pISSN: 2442-3386 eISSN: 2442-4293
Vol 11 No 2 Jul – Des 2025, 203 - 214
Melek IT
INFORMATION TECHNOLOGY JOURNAL

SISTEM PENDETEKSI DINI KEBOCORAN GAS BERBASIS IOT DENGAN KONEKSI KE *WEB SERVER*

Muhammad Fahrul Anam[1] , Firman Hadi Sukma Pratama[2]*
[1]Program Studi Informatika, Universitas Wijaya Kusuma Surabaya, kjalu898@gmail.com
[2]Program Studi Informatika, Universitas Wijaya Kusuma Surabaya, firmanpratama@uwks.ac.id
*) Korespondensi: firmanpratama@uwks.ac.id

Abstrak

Kebocoran gas dalam lingkungan domestik dan komersial mengancam keselamatan manusia dan aset. Penelitian ini menghadirkan rancangan dan implementasi sistem pendeteksi kebocoran gas *G2D(Gas Detection Device)* inovatif dengan sensor gas MQ2. Mikrokontroler NodeMCU berperan sebagai otak sistem untuk mengambil, memproses, dan mengirim data deteksi gas melalui koneksi *Internet of Things (IoT).* Fitur utama sistem ini adalah penerapan komunikasi *real-time* melalui *server*, memungkinkan respons cepat terhadap ancaman kebocoran gas. Integrasi *IoT* dan *server* yang memungkinkan akses dan kontrol jarak jauh, meningkatkan fleksibilitas dalam pemantauan keamanan. Pengujian dilakukan dengan mensimulasikan berbagai skenario kebocoran gas, menghasilkan kinerja dalam mendeteksi dan memberikan pemberitahuan melalui *Web Server*. Sistem pendeteksi kebocoran gas *G2D* ini berpotensi meningkatkan keselamatan sehari-hari melalui teknologi sensor MQ2, *IoT*, dan komunikasi *Web Server* yang efisien.
Kata Kunci : Sensor MQ2, *Internet of Things*, *G2D(Gas Detection Device)*, *Web Server*.

Abstract

Gas leaks in domestic and commercial environments pose a threat to human safety and assets. This research presents the design and implementation of an innovative gas leak detection system, the G2D (Gas Detection Device), using the MQ2 gas sensor. The NodeMCU microcontroller acts as the brain of the system to capture, process, and transmit gas detection data via the Internet of Things (IoT) connection. The main feature of this system is the implementation of real-time communication through a server, enabling quick responses to gas leak threats. The integration of IoT and the server allows for remote access and control, enhancing flexibility in security monitoring. Testing was conducted by simulating various gas leak scenarios, demonstrating performance in detecting and providing notifications via the Web Server. The G2D gas leak detection system has the potential to improve daily safety through the efficient use of MQ2 sensors, IoT, and Web Server communication.

Keywords: *MQ2 Sensor, Internet of Things, G2D (Gas Detection Device), Web Server.*

I. PENDAHULUAN

Pada saat ini, dengan pesatnya perkembangan teknologi, kebutuhan akan sistem keamanan dan pemantauan semakin meningkat, terutama terkait penggunaan gas dalam berbagai industri dan lingkungan domestik. Kebocoran gas merupakan salah satu bahaya potensial yang sering dihadapi, yang dapat menyebabkan kecelakaan serius, kerusakan lingkungan, dan bahkan kehilangan nyawa. Oleh karena itu, pengembangan sistem deteksi kebocoran gas yang andal dan efektif menjadi sangat penting.

Internet of Things (IoT) telah mengubah cara kita menghubungkan dan mengendalikan perangkat di berbagai lingkungan. Dengan konektivitas canggih dan infrastruktur jaringan yang berkembang, perangkat-perangkat ini dapat saling berkomunikasi. Mengintegrasikan sistem deteksi kebocoran gas dengan teknologi *IoT* memberikan keunggulan dalam hal pemantauan jarak jauh, respon cepat, dan pengumpulan data yang akurat.

Dengan koneksi ke *web server*, data yang dikumpulkan oleh sistem deteksi kebocoran gas dapat diakses dengan mudah oleh pengguna dari berbagai lokasi. Informasi tentang kebocoran gas, lokasi kebocoran, dan data terkait lainnya dapat disajikan dalam format yang mudah dimengerti, memungkinkan pengambilan keputusan yang cepat dan efisien. Koneksi ini juga memungkinkan adanya pemberitahuan dan peringatan otomatis melalui *web server* ketika terdeteksi adanya kebocoran gas, memungkinkan petugas atau pengguna untuk segera mengambil tindakan.

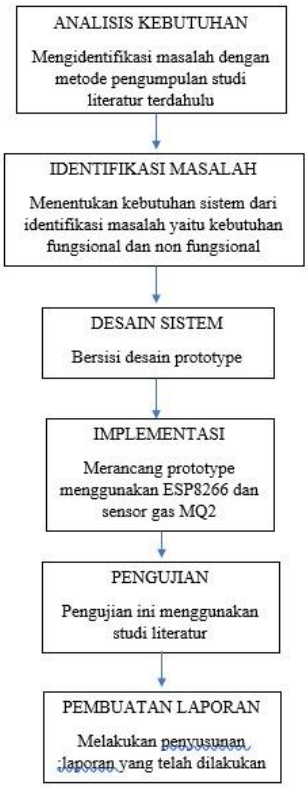
Melalui koneksi ke *web server*, data dari sensor deteksi kebocoran gas dapat diakses dan dipantau secara *real-time* dari perangkat apa pun yang terhubung ke internet, seperti komputer. Informasi mengenai kebocoran gas, tingkat konsentrasi gas, dan lokasi deteksi disajikan secara *real-time*. Ini akan membantu

pengguna untuk segera mengambil tindakan dan mengkoordinasikan respons yang diperlukan, seperti memanggil petugas. Alat ini diberi nama *G2D (Gas Detection Device)*

II. METODE

2.1 Tahapan Penelitian

Penelitian ini bertujuan untuk merancang dan membangun sistem deteksi kebocoran gas berbasis *IoT* dengan koneksi ke *web server*. Tahapan penelitian meliputi identifikasi masalah, analisis kebutuhan fungsional dan non fungsional, *flowchart*, *modeling quick design*, implementasi, pengujian pembuatan laporan. Penelitian ini diharapkan dapat memberikan manfaat bagi pengembangan teknologi *IoT* dan meningkatkan keselamatan dalam penggunaan gas. Adapun tahapan penelitian ditunjukkan oleh Gambar 1.



