelah perangkat terkoneksi ke jaringan internet, dan ketika *user* ingin mengakses aplikasi harus terkoneksi ke internet agar dapat terhubung ke *database* untuk mengambil data dari sensor.

IV. HASIL DAN PEMBAHASAN

Hasil ini merupakan kelanjutan dari perancangan yang telah dibuat, yaitu hasil dari pengujian serta pembahasan, maka dari itu dilakukan pengujian dan evaluasi pada alat yang telah dirancang. Bab ini akan akan dilakukan pembahasan hasil pengujian dan perancangan dari perbandingan penggunaan mikrokontroler *esp2866* dan *esp32* secara *real-time* untuk sistem monitoring ruang server berbasis iot.

A. Perancangan Sistem

Rancangan *interface* pada aplikasi sistem monitoring dapat dilihat pada gambar dibawah ini.



GAMBAR 10

(HASIL RANCANGAN APLIKASI)

Merupakan tampilan awal pada aplikasi sistem monitoring *esp32* dan *esp8266* dimana pada tampilan tersebut semua sensor belum mengirmkan data ke *database* sehingga sehingga tampilan sensor pada aplikasi monitoring masih dalam keadaan *OFF*.

B. Hasil Pengujian Sensor MQ-2, Flame Api Esp32 dan Esp8266

TABEL 1

(PENGUJIAN SENSOR MQ-2 DAN FLAME API ESP32)

NO	Jarak (m)	Mq2	Flame api	Mengirim data ke database
1	0,2	On	On	Terkirim
2	0,4	On	On	Terkirim
3	0,6	On	On	Terkirim
4	0,8	On	On	Terkirim
5	1	Off	Off	Tidak Terkirim

Tabel ini menunjukkan hasil pengujian sensor gas *MQ2* dan sensor api yang terhubung ke *ESP32*. Pengujian dilakukan pada lima jarak berbeda, dari 0,2 m hingga 1 m. Pada jarak 0,2 m hingga 0,8 m, sensor dapat mendeteksi gas dan api, dan data berhasil dikirim ke basis data. Namun, pada jarak 1,0 m, sensor tidak mendeteksi apa pun dan data gagal dikirim. Dilakukan perhitungan pengujian fungsionalitas diperoleh:

$$\text{Keberhasilan} = \frac{4}{5} \times 100\% = 80\%$$

Dari perhitungan yang telah dilakukan dari lima pengujian, empat berhasil, sehingga tingkat keberhasilannya adalah 80%.

TABEL 2

(PENGUJIAN SENSOR MQ-2 DAN FLAME API ESP8266)

NO	Jarak (m)	Mq2	Flame api	Mengirim data ke database
1	0,2	On	On	Terkirim
2	0,4	On	On	Terkirim
3	0,6	On	On	Terkirim
4	0,8	On	On	Terkirim

5	1	Off	Off	Tidak Terkirim
---	---	-----	-----	----------------

Tabel ini menunjukkan hasil pengujian sensor gas *MQ2* dan sensor api yang terhubung ke *ESP8266*. Pengujian dilakukan pada lima jarak yang berbeda, dari 0,2 hingga 1,0 meter. Pada jarak 0,2 hingga 0,8 meter, sensor berhasil mendeteksi gas dan api, dan data terkirim dengan baik ke database. Namun, ketika jarak mencapai 1,0 meter, sensor tidak mendeteksi apa pun, dan data tidak terkirim. Hal ini dapat terjadi karena adanya gangguan pada pengiriman data ketika sensor tidak mendeteksi apa pun. Dilakukan perhitungan pengujian fungsionalitas berdasarkan Tabel 4.1 diperoleh:

$$\% \text{Keberhasilan} = \frac{4}{5} \times 100\% = 80\%$$

Dari perhitungan yang telah dilakukan dari lima percobaan, empat kali berhasil, sehingga tingkat keberhasilannya adalah 80%.

