

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian oleh Alfie Syahri dan Ramadhani Ulansari (2022) yang berjudul *Prototype Alat Pendekksi Kebocoran Gas dan Api dengan Menggunakan Sensor MQ2 dan Sensor Api Berbasis Internet of Things* menjelaskan bahwa alat pendekksi ini memanfaatkan sensor MQ-2 yang terhubung dengan mikrokontroler *ESP8266 NodeMCU BaseBoard Expansion Board*. Sensor ini mampu mendekksi zat seperti alkohol, H₂, *LPG*, CH₄, CO, asap, dan propana. Data hasil pembacaan sensor ditampilkan melalui *LCD*. Komponen utama lain yang digunakan dalam sistem ini meliputi papan *Arduino Uno*, mikrokontroler *ESP8266 NodeMCU*, buzzer, kipas DC, dan *LCD*. Teknik pengumpulan data dilakukan melalui referensi seperti jurnal, *datasheet*, artikel, dan buku. Proses pembangunan alat ini dibagi menjadi dua bagian, yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*). Hasil penelitian menunjukkan bahwa pengguna dapat mengetahui kadar *LPG* yang bocor secara *real-time*, serta mendapatkan notifikasi melalui *LCD* dan *smartphone* yang terhubung dengan internet. Buzzer dan kipas akan menyala otomatis saat terdeteksi kebocoran gas [4].

Penelitian yang dilakukan oleh Fitryan Nugraha, Zaenudin, dan Al Fath Riza Kholdani (2021) berjudul *Perancangan Sistem Pemadam Api dan Penghisap Asap Otomatis Berbasis Arduino Mega dengan Kendali Android* merancang sistem yang menggunakan sensor *infrared* api sebagai pendekksi

kemunculan api serta sensor MQ-2 untuk mendeteksi asap atau gas. Ketika sistem mendeteksi kebakaran, buzzer dan pompa pemadam otomatis akan aktif. Sistem ini juga dilengkapi lampu *LED* merah, *LCD*, dan notifikasi bahaya yang dikirim melalui aplikasi *Telegram*. Sistem ini efektif dalam memberikan peringatan dini dan telah terbukti berfungsi dengan baik melalui pengujian [5].

Penelitian oleh N. Husin (2022) dalam jurnal berjudul *Rancang Bangun Alat Pendekripsi Kebocoran Gas dan Api Berbasis Arduino Uno dengan MQ-2* mengembangkan sistem deteksi kebakaran berbasis *Arduino* yang mampu mendeteksi indikasi kebakaran lebih dini. Sensor api (*flame detector*) dan MQ-2 akan mengirimkan data saat mendeteksi kebocoran atau percikan api. Sistem ini dapat dioperasikan dari jarak jauh dan memantau kondisi melalui *LCD*. Ketika sensor mendeteksi ancaman, sinyal akan dikirim ke *mikrokontroler* untuk mengaktifkan buzzer, dan *LCD* akan menampilkan pesan “ADA KEBAKARAN” bagi petugas yang sedang mengontrol ruang jaga [6].

dari berbagai penelitian sebelumnya, telah dikembangkan beragam pendekatan dalam mendeteksi kebocoran gas dan api, umumnya dengan memanfaatkan kombinasi sensor MQ-2 dan sensor api serta *mikrokontroler* seperti *Arduino* atau *ESP8266*. Meskipun beberapa sistem telah berbasis *website*, namun integrasi dengan sistem pemadam otomatis masih terbatas. Dalam penelitian ini, akan dikembangkan sistem yang menggabungkan deteksi gas menggunakan sensor MQ-2, deteksi api menggunakan *flame*

sensor, serta sistem pemadam otomatis berbasis APAR portabel yang dikendalikan melalui *website* berbasis *CodeIgniter3*. Sistem ini akan memberikan notifikasi *real-time* melalui antarmuka web dan mengaktifkan pemadam api secara otomatis untuk meningkatkan efektivitas pencegahan kebakaran akibat kebocoran gas *LPG* secara *real-time* dan otomatis.

2.2 Landasan Teori

2.2.1 Sistem Monitoring

Proses deteksi kebocoran gas dan penanganan kebakaran melibatkan prinsip dasar perpindahan massa, di mana perpindahan gas dari sumber kebocoran ke udara sekitar merupakan faktor utama dalam deteksi kebocoran. Gas yang bocor ke udara memiliki perbedaan konsentrasi dengan udara sekitar, yang memicu deteksi oleh alat sensor gas. Proses ini terjadi akibat prinsip hukum perpindahan massa, di mana massa gas berpindah dari area yang lebih terkonsentrasi menuju ke area dengan konsentrasi yang lebih rendah. Pada sistem ini, sensor gas akan mendeteksi perubahan konsentrasi gas yang bocor dan mengirimkan data tersebut ke sistem *monitoring* berbasis *website* untuk diproses lebih lanjut [3].

