

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terkait**

Pada penelitian yang dilakukan oleh Alfin Surya dan Riki Mukhaiyar, (2023). pada penelitian ini dibuat alat pengatur suhu otomatis pada kompor gas. Dalam penelitian ini menggunakan *power supply* dengan menggunakan *supply daya* sebesar 12Vdc dan 5Vdc yang digunakan sebagai sumber tegangan, DS18B20 sebagai sensor yang dapat mendeteksi suhu dari  $-55^{\circ}\text{C}$  sampai  $125^{\circ}\text{C}$ , *Driver relay* berguna untuk mengaktifkan pematik api[6].

Penelitian lainnya dilakukan oleh (Rendyansyah, Renaldi Ago, dan Bambang Tutoko, 2022) tentang implementasi sensor gas dan termokopel untuk memonitor proses pemanggangan dimana termokopel tipe-K digunakan sebagai sensor untuk mendeteksi suhunya. Pengguna dapat memonitor dan indikator dari makanan yang dipanggang. Indikatornya berupa *buzzer* yang mengeluarkan suara apabila sinyal dari sensor termokopel mencapai ambang yang ditentukan. Sedangkan LCD akan menampilkan tulisan yang menginformasikan bahwa makanan tersebut telah matang[7].

Era industri 4.0 menuntut industri kecil menengah untuk menyesuaikan diri dengan kemajuan teknologi saat ini. Banyak UKM yang membutuhkan oven untuk memenuhi produksinya. Berdasarkan hal tersebut maka dibutuhkan sistem monitoring dan kontrol temperatur oven agar produksinya terjaga. Desain elektrik menggunakan sensor termokopel sebagai sensor suhu

yang dihubungkan dengan arduino untuk menggerakkan blower dalam ruang pemanas, kemudian data dikirim ke *server* menggunakan ESP 8266 melalui internet. Data tersebut kemudian dapat diakses melalui *smartphone* (Ekohariadi, Yeni Anistyasari, Syarriefudin, dan Ricky Eka, 2022).

Untuk memasak makanan, suhu standar yang dibutuhkan agar makanan itu matang adalah  $100^{\circ}\text{C}$  karena umumnya pada titik didih tersebut bakteri maupun kuman yang terdapat pada makanan tersebut akan mati. Dalam penelitian ini dibuat sistem dengan menggunakan sensor suhu untuk mengidentifikasi ketika suhu mencapai maksimalnya. Aksi menyalakan dan mematikan kompor akan direkam oleh sistem dan disimpan ke dalam basis data (Timbo Faritcan, Endang Paza, 2019) [8].

Penelitian lainnya dilakukan oleh (Riswandi, 2022). Pada penelitian ini dilakukan pengembangan sistem keamanan pada kompor. Untuk meminimalisir dampak dari lupa mematikan kompor dirancang sebuah sistem yang dapat mematikan kompor secara otomatis dengan menggunakan parameter suhu sebagai *input*. *Input* sistem menggunakan sensor suhu DS18B20, sensor ini akan membaca suhu masakan dan akan ditampilkan di LCD [9].

## 2.2 Landasan Teori

### 2.2.1 Software Arduino Uno

*IDE* adalah kependekan dari *Integrated Development Environment* atau lingkungan terintegrasi untuk pengembangan.

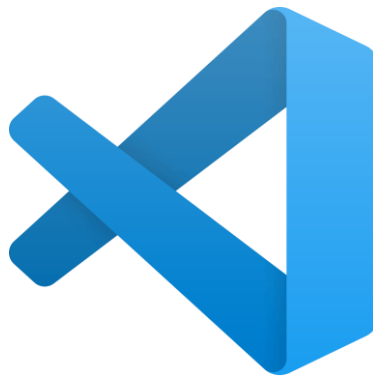
Disebut sebagai “lingkungan” karena *software* ini memungkinkan arduino untuk melakukan pemrograman dan melakukan tugas yang telah dibenamkan dalam sintaks pemrograman. Arduino memiliki bahasa pemrograman yang mirip dengan bahasa C. bahasa ini telah diubah untuk memudahkan pemula untuk menggunakannya dari awal. sebelum produk tersebut tersedia di pasar, pada mikrokontrolller arduino telah ditanamkan suatu program bernama *Bootlader*. Program ini berfungsi sebagai penghubung antara mikrokontroller dan *compiler* arduino. Arduino *IDE* dibuat dengan menggunakan bahasa pemrograman java. Selain itu, Arduino *IDE* dilengkapi dengan C/C++ yang dikenal sebagai *wiring* yang memudahkan *input* dan *output*. *Software processing* diubah menjadi Arduino *IDE* khusus pemrograman sebelum pengembangan Arduino *IDE* ini.



Gambar 2. 1 Arduino *IDE*

### 2.2.2 Visual Studio Code

Visual Studio Code merupakan *software code editor* yang bisa digunakan pada perangkat dengan OS MacOS, Linux, maupun Windows. Visual Studio Code bisa digunakan untuk mengedit kode dari banyak macam bahasa pemrograman pemrograman mulai dari JavaScript, TypeScript, hingga Node.js. Tidak hanya itu, Visual Studio Code juga mendukung bahasa pemrograman seperti PHP, Python, Java, dan NET. Hal ini karena Visual Studio Code memiliki ekosistem yang luas dan *extension* yang banyak. Beberapa fitur Visual Studio Code yang menjadikannya sebagai *software* paling banyak saat ini diantaranya *basic editing*, *intellisense*, *debugging*, *extension marketplace* dan *github integration*.



Gambar 2. 2 Visual Studio Code

### 2.2.3 XAMPP

XAMPP adalah *software web server* lokal yang memungkinkan membuat *website*, aplikasi, dan database secara *offline* menggunakan komputer. Tujuannya adalah untuk memastikan bahwa semua berjalan lancar sebelum mengupload *website*, aplikasi, atau database ke *server*.

*XAMPP* adalah program *open source* yang mencakup berbagai aplikasi berbasis *web server*. Berbagai sistem operasi termasuk Linux, Windows, MacOS, dan solaris dapat digunakan oleh aplikasi ini. *XAMPP* berfungsi sebagai *server* lokal dan sudah termasuk program Apache, MySQL, dan PHP. Awalnya, *XAMPP* muncul karena masalah dalam instalasi Apache dan menambahkan dukungan PHP dan MySQL. Hal ini membuat *XAMPP* menjadi aplikasi yang memudahkan pengembang yang membutuhkan *web server* di localhost dengan satu aplikasi.



Gambar 2. 3 *XAMPP*

#### **2.2.4 Chrome**

*Google Chrome* adalah peramban *web* lintas platform yang dikembangkan oleh *Google*. Peramban ini pertama kali dirilis pada tahun 2008 untuk Microsoft Windows kemudian diporting ke android, iOS, Linux, dan MacOS yang menjadikannya sebagai peramban bawaan dalam sistem operasi. Salah satu fungsi *Google Chrome* yaitu digunakan untuk mengembangkan *web*.



Gambar 2. 4 *Google Chrome*

### **2.2.5 Bootstrap**

*Bootstrap* adalah sebuah kerangka kerja yang terdiri dari HTML, CSS, dan JavaScript yang dirancang untuk mempermudah proses desain *website* yang responsif. Dengan menggunakan *Bootstrap* pengguna dapat dengan cepat mengembangkan tata letak dan antarmuka *website* yang dapat menyesuaikan diri dengan berbagai ukuran layar, termasuk dekstop dan perangkat seluler. Tujuan utama *Bootstrap* adalah untuk memastikan bahwa semua elemen pada sebuah *website* dapat berfungsi secara optimal di berbagai jenis perangkat. Selain fokus pada responsivitas, *Bootstrap* juga dikenal sebagai *framework open source* yang gratis digunakan. Hal ini membuatnya sangat populer di kalangan pengembang *web* karena menyediakan skrip dan sintaks yang dapat digunakan untuk membangun berbagai komponen desain *web* dengan mudah dan efisien. Dengan *Bootstrap* pengguna dapat memanfaatkan berbagai fitur bawaan untuk mempercepat proses pengembangan *website* tanpa perlu memulai dari awal[10].



Gambar 2. 5 *Bootstrap*

### 2.2.6 *Website*

Salah satu jenis media yang dapat ditemukan di *internet* adalah *website*. *Website* adalah kumpulan halaman *web* yang terletak di *internet*, yang biasanya terkumpul dalam sebuah domain atau *subdomain*, yang merupakan lokasinya di *WWW (World WIDE Web)*. Sebuah halaman *web* adalah dokumen yang ditulis dalam format *HTML (Hyper Text Markup Language)*, yang hampir selalu dapat diakses melalui protokol *HTTP*. *HTTP* adalah protokol yang mengirimkan informasi kepada pengguna melalui *web browser* dari *server website* untuk ditampilkan kepada mereka.

Halaman dari *situs web* dapat diakses melalui *URL* yang biasanya disebut *Homepage*. *URL* ini mengatur halaman-halaman situs menjadi sebuah hierarki. Namun, *hyperlink-hyperlink* dihalaman tersebut membantu pembaca memahami susunan keseluruhan dan bagaimana aliran informasi berjalan. Beberapa situs *web* memerlukan langganan *input* data agar pengguna dapat mengakses sebagian atau seluruh isi *situs web*.



Gambar 2. 6 Website

### 2.2.7 UML (Unified Modeling Language)

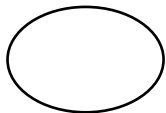



*UML (Unified Modeling Language)* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma “berorientasi objek”. Modelling sebenarnya digunakan untuk menyederhanakan masalah yang kompleks sehingga lebih mudah dipelajari dan dipahami. *UML* dapat digunakan untuk visualisasi, spesifikasi, konstruksi, dan dokumentasi berbagai bagian dari sistem dalam perangkat lunak. Dengan kata lain, seperti seorang arsitek yang membuat dokumen cetak biru untuk digunakan oleh perusahaan konstruksi dalam arsitek perangkat lunak membuat diagram *UML* untuk membantu *programmer/developer* membuat bangunan. Beberapa simbol yang digunakan saat menggambar *UML*:

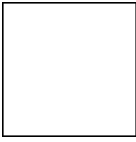



## 1. *Usecase Diagram*

*Usecase* atau *diagram usecase* adalah teknik guna menunjukkan hubungan antara pengguna sistem dengan sistem itu sendiri. *Usecase* mendeskripsikan interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dikembangkan. Secara umum, *usecase* digunakan untuk mengidentifikasi fungsi-fungsi dalam sistem informasi dan siapa saja yang berhak menggunakannya. Hasilnya adalah skema sederhana yang memungkinkan pengguna membaca dan memahami informasi. Berikut adalah beberapa simbol yang digunakan untuk membuat *usecase diagram*.

Tabel 2. 1 Simbol-Simbol *Usecase Diagram*


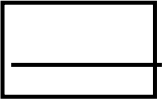
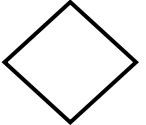
No	Simbol	Nama	Keterangan
1		<i>Usecase</i>	Struktur dan interaksi antara aktor dan sistem
2.		Aktor	Aktor menggambarkan peran orang atau sistem dengan sistem yang sedang dibuat.
3.		<i>Association</i>	<i>Association</i> adalah garis yang menghubungkan dua objek.
4.		<i>Include</i>	<i>Include</i> menunjukkan bahwa <i>use case</i> bergantung pada <i>use case</i> lain.