C. Hasil pengujian sensor DHT11 Esp32 dan Esp8266

TABEL 3

(PENGUJIAN SENSOR DHT11 ESP32)

Waktu	Suhu (°C)	Kelembaban (%)	Status Suhu	Status Kelembaban
10:00 WIB	24.1	48.3	Normal	Normal
10:01 WIB	24.3	49.0	Normal	Normal
10:02 WIB	24.6	50.2	Normal	Normal
10:03 WIB	24.9	51.0	Normal	Normal
10:04 WIB	25.1	52.5	Normal	Normal

Tabel ini memuat hasil pengukuran suhu dan kelembaban di ruang server yang dilakukan dengan sensor *DHT11* yang terhubung ke mikrokontroler *ESP32*. Data diambil setiap menit sekali dan disajikan dalam format angka desimal (*float*) untuk merepresentasikan perubahan kecil pada kondisi lingkungan berpendingin. Nilai suhu dan kelembaban tercatat masih dalam rentang ideal, yaitu suhu antara 18 hingga 27°C dan kelembaban antara 40 hingga 60% RH. Hal ini menandakan bahwa sistem pendinginan dan pengaturan kelembaban di ruang *server* berjalan dengan baik.

TABEL 4

(PENGUJIAN SENSOR DHT11 ESP8266)

Waktu	Suhu (°C)	Kelembaban (%)	Status Suhu	Status Kelembaban
10:00 WIB	25.3	49.5	Normal	Normal
10:01 WIB	25.5	51.2	Normal	Normal
10:02 WIB	25.7	53.0	Normal	Normal
10:03 WIB	26.0	55.1	Normal	Normal
10:04 WIB	26.3	60.2	Normal	Tinggi

Tabel ini menampilkan data pengukuran suhu dan kelembaban yang diperoleh dari sensor *DHT11* yang terhubung dengan modul *ESP8266* (NodeMCU). Pengambilan data dilakukan tiap menit sekali. Suhu yang

tercatat masih dalam batas yang aman, namun pada pembacaan terakhir pukul 10:04 WIB, tingkat kelembaban mencapai 60,2% RH, yang melebihi ambang batas kelembaban yang disarankan. Karena itu, status kelembaban diberi label "Tinggi," menandakan adanya kemungkinan risiko bagi perangkat jika kondisi tersebut terus berlanjut.

D. Hasil pengujian sensor Acs712 Esp32 dan Esp8266

TABEL 5
(HASIL PENGUJIAN SENSOR ACS712ESP32)

No	Variabel Beban	Arus (A)	Daya (W)	Mengirim data ke database
1	Laptop (19v=2.37 A, 45 watt)	0.198759533	43.72710	Terkirim
2	Smartphone 25 W	0.082571044	18.16563	Terkirim
3	Dispenser 380W	1.248655606	274.70422	Terkirim

Tabel ini menunjukkan hasil pengujian pembacaan arus dan daya beberapa perangkat listrik menggunakan sensor ACS712 yang terhubung ke ESP32. Perangkat yang diuji meliputi laptop, smartphone, dan dispenser. Hasil pengujian menunjukkan bahwa sensor dapat membaca nilai arus dan menghitung daya untuk setiap perangkat. Semua data berhasil dikirim ke database.

TABEL 6
(HASIL PENGUJIAN SENSOR ACS712 ESP8266)

No	Variabel Beban	Arus (A)	Daya (W)	Mengirim data ke database
1	Laptop (19v=2.37 A, 45 watt)	0.992295186	218.30495	Terkirim
2	Smartphone (67 W max)	0.077942356	17.14732	Terkirim
3	Dispenser 380W	2.214542335	487.19931	Terkirim

Pengujian ini menggunakan sensor ACS712 yang terhubung dengan mikrokontroler ESP8266. Sensor tersebut digunakan untuk memantau arus dan menghitung daya listrik dari tiga perangkat. Hasil pengukuran menunjukkan bahwa semua data berhasil terkirim ke database. Meskipun nilai arus dan daya sedikit berbeda dengan pengujian dengan ESP32, sistem tetap berjalan lancar dan mampu menjalankan fungsi pemantauan dengan baik.

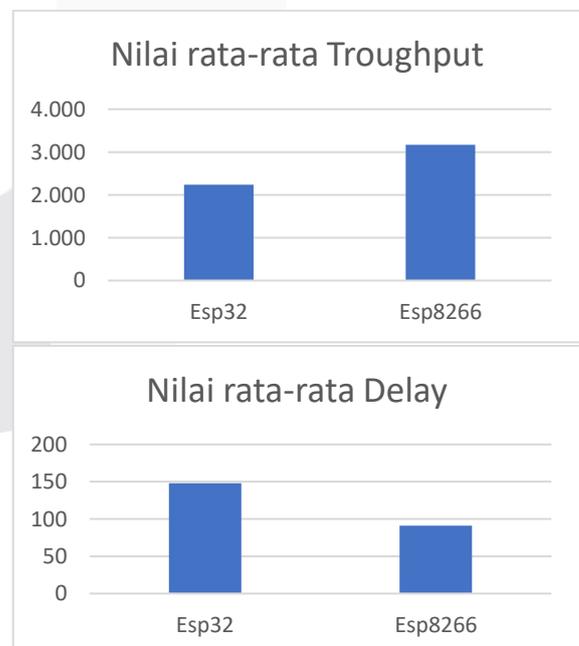
E. Hasil Pengamatan Nilai rata-rata Perbandingan *Quality of Service*