Gambar 1. Tahapan Penelitian

2.2 Identifikasi Masalah

Penelitian ini bertujuan untuk merancang dan membangun sistem deteksi kebocoran gas berbasis *IoT* dengan koneksi ke *web server*. Identifikasi masalah yang ada dalam proses ini meliputi beberapa langkah, (1) Studi literatur dan perancangan sistem: Penelitian ini melibatkan studi literatur tentang deteksi kebocoran gas dan sistem berbasis IoT. Perancangan sistem akan dilakukan berdasarkan hasil studi literatur tersebut. (2) Pengumpulan komponen: Penelitian ini akan mengumpulkan komponen-komponen yang diperlukan, seperti sensor gas, *mikrokontroler*, dan modul *IoT*. (3) Rancang bangun *prototype*: Penelitian ini akan merakit komponen-komponen tersebut menjadi sebuah *prototype* sistem deteksi kebocoran gas. (4) Pengujian *prototype*: Penelitian ini akan melakukan pengujian terhadap *prototype* untuk memastikan fungsionalitas dan kinerjanya. (5) Penyusunan laporan: Penelitian ini akan menyusun laporan berisi hasil studi, perancangan, pengujian, analisis, dan pemantauan *prototype*.

2.3 Analisis Kebutuhan

Setelah melakukan identifikasi masalah, penelitian ini akan merancang dan membangun sistem pendeteksi dini kebocoran gas berbasis *IoT* dengan koneksi ke *web server*. Berdasarkan data dan informasi yang ditemukan, sistem ini dapat mendeteksi kebocoran gas secara akurat, mengirimkan notifikasi ketika terjadi kebocoran gas, terhubung ke *web server* dan mengirimkan data deteksi kebocoran gas secara *real-time*. Analisis kebutuhan ini akan membantu dalam merancang dan membangun sistem pendeteksi dini kebocoran gas berbasis IoT yang efektif dan sesuai dengan kebutuhan pengguna. Maka analisis kebutuhan akan dipecah menjadi 2 bagian yaitu Analisis Kebutuhan Fungsional dan Non Fungsional.

A. Kebutuhan Fungsional

Analisa kebutuhan fungsional adalah penjelasan tentang fitur atau layanan yang disediakan oleh sistem untuk memudahkan pengguna dalam menggunakannya. Kebutuhan fungsional terdiri dari proses, *input*, *output*, *database*, dan kebutuhan pengguna. Dalam penelitian ini, sistem yang diusulkan adalah sistem pendeteksi dini kebocoran gas berbasis *IoT* dengan koneksi ke *web server*. Fungsi-fungsi utama yang akan diperkenalkan dalam sistem ini meliputi kemampuan untuk mendeteksi kebocoran gas secara akurat dan cepat, mengirimkan notifikasi ketika terjadi kebocoran gas, terhubung ke *web server* dan mengirimkan data deteksi kebocoran gas secara *real-time*, serta dapat dioperasikan dengan mudah dan intuitif. Dalam proses pengembangan sistem ini, beberapa langkah yang akan diambil meliputi memilih komponen sistem, mengembangkan perangkat lunak yang diperlukan, membuat *Source Code* untuk mengidentifikasi kebocoran gas, mengimplementasi-kan sistem ke *monitoring* kebocoran gas secara *real-time*, dan melakukan pengujian validasi sistem. Analisa kebutuhan fungsional ini akan membantu dalam merancang dan membangun sistem pendeteksi dini kebocoran gas berbasis *IoT* yang efektif dan sesuai dengan kebutuhan pengguna. Adapun proses yang ada pada sistem ini nanti yaitu perancangan sensor gas MQ2 berbasis iot ke *web server*, ini memiliki fitur yang bisa mendeteksi adanya ke bocoran gas *G2D (Gas Detection Device)* dan mengirim notifikasi ke *webserver*.

B. Kebutuhan Non Fungsional

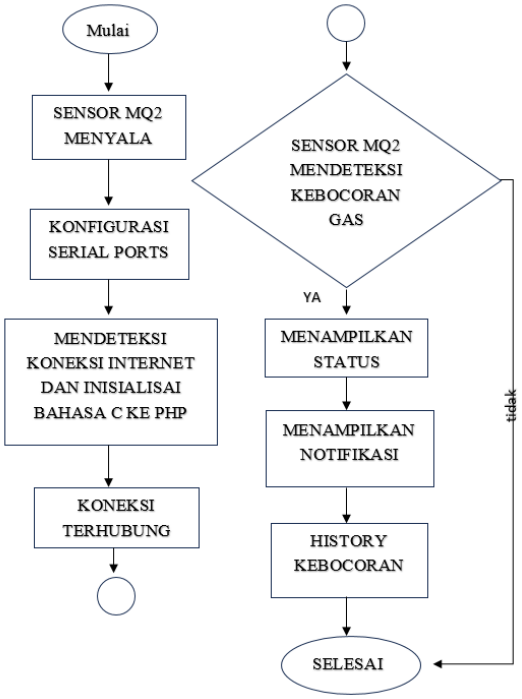
Kebutuhan non-fungsional adalah kriteria atau batasan yang dikenakan pada sistem, seperti batasan waktu,

pISSN: 2442-3386 eISSN: 2442-4293
Vol 11 No 2 Jul – Des 2025, 205 - 214
Melek IT
INFORMATION TECHNOLOGY JOURNAL

batasan pengembangan proses, standarisasi, spesifikasi hardware, dan kebutuhan software. Kebutuhan ini menentukan atribut kualitas perangkat lunak dan dapat berupa kendala atau persyaratan yang mempengaruhi bagaimana sistem beroperasi di masa depan. Kebutuhan non-fungsional sangat penting untuk memastikan bahwa sistem yang dikembangkan memenuhi kebutuhan pengguna dan beroperasi sesuai dengan harapan. Kebutuhan non-fungsional dapat mencakup spesifikasi hardware dan software yang sesuai, serta kemudahan penggunaan sistem dan keamanan sistem.

2.4 *Flowchart* Sistem

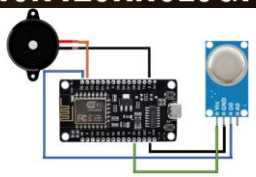
Flowchart sistem adalah representasi visual dari alur suatu proses, yang meliputi langkah-langkah, keputusan, dan urutan aktivitas. Diagram ini membantu dalam menjelaskan alur kerja suatu sistem atau proses, sehingga mempermudah pengambilan keputusan dan identifikasi faktor risiko serta faktor keberhasilan. Adapun *Flowchart* sistem yang digunakan pada sistem ditunjukkan oleh Gambar 2.



Gambar 2. *Flowchart* Sistem

2.5 *Modeling Quick Design*

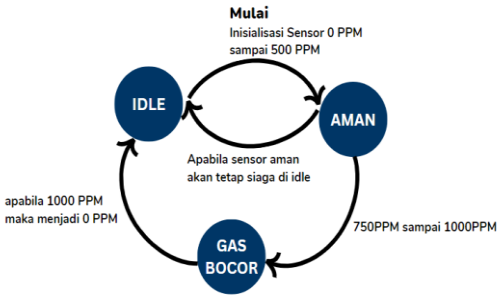
Tahapan ini akan menggambarkan desain rancangan rangkaian sistem secara kasar atau gambaran umum, Tahap *modeling quick design* ini menjadi pijakan awal yang penting dalam mengarahkan proses perancangan dan pengembangan selanjutnya, sehingga sistem dapat dirancang dan di implementasikan dengan lebih sistematis dan efisien. Adapun pemodelan desain ditunjukkan oleh Gambar 3.



Gambar 3. *Modeling Quick Design*

2.6 *State Diagram*

State Diagram adalah diagram yang menggambarkan perilaku sistem dengan menunjukkan kondisi-kondisi yang mungkin dialami oleh sebuah objek dan peristiwa yang terkait dengannya. Diagram ini terdiri dari lingkaran yang mewakili kondisi suatu objek dan panah yang menunjukkan perubahan dari satu kondisi ke kondisi lainnya. Aktivitas yang dilakukan objek dalam setiap kondisi juga ditampilkan. Dalam pendekatan *Object-Oriented (OO)*, *State Diagram* menunjukkan urutan kejadian yang dialami oleh satu objek selama hidupnya dalam merespons peristiwa tertentu. Elemen dasar State Diagram adalah state (kondisi) dan transisi (perubahan dari satu kondisi ke kondisi lainnya). State diagram sistem ditunjukkan oleh Gambar 4.



Gambar 4. *State Diagram*

Diagram di atas menunjukkan diagram status (state diagram) untuk sebuah sistem deteksi kebocoran gas dengan tiga status utama: IDLE, AMAN, dan GAS BOCOR. Berikut adalah penjelasan untuk setiap status dan transisi antar status tersebut:

1. IDLE (Inisialisasi)
   Sensor diinisialisasi dari 0 PPM hingga 500 PPM.
2. AMAN (Aman)
   Sensor mendeteksi rentang aman dari 0 PPM hingga 500 PPM. Jika sensor mendeteksi kondisi aman, sistem akan tetap di status AMAN atau kembali ke IDLE. Jika sensor mendeteksi rentang 750 PPM hingga 1000 PPM (indikasi kebocoran gas), sistem akan beralih ke status GAS BOCOR.
3. GAS BOCOR
   Sensor mendeteksi kebocoran gas dalam rentang 750 PPM hingga 1000 PPM. Jika kondisi ini terjadi, sistem berada di status GAS BOCOR dan akan terus memantau hingga kondisi aman atau IDLE.

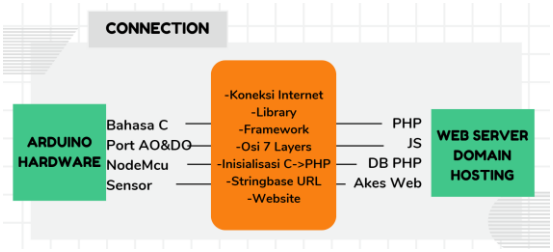
Transisi antar status:

1. Dari IDLE ke AMAN: Jika sensor mendeteksi kondisi aman (0 hingga 500 PPM).
2. Dari AMAN ke GAS BOCOR: Jika sensor mendeteksi rentang 750 PPM hingga 1000 PPM.
3. Dari GAS BOCOR ke IDLE: Jika sensor mendeteksi kondisi aman (tidak ada PPM atau sistem reset ke 0 PPM).

Diagram ini digunakan untuk menggambarkan alur kerja sistem deteksi gas dan respon yang dilakukan saat mendeteksi kondisi tertentu, seperti kondisi aman atau adanya kebocoran gas.

2.7 Connection

Rancangan desain UI ini akan menggambarkan desain rancangan yang akan dipakai untuk memuat tampilan ketika kotak penyimpanan (*MoneyBox Plus*) berhasil melakukan *scan* pada warna yang ada pada uang pecahan uang Rp.100.000,00- dan Rp.50.000,00-



Gambar 5. *Connection*

Diagram pada Gambar 5 memperlihatkan struktur koneksi dari sistem pendeteksi dini kebocoran gas berbasis *Internet of Things* (IoT) yang terhubung ke *server web*. Diagram ini menggambarkan tiga komponen utama yang saling terhubung: perangkat keras Arduino, koneksi, dan *hosting domain web server*. Pada bagian perangkat keras Arduino, pemrograman dilakukan menggunakan bahasa C. Arduino memanfaatkan port Analog Output (AO) dan *Digital Output* (DO) untuk berkomunikasi dengan sensor, sementara modul WiFi NodeMcu digunakan untuk menghubungkan Arduino ke internet, dengan sensor berfungsi mendeteksi keberadaan gas.

Bagian koneksi mencakup beberapa elemen penting. Arduino terhubung ke internet melalui NodeMcu dan menggunakan pustaka (*library*) untuk mempermudah pemrograman dan konektivitas. *Framework* digunakan untuk pengembangan *website IoT*, dan model referensi tujuh lapisan *OSI* digunakan sebagai standar komunikasi jaringan. Proses inisialisasi data dari bahasa C ke PHP dilakukan untuk pengiriman data, dan *URL* digunakan untuk mengirim data dari Arduino ke *server web*. Data yang dikirim oleh sensor kemudian ditampilkan di situs web melalui antarmuka pengguna.

Pada bagian hosting domain *web server*, PHP digunakan sebagai bahasa pemrograman di sisi server untuk menangani data yang diterima. *JavaScript* (JS) digunakan untuk interaktivitas di web, dan basis data (DB PHP) digunakan untuk menyimpan data dari sensor. Proses akses data dilakukan melalui web, memungkinkan pemantauan kebocoran gas secara *real-time*.

Diagram ini menggambarkan cara kerja sistem deteksi kebocoran gas mulai dari pengumpulan data oleh sensor, pengolahan dan pengiriman data oleh Arduino melalui internet, hingga penyimpanan dan tampilan data di server web untuk pemantauan lebih lanjut. Sistem pendeteksi dini kebocoran gas berbasis IoT ini bekerja dengan mengumpulkan data dari sensor gas yang terhubung ke Arduino. Sensor gas mendeteksi kebocoran dan mengirimkan data melalui *port* AO dan DO ke Arduino. Arduino kemudian memproses data tersebut menggunakan program yang ditulis dalam bahasa C. Data yang telah diproses dikirimkan ke *NodeMcu*, modul WiFi yang menghubungkan Arduino ke internet. *NodeMcu* mengirim data tersebut ke server web melalui URL yang telah ditentukan.