2.2.2 APAR Portable

Alat Pemadam Api Ringan (APAR) ialah alat pemadam yang dapat dibawa dan dimanfaatkan guna memadamkan api kecil pada tahap awal kebakaran. APAR umumnya berbentuk tabung berisi

bahan pemadam bertekanan tinggi dan tersedia dalam berbagai jenis, seperti APAR bermedia *serbuk kering (powder)*, *busa (foam)*, *karbon dioksida (CO₂)*, *gas Halon*, *air*, dan *liquid gas (AF11)*. APAR digunakan dalam situasi darurat untuk memadamkan api pada menit-menit awal kebakaran, sering dijumpai di *SPBU*, pusat perbelanjaan, atau *mall*, dan ditempatkan di tempat yang mudah terlihat. Cara kerja APAR bervariasi, misalnya APAR bermedia *busa* bekerja dengan mengisolasi dan mendinginkan api, sementara APAR bermedia *CO₂* efektif dalam pemadaman kebakaran *kelas C* (listrik) dengan menurunkan kadar oksigen di sekitar api [7].



Gambar 2. 1 APAR Portable

2.2.3 Gas LPG

LPG (liquefied petroleum gas) adalah gas bumi hasil pencairan dari komponen inti *propana (C₃H₈)* serta *butana (C₄H₁₀)*. Berdasarkan klasifikasinya, *LPG* terbagi ke dalam tiga klasifikasi yaitu *LPG propana*, *LPG butana*, dan *LPG campuran (mix)* dengan kandungan campuran dari keduanya [8].



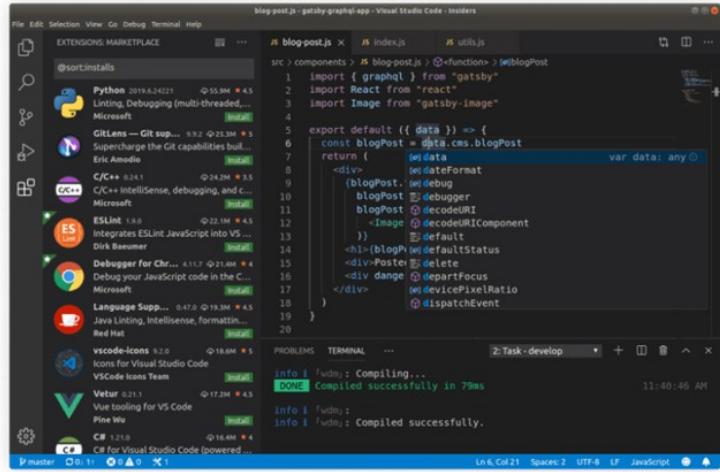
Gambar 2. 2 Gas Lpg

2.2.4 *Website*

Website atau yang juga dikenal dengan istilah *web*, ialah sekumpulan halaman yang menyajikan berbagai informasi seperti teks, data, gambar statis atau dinamis, animasi, suara, dan video yang dapat digabungkan menjadi satu kesatuan yang saling terhubung melalui *hyperlink*. Penggunaan *web service* bertujuan agar klien yang menggunakan beragam *platform*, seperti *desktop*, *website*, atau aplikasi *mobile*, bisa memanfaatkan layanan yang tersedia. Selain itu, tujuan lainnya adalah untuk meminimalkan penggunaan data dan memastikan kecepatan akses yang optimal[10].

2.2.5 *Visual Studio Code*

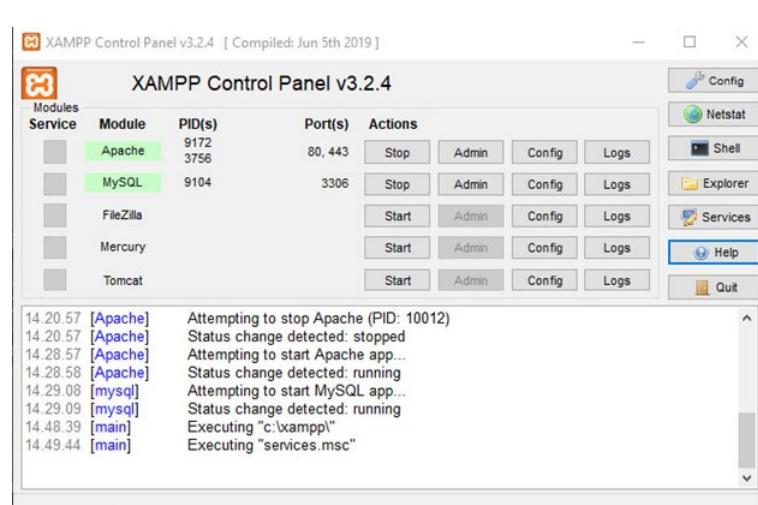
Visual Studio Code ialah aplikasi yang diterbitkan oleh *Microsoft* dengan tujuan sebagai pengembang aplikasi *web* seperti *ASP.NET* dan *Node.js* bagi *web developer*. Teknologi yang relevan seperti *HTML*, *CSS*, *Less*, *Sass*, dan *JSON* juga didukung oleh *Visual Studio Code*. Aplikasi ini mendukung banyak bahasa pemrograman seperti *Batch*, *C++*, *PHP*, *SQL*, *Ruby*, *Razor*, *Visual Basic*, *Java*, *XML*, dan lainnya [9].