Perhitungan hasil pengamatan menunjukkan parameter *throughput esp32* dan *esp8266* sebagai berikut:

TABEL 7
(HASIL PERBANDINGAN QOS ESP32 DAN ESP8266)

Nilai rata-rata Troughput dan Delay Esp32					
No	Parameter QoS	Waktu	Nilai rata-rata	Index	Kategori
1	Troughput (kbps)	03 juni 2025, 21:36 WIB	2,239	4	Sangat Bagus
2	Delay (ms)	03 juni 2025, 21:36 WIB	148	4	Sangat Bagus
Nilai rata-rata Troughput Delay Esp8266					
1	Troughput (kbps)	03 juni 2025, 21:36 WIB	3,166	4	Sangat Bagus
2	Delay (ms)	03 juni 2025, 21:36 WIB	91	4	Sangat bagus

Grafik perbandingan nilai rata-rata perhitungan *throughput* dan *delay* antara *esp32* dan *esp8266* dapat dilihat pada gambar dibawah.



GAMBAR 11
(GRAFIK PERBANDINGAN QOS)

Dari perbandingan nilai *throughput* dan *delay* yang telah dilakukan maka mendapat kesimpulan bahwa *esp8266* lebih unggul dibandingkan dengan *esp32*

dengan selisih nilai rata-rata *throughput* 927 Kbps dan *delay* 57 ms.

V. KESIMPULAN

Berdasarkan hasil pada penelitian dan pembahasan yang telah disampaikan pada bab-bab sebelumnya, dapat disimpulkan sebagai berikut:

1. Sistem monitoring ini disimulasikan sebagai alat dalam melakukan monitoring ruang *server*. Dimana alat ini melakukan penerimaan data kondisi pada lingkungan ruang *server* serta mengirim data tersebut ke *database* dan ditampilkan pada aplikasi untuk dilakukan monitoring.
2. Sistem monitoring ini dibuat dengan beberapa komponen yaitu ESP32, ESP8266, DHT11, MQ2, ACS712, dan sensor flame api, serta beberapa komponen pendukung agar system dapat berjalan dengan semestinya.
3. Berdasarkan hasil pengujian terhadap parameter *throughput* dan *delay* dengan wireshark, dapat disimpulkan bahwa esp8266 lebih unggul dibandingkan dengan esp32 dengan selisih nilai rata-rata *throughput* 927 Kbps dan *delay* 57ms.

4. Hasil pengujian sistem monitoring dengan pengujian *blackbox* mendapat keberhasilan 100%. Pengujian fungsionalitas sistem monitoring mendapat keberhasilan 80%.

REFERENSI

- [1] H. Kusumah and R. A. Pradana, "PENERAPAN TRAINER INTERFACING MIKROKONTROLER DAN INTERNET OF THINGS BERBASIS ESP32 PADA MATA KULIAH INTERFACING," CERITA , no. 5, p. 121, 2019.
- [2] A. Rachmansyah, R. Satra and M. A. Mude, "Perancangan Alat Pemberi Makan dan Monitoring Sisa Pakan," Buletin Sistem Informasi dan Teknologi Islam, no. 3, p. 27, 2022.
- [3] S. M. Anif, D. N. Hayati and Y. , "Pengamanan Pintu Ruangan Menggunakan Arduino Mega 2560, MQ-2,," JURNAL RESTI, no. 1, p. 68, 2019.
- [4] N. Husin, "Rancang Bangun Alat Pendeteksi Kebocoran Gas," Jurnal Esensi Infokom , no. 5, p. 2, 2021.
- [5] R. Inggi and J. Pangala, "PERANCANGAN ALAT PENDETEKSI KEBOCORAN GAS LPG," Jurnal Sistem Informasi dan Sistem Komputer, no. 6, p. 15, 2021.
- [6] M. M and S. Moh, "PEMBUATAN ALAT UKUR RESISTIVITAS," Jurnal Mekanova, no. 7, p. 155, 2021.