Server web menerima data melalui *Source Code* PHP yang memproses dan menyimpannya ke dalam basis data (DB PHP). Data yang tersimpan di database kemudian ditampilkan di situs web, di mana *JavaScript* digunakan untuk memperbarui tampilan web secara dinamis dan menampilkan data secara *real-time* kepada pengguna. Setiap data yang diterima dari sensor disimpan dalam *database*, memungkinkan pengguna untuk melihat riwayat data kebocoran gas melalui antarmuka web. Riwayat ini ditampilkan dalam halaman *history*, memungkinkan analisis lebih lanjut atas data yang telah dikumpulkan. Dengan cara ini, sistem ini memungkinkan pemantauan kebocoran gas secara *real-time* dan memberikan akses ke riwayat data melalui *server web* untuk analisis lebih lanjut.

III. HASIL DAN PEMBAHASAN

3.1 Pengujian Sistem

Pengujian sistem ini dilakukan berdasarkan desain yang telah dirancang sebelumnya, bertujuan untuk mengevaluasi apakah sistem yang telah dibuat dapat berfungsi sesuai dengan harapan. Berikut adalah hasil uji coba sistem yang dikembangkan sesuai dengan desain yang telah dirancang.

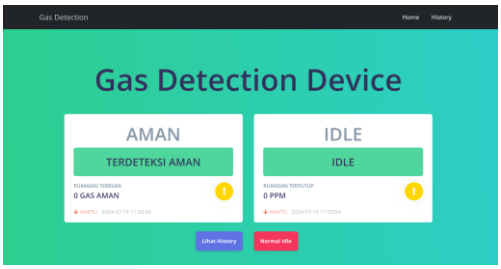
A. Pengujian Menu Utama

Uji coba pada menu utama ini berfungsi sebagai tampilan awal yang digunakan sebagai jembatan untuk menampilkan output hasil inisialisasi sensor dari alat *G2D (Gas Detection Device)* di menu utama seperti pada Gambar 6.

Langkah-langkah penerapan cara kerja deteksi kebocoran gas pada ruangan tertutup dan ruangan terbuka (*Gas Detection Device*) secara sistematis. (1) Langkah pertama adalah menghubungkan alat *G2D(Gas Detection Device)* ke *Power Supply*. Alat *Power Supply* ini menggunakan kabel data USB untuk memberikan daya. (2) Setelah *G2D(Gas Detection*

pISSN: 2442-3386 eISSN: 2442-4293
Vol 11 No 2 Jul – Des 2025, 207 - 214
Melek IT
INFORMATION TECHNOLOGY JOURNAL

Device) terhubung dengan *Power Supply*, mikrokontroler ESP8266 sebagai pusat pengendali akan mulai beroperasi untuk menjalankan sensor pada alat *G2D(Gas Detection Device).* (3) Kemudian apabila ada kebocoran pada ruangan tertutup atau ruangan terbuka maka sensor akan menginisialisasi dan mengirimkan output pada *Website.* (4) Tekan *button History* yang terletak dibawah nilai kebocoran gas, maka akan pindah ke halaman *History* untuk melihat Riwayat kebocoran yang menampilkan tanggal, waktu, status dan ruangan kebocoran gas. Apabila ingin membersihkan halaman Riwayat Kebocoran Gas klik *button Reset History.*



Gambar 6. Menu Utama

Dengan mengikuti Langkah-langkah ini, user dapat secara sistematis mengoperasikan *G2D(Gas Detection Device)* dengan menggunakan sensor MQ2 untuk mendeteksi adanya kebocoran gas secara efektif dan terkontrol.

B. Pengujian Gas Detection Device (G2D)

Uji coba pada *History* ini merupakan akses yang akan digunakan untuk mengamati dan mencatat status dari *Gas Detection Device (G2D)* berdasarkan pengukuran sensor gas yang ditunjukkan pada Tabel 1.

Tabel 1. Pengujian G2D

Tanggal	Waktu	Status	Ruangan
2024-06-01	17:33:26	AMAN	Terbuka
2024-06-01	17:33:29	IDLE	Tertutup
2024-06-01	17:33:29	AMAN	Terbuka
2024-06-01	17:33:32	GAS BOCOR	Tertutup
2024-06-01	17:33:32	AMAN	Terbuka
2024-06-01	17:33:35	GAS BOCOR	Tertutup
2024-06-01	17:33:35	AMAN	Terbuka
2024-06-01	17:33:38	GAS BOCOR	Tertutup
2024-06-01	17:34:03	AMAN	Terbuka
2024-06-01	17:34:06	IDLE	Tertutup
2024-06-01	17:35:22	AMAN	Terbuka
2024-06-01	17:35:22	IDLE	Tertutup
2024-06-01	17:36:40	AMAN	Terbuka
2024-06-01	17:36:43	GAS BOCOR	Tertutup
2024-06-01	17:36:43	AMAN	Terbuka
2024-06-01	17:36:46	GAS BOCOR	Tertutup
2024-06-01	17:36:46	AMAN	Terbuka

3.2 Hasil Penelitian

Berdasarkan Berdasarkan pengujian sistem yang dilakukan saat penelitian terhadap sistem pendeteksi dini kebocoran gas G2D (*Gas Detection Device*) memperoleh hasil sebagai berikut:

A. Layout Kode Smoke Detection System with ESP8266

```
void loop() {
  if ((WiFiMulti.run() == WL_CONNECTED)) {
    int analogSensor1 = analogRead(smokeA0);
    int digitalSensor2 = digitalRead(smokeD0);
    int apiValue = digitalRead(sensorPin);

    Serial.print("Smoke Sensor 1 Analog Value: ");
    Serial.println(analogSensor1);

    Serial.print("Smoke Sensor 2 Digital Value: ");
    Serial.println(digitalSensor2);

    if (apiValue == 1 || digitalSensor2 == 1) {
      digitalWrite(buzzer, HIGH);
    } else {
      digitalWrite(buzzer, LOW);
    }

    if (analogSensor1 > 1000) {
      digitalWrite(buzzer, HIGH);
    } else {
      digitalWrite(buzzer, LOW);
    }
```

```
    payload = "asap1=" + String(digitalSensor2) + "&asap2=" + String(analogSensor1);
    USE_SERIAL.print("Sending payload: ");
    USE_SERIAL.println(payload);

    USE_SERIAL.print("[HTTP] Memulai...\n");
    http.begin(wifiClient, url + "?" + payload);
    int httpResponseCode = http.GET();  // Ubah ke GET karena URL dengan parameter

    if (httpResponseCode > 0) {
      USE_SERIAL.printf("[HTTP] Kode response: %d\n", httpResponseCode);
      USE_SERIAL.println(http.getString());
    } else {
      USE_SERIAL.printf("[HTTP] Post gagal, error: %s\n", http.errorToString(httpResponseCode).c_str());
    }

    http.end();
  }
  delay(3000);
}
```

Gambar 7. Layout kode *Smoke Detection*

Pada Gambar 7 adalah kode sebuah fungsi dalam program Arduino yang disebut "*Smoke Detection System with ESP8266*". Fungsi ini untuk mengkoneksikan inisialisasi sensor Gas MQ2 ke *Website* pada menu utama. Berikut penjelasan mengenai setiap baris kode dalam fungsi "*Smoke Detection System with ESP8266*":

1. *if ((WiFiMulti.run() == WL_CONNECTED))* Memeriksa apakah terhubung ke WiFi. Jika terhubung, maka lanjutkan menjalankan kode di dalam blok ini.
2. *int analogSensor1 = analogRead(smokeA0);* Membaca nilai analog dari sensor asap 1 (pin A0) dan menyimpannya ke variabel analogSensor1.
3. *int digitalSensor2 = digitalRead(smokeD0);* Membaca nilai digital dari sensor asap 2 (pin D0) dan menyimpannya ke variabel digitalSensor2.
4. *int apiValue = digitalRead(sensorPin);*Membaca nilai digital dari pin sensor tambahan (sensorPin) dan menyimpannya ke *variabel apiValue*.
5. *Serial.print("Smoke Sensor 1 Analog Value: ");* Menampilkan teks "*Smoke Sensor 1 Analog Value:* " di *Serial Monitor*.
6. *Serial.println(analogSensor1);* Menampilkan nilai analog dari sensor asap 1 di *Serial Monitor*.

7. *Serial.print("Smoke Sensor 2 Digital Value: ");* Menampilkan teks "*Smoke Sensor 2 Digital Value:* " di *Serial Monitor*.
8. *Serial.println(digitalSensor2);* Menampilkan nilai digital dari sensor asap 2 di *Serial Monitor*.
9. *if (apiValue == 1 || digitalSensor2 == 1)* Jika sensor api *(apiValue)* atau sensor asap digital (digitalSensor2) mendeteksi asap atau api (nilai 1), maka:
10. *digitalWrite(buzzer, HIGH);* Menyalakan *buzzer* untuk memberikan peringatan.
11. *} else {* // 11. Jika tidak, maka:
12. *digitalWrite(buzzer, LOW);* // Mematikan buzzer.
13. *if (analogSensor1 > 1000)* { // Jika nilai analog dari sensor asap 1 lebih dari 1000 (asap terdeteksi):
14. *digitalWrite(buzzer, HIGH);* // Menyalakan buzzer untuk memberikan peringatan.
15. *} else {* // 15. Jika tidak, maka:
16. *digitalWrite(buzzer, LOW);* // Mematikan buzzer.
17. *payload = "asap1=" + String(digitalSensor2) + "&asap2=" + String(analogSensor1);* //Membuat payload dari nilai sensor asap untuk dikirim ke *server*.
18. *USE_SERIAL.print("Sending payload: ");* // Menampilkan teks "*Sending payload*: " di Serial Monitor.
19. *USE_SERIAL.println(payload);*//Menampilkan *payload* yang akan dikirim di *Serial Monitor*.
20. *USE_SERIAL.print("[HTTP]Memulai...\n");* Menampilkan teks "[HTTP] Memulai..." di *Serial Monitor*.
21. *http.begin(wifiClient, url + "?" + payload);* //Membuat koneksi HTTP dengan URL dan menambahkan *payload* sebagai parameter.
22. *int httpResponseCode = http.GET();* //Mengirim HTTP *GET request* dan mendapatkan kode *respons* dari *server*.
23. *if (httpResponseCode > 0) {* // Jika HTTP request berhasil (kode respons lebih dari 0), maka:
24. *USE_SERIAL.printf("[HTTP] Kode response: %d\n", httpResponseCode);* // Menampilkan kode respons di Serial Monitor.
25. *USE_SERIAL.println(http.getString());* //Menampilkan respons dari server di Serial Monitor.
26. *} else {* // Jika HTTP request gagal, maka:
27. *USE_SERIAL.printf("[HTTP] Post gagal, error: %s\n", http.errorToString(httpResponseCode).c_str());*//Menampilkan pesan error di Serial Monitor.
28. *http.end();*//Mengakhiri HTTP request untuk membebaskan sumber daya.
29. *delay(3000);*//Menunggu selama 3 detik sebelum mengulangi *loop*, memberikan jeda waktu sebelum membaca sensor dan mengirim data lagi.

B. Layout Kode koneksi.php

koneksi.php > ...

```php
<?php
$db_host = "localhost:3306";//:3306
$db_user = "root";
$db_pass = "";
$db_name = "asap";

$conn = mysqli_connect($db_host, $db_user, $db_pass, $db_name);

if(mysqli_connect_errno()){
    echo 'Gagal melakukan koneksi ke Database : '.mysqli_connect_error();
}else{
//  echo "berhasil";
}
?>
```

Gambar 8. Layout Kode Koneksi.php

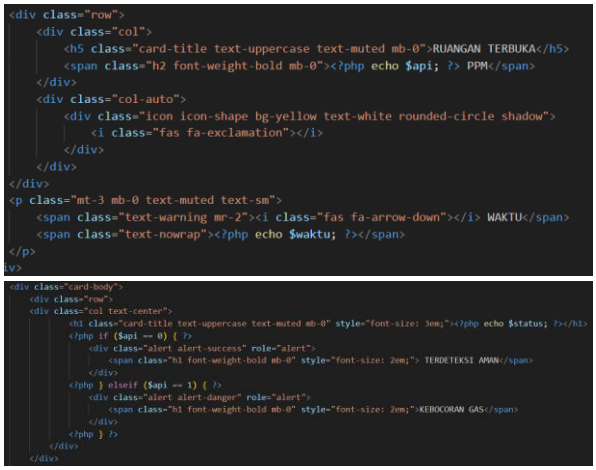
Pada Gambar 8 kode program untuk mengatur status tertentu dalam tabel api pada *database* menjadi nilai default ketika diakses melalui permintaan *POST*. Kode ini berguna dalam skenario di mana ada kebutuhan untuk mengatur ulang atau mengubah status sistem menjadi "*IDLE*" secara otomatis. Berikut penjelasan mengenai setiap baris kode *Set_Idle.php*:

1. *<?php* : Membuka tag PHP untuk menulis kode PHP.
2. *include "koneksi.php";* : Menyertakan file "koneksi.php" yang berisi skrip untuk menghubungkan ke *database*.
3. *date_default_timezone_set('Asia/Jakarta');*:Mengatur zona waktu default menjadi 'Asia/Jakarta'.
4. *if ($_SERVER['REQUEST_METHOD'] === 'POST') {* : Memeriksa apakah metode permintaan HTTP adalah POST.
5. *$q = "UPDATE api SET asap1 = '0', asap2 = '0', status1 = 'AMAN', status2 = 'IDLE', waktu = CURRENT_TIMESTAMP WHERE id = '35';";* :Menyimpan query SQL dalam variabel $q untuk mengatur kolom asap1, asap2, status1, status2, dan waktu dalam tabel api menjadi nilai tertentu, di mana id adalah '35'.
6. *$ck = mysqli_query($conn, $q);* : Menjalankan query SQL yang disimpan dalam $q menggunakan koneksi database $conn dan menyimpan hasilnya dalam $ck.
7. *if ($ck) {* : Memeriksa apakah query berhasil dijalankan.
8. *echo "Status berhasil diatur ke IDLE.";* : Menampilkan pesan "Status berhasil diatur ke IDLE." jika query berhasil dijalankan.
9. *} else {* : Jika query gagal dijalankan, lakukan blok kode ini.

pISSN: 2442-3386 eISSN: 2442-4293
Vol 11 No 2 Jul – Des 2025, 209 - 214
Melek IT
INFORMATION TECHNOLOGY JOURNAL

10. *echo "Gagal mengatur status ke IDLE.";* : Menampilkan pesan "Gagal mengatur status ke IDLE." jika query gagal dijalankan.
11. *}* : Menutup blok if yang memeriksa keberhasilan query.
12. *} else {* : Jika metode permintaan HTTP bukan POST, lakukan blok kode ini.
13. *echo "Permintaan tidak valid.";* : Menampilkan pesan "Permintaan tidak valid." jika metode permintaan bukan POST.
14. *}* : Menutup blok if-else yang memeriksa metode permintaan HTTP.

C. Layout Kode Ruangan



Gambar 9. Layout Kode Ruangan

Pada Gambar 9 adalah kode program sebuah fungsi dalam program *Index.php* yang disebut ruangan terbuka. Fungsi ini untuk mengetahui keadan ruangan terbuka dan jumlah PPM gas di ruangan. Berikut penjelasan mengenai setiap baris kode ruangan terbuka:

1. * * Membuat div container dengan class "row" untuk membuat baris dalam layout grid.
2. * * Membuat div container dengan class "col" untuk membuat kolom dalam baris.
3. *<h5 class="card-title text-uppercase text-muted mb-0">RUANGAN TERBUKA</h5>* Menampilkan judul "RUANGAN TERBUKA" dengan gaya h5, huruf kapital, teks berwarna muted (abu-abu), dan margin bawah 0.
4. *<?php echo $api; ?> PPM* Menampilkan nilai dari variabel PHP $api diikuti dengan "PPM" dalam elemen span, dengan gaya h2, teks tebal, dan margin bawah 0.
5. Menutup div dengan class "col".
6. * * Membuat div container dengan class "col-auto" untuk membuat kolom dengan lebar otomatis sesuai konten.
7. * * Membuat div container dengan class "icon", "icon-shape", latar belakang kuning, teks berwarna putih, bentuk lingkaran, dan bayangan.
8. ** Menampilkan ikon font awesome dengan kelas "*fa-exclamation"*.
9. * * Menutup div dengan class *"icon icon-shape bg-yellow text-white rounded-circle shadow"*.
10. * * Menutup div dengan *class "col-auto"*.
11. * * Menutup div dengan *class "row"*.
12. * * Membuat paragraf dengan margin atas 3, margin bawah 0, teks berwarna muted, dan ukuran teks kecil.
13. * WAKTU* Menampilkan teks "WAKTU" dengan ikon panah bawah, teks berwarna kuning, dan margin kanan 2.
14. *<?php echo $waktu; ?>* Menampilkan nilai dari variabel PHP $waktu dalam elemen span dengan teks tidak terpotong (nowrap).
15. Menutup paragraf dengan *class "mt-3 mb-0 text-muted text-sm"*.
16. Menutup div.
17. Menutup div.
18. Menutup div.
19. * * Membuat *div container* dengan *class "col-xl-4 col-lg-8"* untuk membuat kolom dengan lebar tertentu pada layar xl dan lg.
20. * * Membuat div container *dengan class "card card-stats mb-4 mb-xl-0"* untuk membuat kartu dengan margin bawah 4 dan margin bawah 0 pada layar xl.
21. * * Membuat *div container* dengan *class* "*card-body*" untuk isi dari kartu.
22. * * Membuat div container dengan *class "row"* untuk membuat baris dalam *layout grid.*

23. * *
    Membuat div container dengan *class "col text-center"* untuk membuat kolom dengan teks rata tengah.
24. *<h1 class="card-title text-uppercase text-muted mb-0" style="font-size: 3em;"><?php echo $status2; ?></h1>*
    Menampilkan nilai dari variabel PHP $status2 dalam elemen h1 dengan gaya judul kartu, huruf kapital, teks berwarna muted, margin bawah 0, dan ukuran font 3em.
25. *<?php if ($asap <= 750) { ?>*
    Membuka blok kode PHP dengan kondisi jika nilai $asap kurang dari atau sama dengan 750.
26. * *
    Membuat div dengan class "alert alert-success" untuk menampilkan pesan sukses dengan peran alert.
27. *IDLE*
    Menampilkan teks "IDLE" dalam elemen span dengan gaya h1, teks tebal, margin bawah 0, dan ukuran font 2em.
28. * * Menutup div dengan class "alert alert-success".
29. *<?php } elseif ($asap > 750 && $asap <= 900) { ?>*
    *M*embuka blok kode PHP dengan kondisi jika nilai $asap lebih besar dari 750 dan kurang dari atau sama dengan 900.
30. * *
    Membuat div dengan *class "alert alert-success"* untuk menampilkan pesan sukses dengan peran alert.
31. * TERDETEKSI AMAN*
    Menampilkan teks "TERDETEKSI AMAN" dalam elemen span dengan gaya h1, teks tebal, margin bawah 0, dan ukuran font 2em.
32. * *
    Menutup div dengan *class "alert alert-success"*.
33. *<?php } elseif ($asap > 900) { ?>*
    Membuka blok kode PHP dengan kondisi jika nilai $asap lebih besar dari 900.
34. * *
    Membuat div dengan *class "alert alert-danger"* untuk menampilkan pesan bahaya dengan peran alert.
35. *KEBOCORAN GAS*
    Menampilkan teks "KEBOCORAN GAS" dalam elemen span dengan gaya h1, teks tebal, margin bawah 0, dan ukuran font 2em.
36. * *
    Menutup div dengan class "alert alert-danger".
37. *<?php } elseif ($asap < 500) { ?>* Membuka blok kode PHP dengan kondisi jika nilai $asap kurang dari 500.
38. * * Membuat div dengan *class "alert alert-success"* untuk menampilkan pesan sukses dengan peran *alert.*
39. *- - -* Menampilkan teks "- - -" dalam elemen span dengan gaya h1, teks tebal, margin bawah 0, dan ukuran font 2em.
40. * *
    Menutup div dengan class "alert alert-success".
41. *<?php } ?>* Menutup blok kode PHP.
42. * * Menutup div dengan *class "col text-center"*.
43. * * Menutup div dengan *class "row"*.
44. * * Menutup div dengan *class "card-body"*.
45. * * Menutup div dengan *class "card card-stats mb-4 mb-xl-*0".
46. * * Menutup div dengan *class "col-xl-4 col-lg-8"*.