Gambar 2. 3 NodeMCU ESP8266

2.2.6 XAMPP

XAMPP ialah perangkat lunak dengan sistem operasi yang beragam. Fungsi *XAMPP* adalah sebagai *server* lokal (*localhost*) yang memiliki berbagai unit seperti *Apache HTTP Server*, *MySQL*, dan penerjemah untuk bahasa *PHP*. Istilah *XAMPP* merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP*, dan *Perl* [8].



Gambar 2. 4 XAMPP

2.2.7 *MySQL*

MySQL (My Structured Query Language) adalah perangkat lunak *open source* yang digunakan sebagai pengelola *database*. *MySQL* merupakan aplikasi *DBMS (Database Management System)* yang terkenal karena gratis, handal, selalu diperbarui, dan banyak digunakan bersama *web server* karena proses instalasinya yang mudah[8].

2.2.8 *Bootstrap*

Dalam pengembangan web agar lebih praktis dan efisien, digunakan *Bootstrap* sebagai *framework front-end* yang bersifat gratis. Di dalamnya terdapat *HTML* dan *CSS* berbasis desain dengan *template* yang bisa digunakan untuk berbagai kebutuhan seperti tipografi, tombol, navigasi, formulir, serta komponen antarmuka lainnya, termasuk juga *ekstensi JavaScript* [3].

2.2.9 *PHP (Personal Home Page)*

PHP atau *Hypertext Preprocessor* adalah bahasa pemrograman yang umum digunakan dalam pembuatan dan pengembangan *website*. Awalnya diciptakan oleh Rasmus Lerdorf pada tahun 1994 dengan fungsi sebagai *Personal Home Page Tools*, yang kemudian berkembang menjadi *Forms Interpreter (FI)*. Sejak versi 3.0, bahasa ini dikenal sebagai *PHP (Hypertext Preprocessor)*[8].

2.2.10 *CodeIgniter*

CodeIgniter adalah *Web Application Framework (WAF)* yang dirancang untuk memudahkan pengembang *web* dalam membuat aplikasi. Di dalamnya terdapat berbagai pustaka kode dan alat bantu yang bisa dikombinasikan dalam satu kerangka kerja (*framework*)[11].



Gambar 2. 5 Codeigniter

2.2.11 UML (*Unified Modeling Language*)

Unified Modeling Language (UML) ialah alat bantu standar yang dipergunakan dalam pengembangan perangkat lunak untuk memvisualisasikan, merancang, dan mendokumentasikan sistem. Jika diibaratkan, seperti seorang arsitek yang membuat denah bangunan agar para pekerja konstruksi dapat membangun dengan presisi, *UML* membantu pengembang memahami struktur dan alur kerja suatu perangkat lunak sebelum proses coding dimulai.

Sebagai metode pemodelan visual, *UML* sangat berguna dalam perancangan sistem berbasis objek. Diagram-diagramnya tidak hanya digunakan untuk menggambarkan struktur kode, tetapi juga untuk mendefinisikan proses bisnis dan hubungan antar komponen dalam

sistem. Berbagai jenis diagram *UML* sering dimanfaatkan dalam pengembangan perangkat lunak guna memberikan gambaran yang lebih jelas tentang bagaimana sistem akan berfungsi secara keseluruhan.

1. *Use Case*: adalah representasi dari fungsionalitas yang menjadi harapan dari sebuah sistem, me interaksi antara *aktor* dan sistem.

Dalam *Use Case*, *aktor* merujuk pada entitas, baik manusia ataupun sistem lain, yang berkaitan dengan sistem untuk menjalankan suatu tugas.

Tabel 2.1 Simbol *Use Case Diagram*

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

2. *Activity Diagram*: Merupakan diagram *UML* yang digunakan untuk memberikan gambaran perilaku dinamis sebuah sistem atau komponen tertentu dengan menunjukkan alur kontrol antar aksi yang dijalankan oleh sistem.

Tabel 2.2 Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Initial Node	Bagaimana objek dibentuk atau diawali
4		Activity Final Node	Bagaimana objek dibentuk dan diakhiri
5		Decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

3. *Sequence Diagram*: adalah diagram yang menggambarkan interaksi antara objek-objek dalam sistem ataupun yang ada di luar sistem (seperti pengguna, tampilan, dan lain-lain) melalui pesan yang diatur menurut urutan waktu. *Sequence diagram* sering dipergunakan dalam mendeskripsikan skenario hingga tahapan yang diambil sebagai respons terhadap suatu peristiwa untuk menciptakan *output* khusu yang diinginkan.

Tabel 2.3 Simbol Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menggambar orang yang sedang berinteraksi dengan sistem.
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
3		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari sistem
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
5		<i>A focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message
6		<i>A message</i>	Menggambarkan Pengiriman Pesan

4. *Class Diagram*: ialah diagram yang menggambarkan struktur serta deskripsi dari kelas, paket, dan objek yang saling berkaitan, termasuk hubungan seperti pewarisan, asosiasi, dan lain sebagainya.

Tabel 2.4 Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1	—	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2	◇	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5	⊣-----	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6	-----→	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7	—	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya