

Implementasi mekanisme *sleep-wake* pada sistem pendeteksi kebocoran gas LPG

Nanda Rizky Mahardika^{1*}, Irma Nirmala², Kartika Sari³

^{1,2,3}Jurusan Rekayasa Sistem Komputer, Universitas Tanjungpura

Jl. Prof.Dr.H.Hadari Nawawi, Pontianak 78124, Kalimantan Barat, Indonesia

¹nrizky2@gmail.com, ²irma.nirmala@siskom.untan.ac.id, ³*kartika.sari@siskom.untan.ac.id

ABSTRAK

Liquefied Petroleum Gas (LPG) merupakan salah satu sumber energi yang paling banyak digunakan di seluruh dunia, baik di sektor rumah tangga maupun industri. Pemanfaatannya memberikan manfaat besar dalam pemenuhan kebutuhan energi, peningkatan kualitas hidup, dan mendukung perkembangan industri. Namun demikian, kebocoran pada tabung, regulator, maupun selang penghubung LPG dapat menimbulkan risiko kebakaran. Penelitian ini mengimplementasikan mekanisme *sleep-wake* pada sistem pendeteksi kebocoran gas berbasis mikrokontroler ESP32 untuk mengoptimalkan konsumsi daya pada perangkat elektronik, khususnya mikrokontroler dan sensor. Sensor MQ-6 digunakan untuk membaca kadar LPG di udara sekitar dalam satuan ppm, sementara sensor suara FC-04 berfungsi mendeteksi bunyi desis kebocoran dan mengirimkan sinyal *external interrupt* untuk membangunkan ESP32 dari kondisi *deep sleep*. Data hasil pengukuran dikirim ke server ThingSpeak dan peringatan dikirimkan melalui Telegram sesuai dengan kondisi kadar gas yang terdeteksi. Hasil penelitian menunjukkan bahwa sistem mampu mendeteksi kadar LPG dengan akurasi sebesar 97%. Sistem juga berhasil mengirimkan data kadar gas ke ThingSpeak dan notifikasi Telegram secara bersamaan. Berdasarkan pengujian, sistem dengan mode *deep sleep* dapat beroperasi selama 5 jam 47 menit (347 menit), sedangkan sistem tanpa *deep sleep* hanya bertahan 4 jam 11 menit (251 menit), dengan tingkat optimisasi daya sebesar 27,7%.

Kata kunci: *Liquified Patroleum Gas* (LPG), *deep sleep*, MQ-6, ThingSpeak, Telegram

ABSTRACT

Liquefied Petroleum Gas (LPG) is one of the most widely used energy sources worldwide, both in household and industrial sectors. Its utilization provides significant benefits in fulfilling energy demands, improving quality of life, and supporting industrial growth. However, leakage in LPG cylinders, regulators, or connecting hoses may pose a serious risk of fire. This study implements a *sleep-wake* mechanism in an LPG leakage detection system based on the ESP32 microcontroller to optimize power consumption of the electronic components, particularly the microcontroller and sensors. The MQ-6 sensor is employed to measure LPG concentration in the surrounding air in parts per million (ppm), while the FC-04 sound sensor detects the hissing sound of leakage and generates an *external interrupt* signal to wake the ESP32 from *deep sleep* mode. Measurement data are transmitted to the ThingSpeak server, and warning notifications are delivered to users via Telegram according to the detected gas levels. The results show that the system can detect LPG concentrations with an accuracy of 97%. Furthermore, the system successfully transmits gas concentration data to ThingSpeak and sends Telegram notifications in real time. Experimental results indicate that the system operating in *deep sleep* mode achieved a runtime of 5 hours and 47 minutes (347 minutes), while the system without *deep sleep* lasted only 4 hours and 11 minutes (251 minutes), resulting in a power optimization rate of 27.7%.

Keywords: *Liquefied Petroleum Gas* (LPG), *deep sleep*, MQ-6, ThingSpeak, Telegram

1. PENDAHULUAN

Liquefied Petroleum Gas (LPG) merupakan salah satu sumber energi yang banyak digunakan di seluruh dunia, baik pada sektor rumah tangga maupun industri. LPG merupakan campuran gas hidrokarbon yang pada kondisi atmosfer berbentuk gas. Karena adanya proses penurunan suhu dan penambahan tekanan, senyawa ini dapat berubah wujud menjadi cair [1]. Pemanfaatan LPG memberikan banyak keuntungan, antara lain pemenuhan kebutuhan energi, peningkatan kualitas hidup, serta dukungan terhadap perkembangan industri. Namun, kebocoran pada tabung, regulator, maupun selang penghubung LPG dapat menimbulkan risiko kebakaran. Kebocoran biasanya ditandai dengan bau gas yang menyengat, terbentuknya embun di sekitar tabung, atau bunyi desis pada regulator [2]. Dengan tekanan yang cukup tinggi, gas LPG akan cepat menyebar saat kebocoran terjadi.

Seiring meningkatnya kebutuhan sistem pemantauan lingkungan dan sistem terdistribusi, teknologi sensor nirkabel semakin berkembang karena mampu mengumpulkan data secara efisien dan berkelanjutan. Sensor nirkabel, atau sensor node, kini menjadi komponen utama dalam aplikasi *Internet of Things* (IoT) untuk pemantauan lingkungan, kualitas udara, hingga deteksi kebocoran gas. Tantangan utama dalam penggunaan sensor node adalah efisiensi energi, karena perangkat ini umumnya berbasis baterai dan beroperasi dalam waktu lama [3]. Salah satu pendekatan yang banyak digunakan untuk menghemat daya adalah mekanisme *sleep-wake*, di mana node sensor beralih antara mode *sleep* dan *wake* untuk menekan konsumsi energi [4].