3.3 Hasil Uji Coba

Uji coba sistem pendeteksi kebocoran gas G2D dilakukan dengan mensimulasikan berbagai skenario kebocoran gas untuk menguji kinerja sensor MQ2 dan kehandalan komunikasi melalui IoT dan Web Server. Berikut adalah hasil dari beberapa skenario uji coba:

A. Skenario 1 : Kebocoran Gas di Ruangan Tertutup

1. Kondisi: Kebocoran gas di ruangan tertutup dengan konsentrasi gas meningkat secara bertahap.
2. Hasil: Sensor MQ2 mendeteksi kenaikan konsentrasi gas dan mengirimkan data ke NodeMCU.
3. Respons: Data dikirim secara *real-time* ke *server* dan pemberitahuan kebocoran gas muncul di *Web Server* dalam waktu kurang dari 3 detik.
4. Status: GAS BOCOR.

B. Skenario 2 : Kebocoran Gas di Ruangan Terbuka

1. Kondisi: Kebocoran gas di ruangan terbuka dengan konsentrasi gas menyebar lebih cepat.
2. Hasil: Sensor MQ2 mendeteksi adanya gas meskipun konsentrasi lebih rendah dibandingkan ruangan tertutup.

pISSN: 2442-3386 eISSN: 2442-4293
Vol 11 No 2 Jul – Des 2025, 211 - 214
Melek IT
INFORMATION TECHNOLOGY JOURNAL

3. Respons: Data dikirim ke server dan pemberitahuan muncul di *Web Server* dalam waktu kurang dari 3 detik.
4. Status: GAS BOCOR.

IV. PENUTUP

4.1. Kesimpulan

Berdasarkan hasil penelitian dan analisis terhadap sistem pendeteksi kebocoran gas G2D (Gas Detection Device) berbasis mikrokontroler NodeMCU, berikut adalah kesimpulan yang menjawab rumusan masalah:

1. Cara Kerja Sensor M2Q dalam Mendeteksi Kebocoran Gas: Sensor M2Q berfungsi dengan mendeteksi perubahan konsentrasi gas di sekitarnya. Ketika konsentrasi gas melebihi ambang batas yang telah ditentukan, sensor ini menghasilkan sinyal listrik yang dikirim ke mikrokontroler NodeMCU. NodeMCU kemudian memproses sinyal tersebut dan memutuskan apakah ada kebocoran gas yang perlu dilaporkan. Sensor M2Q menunjukkan keandalan dan sensitivitas yang tinggi dalam mendeteksi keberadaan gas berbahaya, sehingga cocok digunakan dalam sistem G2D.
2. Penggunaan Mikrokontroler NodeMCU sebagai Pusat Pengolahan Data: NodeMCU berperan sebagai pusat pengolahan data dalam sistem G2D. Data yang diterima dari sensor M2Q diproses secara real-time oleh NodeMCU. Penggunaan NodeMCU sebagai mikrokontroler meningkatkan efisiensi sistem karena kemampuannya dalam melakukan pemrosesan data dengan cepat dan mengirimkan notifikasi secara instan saat terdeteksi kebocoran gas. Selain itu, NodeMCU memungkinkan integrasi yang mudah dengan modul Wi-Fi, yang memastikan sistem dapat terhubung ke jaringan internet untuk pemantauan jarak jauh.
3. Cara Kerja Koneksi Sensor M2Q ke Web Server: NodeMCU menggunakan modul ESP8266 untuk menghubungkan sensor M2Q ke web server. Data yang dikumpulkan oleh sensor M2Q dikirim melalui NodeMCU menggunakan koneksi Wi-Fi. NodeMCU menginisialisasi dan mengirim data ke web server melalui URL yang telah ditentukan menggunakan protokol HTTP. Di web server, data ini diproses oleh Source Code PHP dan disimpan dalam basis data. Informasi ini kemudian ditampilkan di website dalam bentuk yang mudah dipahami oleh pengguna, memungkinkan pemantauan kondisi gas secara real-time dan akses riwayat data.

Kesimpulan di atas menunjukkan bahwa sistem pendeteksi kebocoran gas G2D dengan menggunakan sensor M2Q dan mikrokontroler NodeMCU tidak hanya efektif dalam mendeteksi kebocoran gas, tetapi juga efisien dalam pemrosesan data dan respons sistem, serta memungkinkan pemantauan jarak jauh yang handal melalui koneksi web server.

4.2. Saran

Dari hasil penelitian sistem pendeteksi kebocoran gas G2D, saran untuk pengembangan alat ini adalah sebagai berikut :

1. Tambahkan Fitur Baru: Tingkatkan fungsi dan kenyamanan G2D dengan fitur alarm suara atau visual yang lebih canggih dan otomatisasi untuk menutup aliran gas saat terdeteksi kebocoran.
2. Tingkatkan Akurasi dan Kecepatan: Optimalkan algoritma deteksi dan pemrosesan data sensor untuk meningkatkan akurasi dan respons waktu. Pastikan sensor dikalibrasi secara berkala.
3. Buat Aplikasi Mobile: Kembangkan aplikasi mobile yang terhubung dengan G2D untuk memudahkan pengguna memantau kondisi gas dan menerima notifikasi di smartphone.
4. Skalabilitas: Rancang G2D agar bisa menambah modul atau sensor tambahan sesuai kebutuhan, seperti mendeteksi jenis gas lainnya atau area yang lebih luas. Integrasikan dengan sistem keamanan rumah pintar.
5. Kembangkan Model Bisnis: Jadikan G2D produk komersial dengan strategi pemasaran yang baik, analisis pasar, dan kerjasama dengan perusahaan gas atau layanan keamanan. Tawarkan layanan pemantauan dan pemeliharaan berkala sebagai nilai tambah.

Saran tersebut diharapkan dapat membantu dalam mengembangkan G2D menjadi lebih baik, lebih inovatif, dan dapat memberikan manfaat yang lebih besar bagi pengguna. dan dapat memberikan manfaat yang lebih besar bagi pengguna.

DAFTAR PUSTAKA

[1] Purnomo, “Model Prototyping Pada Pengembangan Sistem Informasi,” *Jurnal Informatika Merdeka pasuruan,* p. 55, 2017.

[2] l. awaludin, “Apakah Itu Sensor ?,” 22 Agustus 2018. [Online]. Available:

https://sensornetwork.mipa.ugm.ac.id/2018/08/22/153/.

[3] M. Riadi, "Warna (Definisi, Unsur, Jenis dan Psikologi)," 9 oktober 2020. [Online]. Available: https://www.kajianpustaka.com/2020/10/warna-definisi-unsur-jenis-dan-psikologi.html.

[4] Aguskhumaidi, "Mikrokontroler Arduino," 5 september 2019. [Online]. Available: https://lecturer.ppns.ac.id/aguskhumaidi/2019/09/05/mikrokontroler-arduino/.

[5] M. D. Utami, "PERANCANGAN DAN ANALISA KINERJA SISTEM AKUISISI DATA," *TRANSIENT, VOL. 9, NO. 3,* p. 361, 2020.

[6] M. A. Ikhsan, M. Yahya dan F. A. Fiolana, "Pendeteksi Kekeruhan Air Di Tandon Rumah Berbasis Arduino Uno," *Jurnal Qua Teknika, Vol. 8 No. 2,* pp. 17-29, 2018.

[7] B. Gramedia, "Pengertian Uang: Fungsi, Ragam, dan Teori Nilai Uang," 21 Agustuts 2021. [Online]. Available: https://www.gramedia.com/literasi/uang/#Pengertian_Uang_-_Apa_itu_Uang.

[8] F. N. H. Chairul Gunawan, "Prototipe Light Meter Fotografi Studio Menggunakan Mikrokontroler," *JURNAL MEDIA INFORMATIKA BUDIDARMA,* pp. 769-778, 2021.

[9] I. M. R. A. K. I. N. S. D. Y. Anantajaya, "REVIEW APLIKASI SENSOR PADA SISTEM," *Jurnal SPEKTRUM,* p. vol 8, 2021.

[10] M. Z. Z. Erfian Junianto, "Penerapan Metode Palette untuk Menentukan Warna," *JURNAL INFORMATIKA,* p. vol 5 no 1, 2018.

[11] R. I. A. G. Gansar Suwanto, "DETEKSI KEASLIAN UANG KERTAS BERDASARKAN," *JIP (Jurnal Informatika Polinema),* p. vol 7 no 2, 2021.

[12] ditempel, "Menggunakan Modul LCD 16 x 2," 19 maret 2021. [Online]. Available: https://www.ditempel.com/2021/03/menggunakan-modul-lcd-16-x-2.html.

[13] R. Setiawan, "Flowchart Adalah: Fungsi, Jenis, Simbol, dan Contohnya," 4 Agustus 2021. [Online]. Available: https://www.dicoding.com/blog/flowchart-adalah/.

[14] M. F. Fadhil, "Diagram alir," 3 maret 2022. [Online]. Available: https://www.diklatkerja.com/blog/diagram-alir.

[15] J. Cohen, "Color Ontology and Color Science," 22 January 2010. [Online]. Available: www.researchgate.net.

[16] J. W. Simatupang, B. Prasetyo, M. Galina dan A. Suhartomo, "Prototipe Mesin Penjual Air Mineral Otomatis berbasi Arduino Mega 2560 dan RC522," *ELEKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika Vol. 10 No.2,* pp. 484-499, 2022.

[17] T. Widiyaman, "Pengertian Modul Wifi ESP8266," 1 april 2023. [Online]. Available: https://www.warriornux.com/pengertian-modul-wifi-esp8266/.

[18] A. elektro, "Cara Kerja Dan Karakteristik Sensor Gas MQ-2," 29 september 2018. [Online]. Available: https://www.andalanelektro.id/2018/09/cara-kerja-dan-karakteristik-sensor-gas-mq2.html?m=1.

[19] E. A. Prastyo, "Software Arduino IDE," 18 07 2018. [Online]. Available: https://www.arduinoindonesia.id/2018/07/software-arduino-ide.html.

[20] R. Setiawan, "Memahami Apa Itu Internet of Things," 8 September 2021. [Online]. Available: https://www.dicoding.com/blog/apa-itu-internet-of-things/.

[21] R. Setiawan, "Prototype," 11 Agustus 2021. [Online]. Available: https://www.dicoding.com/blog/apa-itu-prototype-kenapa-itu-penting/.

[22] D. Purnomo, "Model Prototyping Pada Pengembangan," *Jurnal Informatika Merdeka Pasuruan,* p. 55, 2017.

[23] T. Suryana, "Implementasi Modul Sensor MQ2 Untuk," *Jurnal Komputa Unikom 2021,* p. 3, 2021.

[24] E. A. Prastyo, "Software Arduino IDE," 10 januari 2013. [Online]. Available: https://www.arduinoindonesia.id/2018/07/software-arduino-ide.html.

[25] R. Setiawan, "Flowchart," 4 agustus 2021. [Online]. Available: https://www.dicoding.com/blog/flowchart-adalah/.

[26] A. N. Hesanty, "Apa Itu Flowchart: Pengertian dan Penerapannya di Berbagai Bidang," 19 mei 2023. [Online]. Available: https://www.niagahoster.co.id/blog/flowchart-adalah/.

pISSN: 2442-3386 eISSN: 2442-4293
Vol 11 No 2 Jul – Des 2025, 213 - 214
Melek IT
INFORMATION TECHNOLOGY JOURNAL

[27] Mushthofa, "Diagram alir," 11 juni 2023. [Online]. Available: https://id.wikipedia.org/wiki/Diagram_alir.

[28] kamaln, "Contoh Benda Gas: Pengertian, Sifat, dan Proses Perubahannya," 2021. [Online]. Available: https://www.gramedia.com/literasi/contoh-benda-gas/.

[29] d. n. p. s. suherman, "PERANCANGAN ALAT PENDETEKSI KEBOCORAN GAS MENGGUNAKAN," *Jurnalllmiah Skylandsea* , p. 4, 2019.

[30] A. Razor, "Buzzer Arduino : Pengertian, Cara Kerja, dan Contoh Program," 1 maret 2022. [Online]. Available: https://www.aldyrazor.com/2020/05/buzzer-arduino.html?m=1.

[31] salsy, "NodMcu BAse," 13 07 2023. [Online]. Available: https://www.cytron.io/p-base-board-for-nodemcu-v3.

[32] Cloudflare, "What is the OSI Model?," 2024. [Online]. Available: https://www.cloudflare.com/learning/ddos/glossary/open-systems-interconnection-model-osi/.

[33] A. W. Services, "Apa Itu Model OSI?," 2023. [Online]. Available: https://aws.amazon.com/id/what-is/osi-model/.

[34] A. K. Lab, "What is an IP Address – Definition and Explanation," 2 januari 2024. [Online]. Available: https://www.kaspersky.com/resource-center/definitions/what-is-an-ip-address.

[35] s. m. kerner, "DEFINITION Internet Protocol (ip)," 02 januari 2024. [Online]. Available: https://www.techtarget.com/searchunifiedcommunications/definition/Internet-Protocol.

[36] N. V. Siringoringo, "Pengertian State Diagram," 03 februari 2020. [Online]. Available: https://medium.com/state-diagram/pengertian-state-diagram-67d3a4069da3.

[Halaman ini dibiarkan kosong]