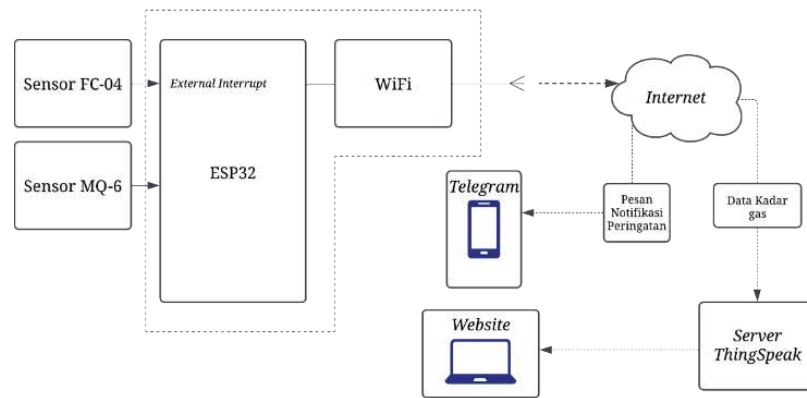
Penelitian sejenis yang sudah ada sebelumnya seperti pada penelitian [5] telah menerapkan *sleep-wake* berbasis *timer* pada mikrokontroler Atmega328p dengan bantuan modul RTC untuk menghasilkan sinyal *interrupt*. Sistem tersebut menggunakan GSM/GPRS sebagai media komunikasi untuk mengirim data sensor. Penelitian lain [6] dan [7] lebih berfokus pada aspek deteksi kebocoran gas berbasis IoT dengan *buzzer* atau modul aplikasi Blynk sebagai notifikasi kepada pengguna, namun belum menekankan pada upaya penghematan daya. Dengan demikian, terdapat celah penelitian (*research gap*) yaitu perlunya sistem deteksi kebocoran gas yang tidak hanya dapat mengirim data dan notifikasi secara *real-time*, tetapi juga hemat daya agar mampu beroperasi lebih lama menggunakan sumber energi baterai.

Berdasarkan latar belakang tersebut, penelitian ini mengimplementasikan mekanisme *sleep-wake* pada mikrokontroler ESP32. Sistem dirancang agar tetap berada pada mode *deep sleep* hingga terjadi *external interrupt* dari sensor suara FC-04 yang mendeteksi bunyi desis kebocoran gas. Selanjutnya, sensor MQ-6 membaca konsentrasi gas dalam satuan ppm dan mengirimkan data ke *server* ThingSpeak melalui koneksi WiFi. Notifikasi peringatan kebocoran juga dikirimkan secara *real-time* melalui aplikasi Telegram. Dengan kombinasi ini, sistem diharapkan mampu mendeteksi kebocoran gas secara akurat sekaligus mengoptimalkan konsumsi daya sehingga memperpanjang umur baterai.

2. METODE PENELITIAN

2.1 Perancangan Sistem

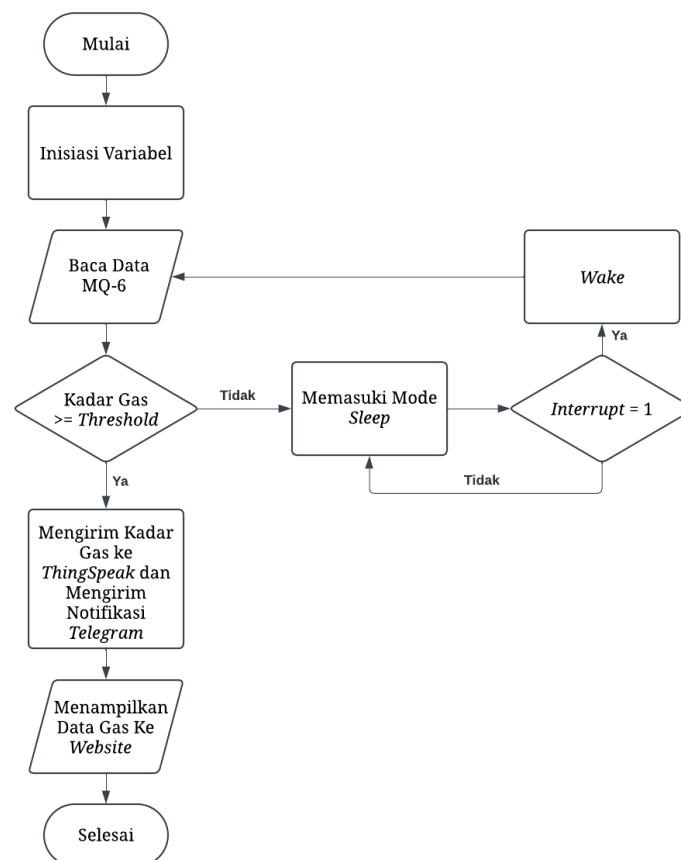
Gambar 1 menggambarkan rancangan umum dari sistem yang dibangun. Sensor suara FC-04 digunakan untuk mendeteksi suara desis yang menjadi indikasi kebocoran gas di sekitar tabung. Sensor ini menghasilkan keluaran berupa nilai digital, yaitu 1 atau 0. Keluaran dari sensor berfungsi sebagai *external interrupt* yang merupakan *wake-up source* untuk membangunkan sistem. Selanjutnya, sensor MQ-6 berfungsi untuk mengukur kadar/konsentrasi gas LPG di udara sekitar dengan satuan ppm (*parts per million*). Data hasil pengukuran diproses oleh mikrokontroler ESP32, kemudian dikirimkan ke *server* ThingSpeak melalui koneksi WiFi. Data yang tersimpan di *server* ThingSpeak dapat diakses oleh pengguna melalui aplikasi berbasis *web*. Selain itu, sistem juga mengirim notifikasi peringatan kebocoran gas kepada pengguna jika nilai kadar gas yang terbaca melebihi *threshold* yang telah ditentukan.



Gambar 1. Diagram blok sistem

2.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak merupakan perancangan alur kerja sistem yang diterapkan pada penelitian ini. Gambar 2 menampilkan diagram alir sistem yang menggambarkan tahapan kerja dari awal hingga akhir secara berurutan. Alur kerja dimulai dengan inisiasi variabel, *library*, dan konfigurasi pin sensor yang terhubung. Setelah itu, dilakukan pengecekan kondisi terhadap nilai kadar gas yang terbaca oleh sensor MQ-6. Jika nilai kadar gas dibawah *threshold*, maka sistem akan masuk ke mode *deep sleep* secara kontinyu. Jika sensor suara FC-04 mendeteksi suara kebocoran gas dan memberikan sinyal *interrupt*, sistem akan dibangunkan dari mode *deep sleep*. Sistem yang *wake* akan membaca kembali nilai kadar gas. Jika nilai melebihi *threshold*, maka ESP32 akan mengirim data kadar gas ke *server* ThingSpeak dan mengirim notifikasi Telegram ke pengguna.



Gambar 2. Rancangan diagram alir perangkat lunak

Dalam penelitian ini tidak digunakan simulasi perangkat keras melalui program khusus seperti Proteus atau MATLAB Simulink. Prototipe sistem langsung diwujudkan melalui perancangan dan implementasi pada perangkat keras nyata untuk memperoleh hasil pengujian yang lebih representatif. Namun, perancangan perangkat lunak (algoritma) dilakukan secara sistematis melalui *flowchart* sehingga alur kerja sistem telah tervalidasi secara logika sebelum diimplementasikan. Algoritma *sleep-wake* yang diterapkan bekerja berdasarkan dua kondisi utama:

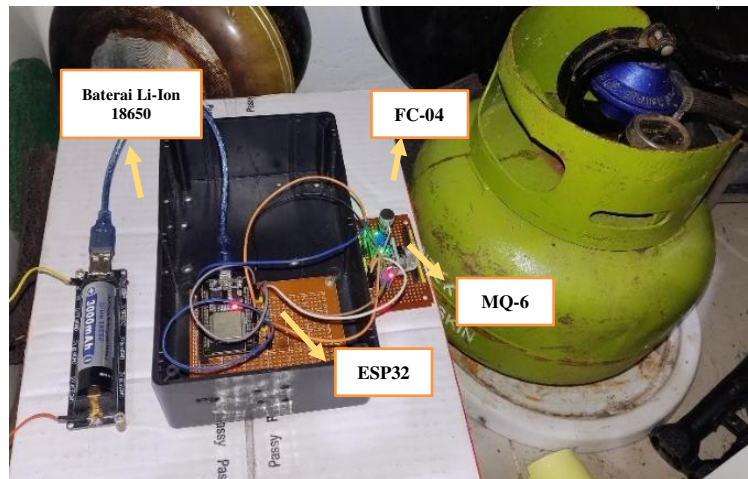
- a. *Mode Sleep (deep sleep)*: ESP32 dimasukkan ke mode hemat daya ketika nilai kadar gas berada di bawah ambang batas (*threshold*) dan tidak ada aktivitas pada sensor suara. Pada kondisi ini, CPU, sebagian besar RAM, dan perifer digital dinonaktifkan, hanya RTC dan ULP *coprocessor* yang tetap aktif untuk menjaga fungsi minimum.
- b. *Mode Wake*: Sistem dibangunkan oleh dua pemicu, yaitu (a) sinyal *external interrupt* dari sensor FC-04 ketika mendeteksi suara desis kebocoran gas, atau (b) nilai kadar gas MQ-6 yang melebihi *threshold*. Pada kondisi ini, ESP32 kembali aktif, melakukan pembacaan ulang kadar gas, dan jika terdeteksi melebihi ambang batas, data dikirim ke *server* ThingSpeak serta notifikasi dikirim ke pengguna melalui Telegram.

2.3 Implementasi Sistem

Implementasi sistem merupakan proses yang menggabungkan seluruh komponen, baik perangkat keras maupun perangkat lunak, sesuai dengan perancangan. Gambar 3 menunjukkan implementasi perangkat keras sistem yang terdiri dari mikrokontroler ESP32 yang terhubung dengan sensor MQ-6 dan FC-04. Modul sensor MQ-6 berfungsi untuk membaca konsentrasi gas di udara sekitar, sedangkan modul sensor FC-04 berfungsi untuk mendeteksi suara serta memberikan sinyal *interrupt* untuk membangunkan sistem. Sistem ini menggunakan baterai lithium-ion berkapasitas 3000 mAh sebagai sumber daya utama, sehingga efisiensi konsumsi daya menjadi aspek penting agar perangkat dapat beroperasi lebih lama.

Sensor MQ-6 bekerja dengan prinsip perubahan resistansi pada material semikonduktor SnO_2 yang sensitif terhadap molekul hidrokarbon. Pada kondisi normal resistansi tinggi, tetapi ketika terpapar LPG (propana/butana), resistansi menurun sehingga menghasilkan perubahan tegangan analog (Volt). Tegangan ini kemudian dibaca oleh ADC pada ESP32 dan dikonversi menjadi nilai konsentrasi gas dalam satuan ppm (*parts per million*). Dengan demikian, sinyal yang diproses ESP32 dari MQ-6 adalah tegangan analog yang mewakili kadar gas LPG di udara.

Sementara itu, sensor suara FC-04 dilengkapi mikrofon kondensor yang menangkap gelombang akustik dari bunyi desis kebocoran gas. Gelombang suara diubah menjadi tegangan listrik yang dibandingkan dengan nilai ambang (*threshold*) melalui komparator internal modul. Jika amplitudo suara melebihi ambang, sensor menghasilkan keluaran digital berupa logika HIGH/LOW (sekitar 0–3,3 V) yang dikirim ke pin eksternal-interrupt ESP32. Sinyal ini berfungsi untuk membangunkan mikrokontroler dari mode *deep sleep* sehingga sistem dapat segera melakukan pembacaan ulang terhadap kadar gas yang ada. Dengan mekanisme ini, sistem hanya aktif ketika benar-benar diperlukan, yaitu saat terdeteksi adanya indikasi kebocoran gas, sehingga konsumsi energi lebih efisien. Integrasi mode *sleep-wake* memungkinkan penghematan daya yang signifikan, terbukti dari hasil pengujian bahwa perangkat mampu beroperasi lebih lama pada mode *deep sleep* dibandingkan tanpa *sleep mode*.



Gambar 3. Implementasi perangkat keras sistem

2.4 Prosedur Pengujian

1. Prosedur pengujian kadar gas

Pengujian dilakukan dengan melakukan perbandingan nilai tegangan antara *analog-to-digital converter* (ADC) ESP32 dengan nilai yang diukur menggunakan digital multimeter. Pembacaan menggunakan multimeter dilakukan dengan menghubungkan *probe* hitam multimeter ke kaki pin *Ground* (GND) sensor MQ-6 dan *probe* merah ke kaki pin A0. Sedangkan pembacaan nilai tegangan ADC dilihat pada *serial monitor* Arduino IDE.

2. Prosedur pengujian pengiriman notifikasi

Pengujian diawali dengan menyiapkan *bot* Telegram pada aplikasi Telegram. Proses pembuatan *bot* bertujuan untuk mendapatkan API dan ID *bot* yang digunakan pada kode program untuk mengirimkan pesan peringatan kebocoran gas. Setelah melakukan inisialisasi *library* dan jaringan WiFi yang diperlukan, maka dibuat sebuah kondisi dimana pesan peringatan dikirimkan saat nilai kadar gas melebihi *threshold* yang ditentukan.

3. Prosedur pengujian konsumsi arus

Pengujian dilakukan dengan menggunakan multimeter digital untuk mengukur arus yang berjalan pada sistem. Pengukuran dilakukan dengan cara mengupas kabel USB dan memotong *line* VCC pada kabel. *Line* VCC pada kabel kemudian dihubungkan secara seri dengan *probe* multimeter untuk membaca nilai arus yang mengalir.

4. Prosedur pengujian durasi penggunaan daya

Pengujian durasi penggunaan daya dilakukan dengan melakukan pemantauan tegangan secara berkala menggunakan multimeter digital dan memantau durasi hidup perangkat mulai dari baterai terisi penuh hingga baterai habis. Simulasi kebocoran gas dilakukan dengan membocorkan gas pada pemantik api portabel terhadap perangkat.

3. HASIL DAN PEMBAHASAN

3.1 Pengujian Pembacaan Kadar Gas

Pada pengujian ini dilakukan pengujian terhadap sensor MQ-6 untuk melihat kinerja sensor dalam membaca kadar gas. Pengujian dilakukan dengan melakukan perbandingan nilai *analog-to-digital converter* (ADC) ESP32 dengan nilai yang diukur menggunakan digital multimeter. Pengujian dilakukan sebanyak 30 kali dengan membandingkan nilai tegangan antara ADC ESP32 dengan *digital multimeter*. Adapun perbandingan tegangan dapat dilihat pada Tabel 1.

Tabel 1. Perbandingan pengukuran tegangan

No	Hasil pengukuran		Galat absolut (%)	Galat relatif (%)	Akurasi (%)
	Nilai alat ukur (Volt)	Nilai sensor (Volt)			
1	0,75	0,69	0,06	8,69	91,31
2	1,15	1,16	0,01	0,86	99,14
3	0,14	0,19	0,05	4,2	95,8
4	1,70	1,78	0,08	4,49	95,51
5	1,44	1,34	0,1	7,46	92,54
6	1,46	1,5	0,04	2,74	97,26
7	0,63	0,62	0,01	1,59	98,41
8	0,48	0,46	0,02	4,16	95,84
9	1,20	1,10	0,1	8,33	91,67
10	1,28	1,30	0,02	1,56	98,44
11	1,48	1,46	0,02	1,35	98,65
12	0,89	0,81	0,08	8,98	91,02
13	1,24	1,27	0,03	2,4	97,6
14	1,26	1,29	0,03	2,38	97,62
15	1,31	1,33	0,02	1,53	98,47
16	0,36	0,38	0,02	5,55	94,45
17	0,57	0,58	0,01	1,75	98,25
18	0,71	0,71	0	0	100
19	0,78	0,79	0,01	1,28	98,72
20	0,79	0,8	0,01	1,25	98,75
21	0,83	0,85	0,02	2,41	97,59
22	0,14	0,14	0	0	100
23	0,86	0,89	0,03	3,48	96,52
24	1,03	1,05	0,02	1,94	98,06
25	1,08	1,08	0	0	100
26	0,95	0,99	0,04	4,21	95,79
27	1,07	1,1	0,03	2,8	97,2
28	0,07	0,07	0	0	100
29	0,87	0,85	0,02	2,29	97,71
30	1,06	1,08	0,02	1,89	98,11
Rata-rata			0,0027	2,99	97,01

Nilai galat relatif rata-rata yang didapat adalah 2,986% dengan akurasi rata-rata sebesar 97%. Selanjutnya dilakukan pengujian pengukuran kadar gas dengan satuan ppm (*parts-per-millions*). Pengujian dilakukan dengan memberikan paparan gas menggunakan pemantik api yang memiliki kandungan yang kurang lebih sama dengan yang terdapat pada tabung LPG. Adapun hasil pengujian dapat dilihat pada Tabel 2.

Pengujian kadar gas menggunakan sensor MQ-6 dilakukan dengan memanfaatkan pemantik api sebagai sumber gas uji, mengingat pemantik mengandung butana yang memiliki karakteristik serupa dengan LPG. Hasil pengujian pada Tabel menunjukkan adanya perbedaan signifikan antara kondisi pemantik dilepas dan ditekan. Pada kondisi pemantik dilepas (pengujian ke-1, ke-3, dan ke-5), sensor menunjukkan nilai relatif stabil pada kisaran 2-3 ppm. Nilai ini dapat dikategorikan sebagai kadar latar (*background*) gas yang secara alami ada di lingkungan, serta menunjukkan bahwa sensor tetap memberikan respons meskipun tidak ada paparan gas secara langsung.

Pada setiap siklus percobaan, pemantik dilepas selama ± 10 detik agar kondisi lingkungan kembali stabil sebelum dilakukan pengukuran berikutnya. Sebaliknya, pada kondisi pemantik ditekan (pengujian ke-2 dan ke-4), konsentrasi gas yang terbaca meningkat cukup signifikan dengan kisaran 10–18 ppm pada pengujian ke-2 dan 33–37 ppm pada pengujian ke-4. Peningkatan nilai ppm ini terjadi akibat pelepasan gas dari pemantik yang ditekan selama ± 5 detik pada jarak tetap ± 5 cm dari sensor, sehingga gas butana terdispersi ke area sekitar sensor dan meningkatkan konsentrasi dibandingkan kondisi normal.

Tabel 2. Pengukuran kadar gas

Pengujian ke-	Kondisi Pemantik	Kadar Gas (ppm)
1	dilepas	3
	dilepas	3
	dilepas	3
	dilepas	3
	dilepas	3
2	ditekan	10
	ditekan	12
	ditekan	15
	ditekan	17
	ditekan	18
3	dilepas	2
	dilepas	2
	dilepas	2
	dilepas	2
	dilepas	2
4	ditekan	37
	ditekan	36
	ditekan	36
	ditekan	36
	ditekan	33
5	dilepas	2
	dilepas	2
	dilepas	2
	dilepas	2
	dilepas	2

Namun demikian, nilai konsentrasi gas yang terukur masih dipengaruhi oleh variabel lain, seperti arah hembusan, sirkulasi udara, dan konsistensi tekanan pada pemantik. Faktor-faktor ini menyebabkan hasil pengukuran lebih merepresentasikan respon sensor terhadap adanya paparan gas dibandingkan mencerminkan konsentrasi absolut gas LPG secara presisi. Hasil ini menunjukkan bahwa sensor MQ-6 dapat mendeteksi indikasi kebocoran gas.

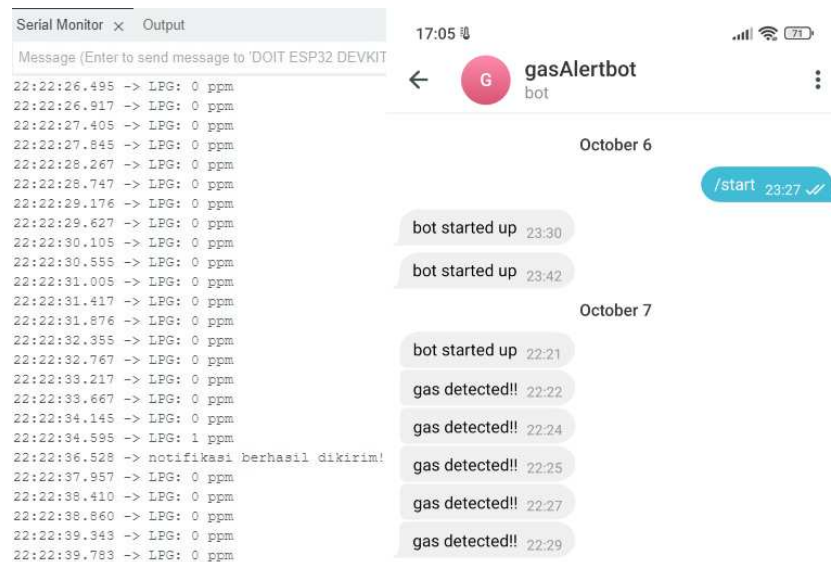
3.2 Pengujian Pengiriman Notifikasi

Pengujian pengiriman dilakukan untuk mengetahui tingkat keberhasilan dalam mengirim notifikasi ke *bot* Telegram. Pengujian dilakukan sebanyak 5 kali dan dilakukan perbandingan waktu saat sistem mengirim nilai yang terbaca oleh sensor dengan waktu terkirimnya pesan notifikasi. Adapun perbandingan waktu dapat dilihat pada Tabel 3.

Tabel 3. Perbandingan waktu pengiriman

No	Waktu pada <i>serial monitor</i> (WIB)	Waktu terkirimnya notifikasi (WIB)
1	22:22	22:22
2	22:24	22:24
3	22:25	22:25
4	22:27	22:27
5	22:29	22:29

Sistem berhasil mengirim pesan notifikasi sebanyak 5 dari 5 kali pengiriman pesan notifikasi Telegram pada waktu yang sama hingga skala menit. Tampilan pada aplikasi Telegram dan *serial monitor* dapat dilihat pada Gambar 4.



Gambar 4. Tampilan aplikasi Telegram dan serial monitor

3.3 Pengujian Konsumsi Arus

Pengujian dilakukan untuk mengetahui kebutuhan arus dengan menggunakan multimeter digital. Tujuan dari pengujian adalah untuk mengukur dan melakukan perbandingan konsumsi arus antara dua *state*, yaitu saat sistem dalam mode *deep sleep* dan saat sistem *wake* dari mode *deep sleep*. Dilakukan 30 kali pengukuran arus antara dua *state* secara berurutan, dengan membocorkan gas pada pemantik api untuk membangunkan sistem. Adapun hasil dari pengujian dapat dilihat pada Tabel 4.

Tabel 4. Perbandingan konsumsi arus

Pengujian Ke-	Konsumsi Arus Sleep (A)	Konsumsi Arus Wake (A)	Pengujian Ke-	Konsumsi Arus Sleep (A)	Konsumsi Arus Wake (A)
1	0,13	0,18	16	0,13	0,17
2	0,13	0,18	17	0,13	0,17
3	0,13	0,17	18	0,13	0,18
4	0,13	0,17	19	0,13	0,18
5	0,13	0,18	20	0,13	0,18
6	0,13	0,17	21	0,13	0,18
7	0,13	0,17	22	0,13	0,18
8	0,13	0,18	23	0,13	0,17
9	0,13	0,18	24	0,13	0,17
10	0,13	0,18	25	0,13	0,17
11	0,13	0,17	26	0,13	0,18
12	0,13	0,18	27	0,13	0,18
13	0,13	0,18	28	0,13	0,18
14	0,13	0,18	29	0,13	0,17
15	0,13	0,18	30	0,13	0,18
Rata-rata				0,13 A	0,177 A
Selisih Arus				0,047 A	

Hasil rata-rata konsumsi arus saat dalam kondisi *sleep* adalah sebesar 0,13 A dan rata-rata konsumsi arus saat dalam kondisi *wake* adalah sebesar 0,177 A, dimana didapat selisih rata-rata konsumsi arus sebesar 0,047 A. Konsumsi arus sistem dalam kondisi *sleep* memiliki nilai yang konstan dan lebih rendah dibandingkan dengan sistem dalam kondisi *wake*. Konsumsi arus yang lebih rendah secara langsung berkontribusi pada pengurangan konsumsi daya, sehingga memungkinkan efisiensi energi yang lebih baik dan memperpanjang masa pakai sumber daya.

3.4 Pengujian Durasi Penggunaan Daya

Pengujian bertujuan untuk mengetahui durasi penggunaan daya pada sistem. Pengujian dilakukan dalam dua skenario: skenario pertama menggunakan mode *deep sleep*, sedangkan skenario kedua tanpa menggunakan *deep sleep*. Dilakukan pengamatan tegangan baterai secara berkala mulai dari kondisi baterai penuh dan alat dihidupkan hingga baterai habis dan alat mati. Pengukuran tegangan dilakukan dengan *interval* waktu ± 30 menit sekali menggunakan multimeter digital. Simulasi kebocoran gas juga dilakukan beberapa kali menggunakan pemantik api portabel untuk menguji fungsionalitas sistem dalam mengirimkan data ke ThingSpeak dan notifikasi Telegram. Adapun hasil pengamatan tegangan dapat dilihat pada Tabel 5.

Tabel 5. Pengamatan tegangan (*deep sleep*)

No	Timestamp	Tegangan (V)	Kadar gas (ppm)	Thing Speak (time-stamp)	Telegram (timestamp)	External Interrupt	State	
							Wake	Sleep
1	21:23	4,2	-	-	-	0	✕	✓
2	21:53	4,15	-	-	-	0	✕	✓
3	22:25	4,08	68	22:25	22:25	1	✓	✕
4	22:53	4,05	-	-	-	0	✕	✓
5	23:23	4,03	-	-	-	0	✕	✓
6	23:51	3,99	40	23:51	23:51	1	✓	✕
7	00:23	3,95	-	-	-	0	✕	✓
8	00:53	3,92	-	-	-	0	✕	✓
9	01:24	3,9	34	01:24	01:24	1	✓	✕
10	01:53	3,86	-	-	-	0	✕	✓
11	02:23	3,81	-	-	-	0	✕	✓
12	02:53	3,64	-	-	-	0	✕	✓
13	03:10	2,96	-	-	-	0	✕	✓
Durasi: 5 jam 47 menit (347 menit)								

Tabel 5 menunjukkan pengamatan tegangan dan kadar gas pada sistem yang menggunakan mode *deep sleep*, termasuk durasi waktu sistem dalam keadaan tidur (*sleep*) dan terjaga (*wake*). Setiap baris pada tabel mencatat waktu (*timestamp*), tegangan (V), kadar gas (ppm), waktu pengiriman data ke ThingSpeak dan Telegram, serta status eksternal interrupt dan keadaan sistem (apakah dalam status *wake* atau *sleep*). Dari pengamatan tersebut, dapat dilihat bahwa sistem menggunakan daya dalam mode *deep sleep* selama 5 jam 47 menit atau 347 menit. Mode *deep sleep* ini menunjukkan efisiensi daya yang tinggi, dengan sistem lebih banyak berada dalam keadaan tidur (*sleep*), di mana tidak ada pengambilan data yang dilakukan, hanya dalam waktu-waktu tertentu sistem terbangun untuk mengirim data ke *server* dan memantau kadar gas.

Tabel 6. Pengamatan tegangan (normal)

No	Timestamp	Tegangan (V)	Kadar Gas (ppm)	ThingSpeak (timestamp)	Telegram (timestamp)
1	23:13	4,2	-	-	-
2	23:43	4,07	-	-	-
3	00:12	4,03	13	00:12	00:12
4	00:43	4	-	-	-
5	01:13	3,95	-	-	-
6	01:42	3,9	11	01:42	01:42
7	02:13	3,85	-	-	-
8	02:43	3,8	-	-	-
9	03:13	3,62	44	03:13	03:13
10	03:25	2,96	-	-	-
Durasi: 4 Jam 11 Menit (251 Menit)					

Pada sistem yang tidak menggunakan *deep sleep* (normal), durasi penggunaan daya adalah selama 4 jam 11 menit atau 251 menit. Sedangkan pada sistem yang menggunakan *deep sleep*, durasi penggunaan daya adalah selama 5 jam 47 menit atau 347 menit. Dengan diketahui kapasitas baterai sebesar 300 mAh, maka konsumsi daya rata-rata pada masing-masing skenario dapat dihitung menggunakan persamaan (1).

$$P = \frac{E}{t} \quad (1)$$

$$P_{\text{deep sleep}} = \frac{3000 \text{ mAh}}{347 \text{ menit}} \approx 8,64 \text{ mA/menit}$$

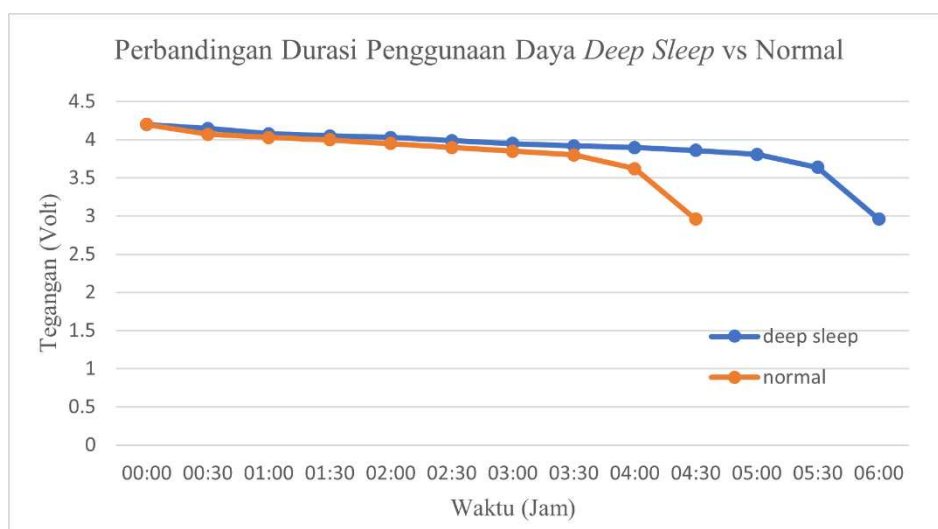
$$P_{\text{normal}} = \frac{3000 \text{ mAh}}{251 \text{ menit}} \approx 11,95 \text{ mA/menit}$$

Untuk mengetahui daya baterai pada masing-masing pengujian, maka dapat dicari persentase optimisasi baterai menggunakan persamaan (2).

$$\text{persentase optimisasi daya} = \left(\frac{P_{\text{normal}} - P_{\text{deep sleep}}}{P_{\text{normal}}} \right) \times 100 \quad (2)$$

$$\text{persentase optimisasi daya} = \left(\frac{11,95 - 8,64}{11,95} \right) \times 100 \approx 27,7\%$$

Didapat persentase optimisasi daya sesuai perhitungan adalah sebesar 27,7%. Kemudian dilakukan simulasi kebocoran gas dengan total sebanyak 6 kali percobaan, dimana sistem berhasil mengirim data kadar gas ke ThingSpeak dan mengirim pesan notifikasi Telegram di setiap percobaan. Grafik perbandingan durasi penggunaan daya antara sistem yang menggunakan mode *deep sleep* dan sistem yang tidak menggunakan mode *deep sleep* (normal) dapat dilihat pada Gambar 5.



Gambar 5. Perbandingan durasi penggunaan daya *deep sleep* vs. normal

Gambar 5 menunjukkan grafik perbandingan durasi penggunaan daya antara sistem yang menggunakan mode *deep sleep* dengan sistem yang berjalan pada mode normal. Sumbu horizontal (X) merepresentasikan waktu pengamatan dalam satuan jam, sedangkan sumbu vertikal (Y) menunjukkan tegangan baterai dalam volt (V). Terlihat bahwa baik pada mode normal maupun *deep sleep*, tegangan baterai menurun seiring bertambahnya waktu penggunaan. Namun, laju penurunan tegangan pada sistem normal lebih cepat dibandingkan sistem *deep sleep*. Pada sekitar jam ke-4, tegangan baterai pada sistem normal sudah turun mendekati batas minimum (sekitar 3,0 V) sehingga sistem berhenti beroperasi.

Sebaliknya, sistem dengan *deep sleep* masih mampu bertahan hingga mendekati jam ke-6 sebelum akhirnya kehabisan daya. Hasil ini sejalan dengan perhitungan persentase optimisasi daya sebesar 27,7%, yang berarti sistem *deep sleep* dapat beroperasi lebih lama sekitar 1 jam 38 menit dibandingkan sistem tanpa *deep sleep*.

Jika dibandingkan dengan penelitian-penelitian sebelumnya, sebagian besar sistem deteksi kebocoran gas hanya berfokus pada fungsi monitoring dan pemberian notifikasi tanpa memperhatikan aspek efisiensi energi. Ada pula penelitian yang telah menerapkan mode hemat daya, namun pendekatannya masih terbatas pada pengaturan waktu atau penggunaan mikrokontroler dengan fitur *low power* tanpa integrasi IoT. Selain itu, beberapa sistem deteksi gas berbasis IoT lebih menekankan pada komunikasi data dan notifikasi melalui aplikasi, tetapi belum mengoptimalkan konsumsi daya sehingga durasi operasional perangkat berbasis baterai relatif singkat. Perbedaan utama penelitian ini terletak pada kombinasi mekanisme *sleep-wake* berbasis *external interrupt* sensor suara dengan integrasi sensor gas MQ-6, komunikasi berbasis WiFi, serta notifikasi *real-time* melalui Telegram. Pendekatan ini tidak hanya memastikan deteksi kebocoran gas yang akurat, tetapi juga menghasilkan efisiensi energi yang lebih baik sehingga perangkat dapat beroperasi lebih lama dengan sumber daya terbatas.

4. KESIMPULAN

Mekanisme *sleep-wake* dapat mengoptimalkan penggunaan daya dengan menidurkan sistem (*sleep*) saat tidak ada peristiwa yang memerlukan aksi. Selama berada pada mode *sleep*, CPU, sebagian besar RAM, dan periferal digital dinonaktifkan, sehingga konsumsi daya menjadi lebih rendah dan masa penggunaan sistem dapat diperpanjang. Hasil pengujian menunjukkan bahwa sistem yang mengimplementasikan mekanisme *sleep-wake* mampu bertahan dengan durasi operasi selama 5 jam 47 menit, sedangkan sistem tanpa *deep sleep* hanya mampu beroperasi selama 4 jam 11 menit. Perbedaan ini menunjukkan adanya persentase optimisasi daya sebesar 27,7%. Durasi operasi tersebut dihitung berdasarkan kapasitas baterai penuh (100%) hingga habis (0%), sehingga mencerminkan performa nyata dari sistem dalam kondisi pemakaian penuh.

Selain itu, sistem yang dirancang berhasil mendeteksi kebocoran gas dalam satuan ppm menggunakan sensor MQ-6, mengirimkan data kadar gas ke *server* ThingSpeak, serta memberikan notifikasi peringatan melalui aplikasi Telegram. Dengan efisiensi konsumsi daya yang lebih baik, sistem ini dapat beroperasi lebih lama dibandingkan sistem tanpa *deep sleep* dan tetap menjalankan fungsi pemantauan gas secara efektif. Untuk pengembangan berikutnya, sistem dapat ditingkatkan dengan memanfaatkan versi ESP32 terbaru atau menggunakan modul komunikasi nirkabel berdaya rendah seperti *Bluetooth Low Energy* (BLE), sehingga durasi operasi dapat diperpanjang lebih jauh dan konsumsi energi dapat ditekan.

REFERENSI

- [1] R. Hilman, "Mengetahui Lebih Lanjut Terkait LPG sebagai Bahan Bakar Rumah Tangga," <https://solarindustri.com/blog/apa-itu-lpg/>. Diakses: 21 Januari 2025. [Daring]. Tersedia pada: <https://solarindustri.com/blog/apa-itu-lpg/>
- [2] E. H. Firmansyah, D. Syauqy, dan S. R. Akbar, "Implementasi Low Power Mode Pada Perangkat Sistem Pendeteksi Dini Kebocoran Gas Menggunakan ATmega328p," *Jurnal Pengembangan Teknologi dan Ilmu Komputer*, vol. 2, no. 9, hlm. 2751–2759, Sep 2018, [Daring]. Tersedia pada: <http://j-ptiik.ub.ac.id>
- [3] D. Syafril, Y. P. Wijaya, dan P. Madona, "Penghematan Daya Pada Sensor Node Menggunakan Metode Pengaturan Waktu Kirim Data," *Jurnal Teknik Elektro dan Komputer*, no. 1, hlm. 18–28, Apr 2013, [Daring]. Tersedia pada: <https://www.researchgate.net/publication/268926119>
- [4] A. P. Wirawan, M. A. Saktiono, dan A. A. Wahab, "Penghematan Konsumsi Daya Node Sensor Nirkabel untuk Aplikasi Structural Health Monitoring Jembatan," *SNATIKA*, vol. 3, hlm. 36–40, 2015.
- [5] P. R. Santoso, S. R. Akbar, dan E. S. Pramukantoro, "Implementasi Mekanisme Sleep-Wake Pada Node Sensor Berbasis GSM," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 4, hlm. 3358–3367, Apr 2019, [Daring]. Tersedia pada: <http://j-ptiik.ub.ac.id>
- [6] A. A. Rosli dan M. F. L. Abdullah, "Automated Smoke and Gas Leakage Detector with IoT Monitoring System in Rural Area," *Evolution in Electrical and Electronic Engineering*, vol. 3, no. 1, hlm. 18–28, Jun 2022, doi: 10.30880/eeee.2022.03.01.003.
- [7] N. Evalina dan H. A. Azis, "Implementation and design gas leakage detection system using ATmega8 microcontroller," *IOP Conf Ser Mater Sci Eng*, vol. 821, no. 1, hlm. 012049, Apr 2020, doi: 10.1088/1757-899X/821/1/012049.

-
- [8] Ultra Librarian, "ATmega328p Pinout: Architecture and Application," <https://www.ultralibrarian.com/2023/03/16/atmega328p-pinout-architecture-and-application-ulc>.
- [9] Suprianto, "Pengertian Daya Listrik dan Rumus untuk Menghitungnya," <https://blog.unnes.ac.id/antosupri/pengertian-daya-listrik-dan-rumus-untuk-menghitungnya/>.
- [10] A. Shrivastava, R. Prabhaker, R. Kumar, dan R. Verma, "GSM BASED GAS LEAKAGE DETECTION SYSTEM," *International Journal of Technical Research and Applications*, vol. 1, no. 2, hlm. 42–45, Mei 2013, [Daring]. Tersedia pada: www.ijtra.com
- [11] R. Malaric, *Instrumentation and Measurement in Electrical Engineering*. Boca Raton: Brown Walker Press, 2011.
- [12] ezContents, "Programming ATmega328P Barebone," <https://ezcontents.org/programming-atmega328p-barebone>.
- [13] R. Brown, "ATMega328P Microcontroller Pinout & Features," <https://nerdytechy.com/atmega328p-pinout/>.
- [14] The MathWorks Inc., "Configure Accounts and Channels," <https://www.mathworks.com/help/ThingSpeak/users.html>.
- [15] Zhengzhou Winsen Electronics Technology, "Flammable Gas Sensor: MQ-6 (Manual)," Zhengzhou, Jul 2021. [Daring]. Tersedia pada: www.winsen-sensor.com
- [16] R. T. Yunardi, Winarno, dan Pujiyanto, "Analisa Kinerja Sensor Inframerah dan Ultrasonik untuk Sistem Pengukuran Jarak pada Mobile Robot Inspection," *SETRUM*, vol. 6, no. 1, hlm. 33–41, Jun 2017.
- [17] O. B. Kharisma dan H. B. P. Utama, "PENGEMBANGAN SISTEM PENGAMAN PINTU LABORATORIUM ROBOTIKA UIN SULTAN SYARIF KASIM BERDASARKAN SIULAN BERBASIS SENSOR FC-04 DAN MIKROKONTROLER ATMEGA 328," *JST (Jurnal Sains dan Teknologi)*, vol. 7, no. 1, hlm. 114–125, Apr 2018, doi: 10.23887/jstundiksha.v7i1.12930.
- [18] N. Anggraini, F. Fahrianto, dan A. Zulni, "SISTEM PENDETEKSI KEBOCORAN GAS DAN KUALITAS UDARA DI LABORATORIUM PENDIDIKAN KIMIA UIN SYARIF HIDAYATULLAH JAKARTA," *Seminar Prosiding SENATKOM*, Nov 2015.