No	Simbol	Nama	Keterangan
5.		<i>System</i>	Pada suatu titik yang diberikan.
6.		<i>Extend</i>	<i>Extend</i> digunakan untuk menunjukkan bahwa satu <i>use case</i> dapat memperluas perilaku <i>use case</i> lain

## 2. *Class Diagram*

*Class diagram* adalah bagian dari *UML*. Dalam *class diagram* ini struktur, deskripsi, dan hubungan antar kelas digambarkan. Diagram ini menampilkan *class* objek yang membentuk sistem serta hubungan di antara *class* objek tersebut. Setiap *class* memiliki atribut dan metode atau operasi.

Tabel 2. 2 Simbol - Simbol *Class Diagram*


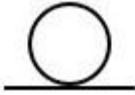
No	Simbol	Nama	Keterangan
1.		<i>Generalization</i>	<i>Generalization</i> , merupakan dimana objek <i>descendent</i> membagikan perilaku dan struktur data objek induknya.
2.		<i>Class</i>	<i>Class</i> , adalah kelompok objek yang berbagi satu sama lain.
3.		<i>Nary Association</i>	<i>Nary Association</i> , digunakan untuk asosiasi terhindar dengan objek lainnya.

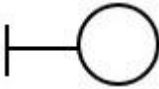


No	Simbol	Nama	Keterangan
4.		<i>Collaboration</i>	<i>Collaboration</i> merupakan deskripsi urutan aksi yang menampilkan interaksi sistem dengan aktor dan menghasilkan konsekuensi yang dapat diukur.
5.		<i>Dependency</i>	Hubungan dimana satu kelas bergantung pada kelas lain. Digambarkan sebagai garis putus-putus dengan ujung panah sederhana
6.		<i>Association</i>	<i>Association</i> adalah hubungan yang menghubungkan satu objek dengan objek lainnya melalui garis.

### 3. *Sequence Diagram*

*Sequence diagram* adalah suatu diagram yang menjelaskan interaksi objek dan menunjukkan interaksi objek dan menunjukkan (memberi tanda atau petunjuk) komunikasi diantara objek-objek tersebut.

Tabel 2. 3 Simbol-Simbol *Sequence Diagram*




No	Simbol	Nama	Keterangan
1.		Aktor	Menggambarkan orang yang sedang berinteraksi dengan sistem
2.		<i>Entity Class</i>	<i>Entity class</i> , merupakan bagian sistem yang membentuk deskripsi awal sistem dan berisi kumpulan kelas dalam bentuk entitas yang mendasari untuk membuat <i>database</i> .

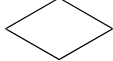
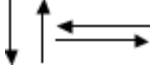
No	Simbol	Nama	Keterangan
3		<i>Boundary Class</i>	Berisi kumpulan kelas yang berfungsi sebagai interface atau interaksi dengan sistem untuk satu atau lebih aktor.
4		<i>Control Class</i>	Suatu entitas yang mengandung logika aplikasi yang tidak bertanggung jawab kepada entitas.
5		<i>A focus of Control &amp; A Life Line</i>	Menggambarkan tempat dimulai dan diakhiri <i>message</i>

#### 4. *Activity Diagram*

*Activity diagram* menunjukkan alur kerja sebuah proses bisnis dan urutan aktivitasnya. Diagram ini sangat mirip dengan *flowchart* karena dapat menunjukkan alur kerja dari satu aktivitas ke aktivitas lain atau satu aktivitas ke keadaan sesaat.

Tabel 2. 4 Simbol-Simbol *Activity Diagram*

No	Simbol	Nama	Keterangan
1.		<i>Activity</i>	<i>Activity</i> , menunjukan gambaran suatu proses.
2.		<i>Initial Node</i>	Menunjukkan titik permulaan dari alur aktivitas.
3.		<i>Activity Final Node</i>	Menunjukkan bahwa alur aktivitas telah selesai..

No	Simbol	Nama	Keterangan
4.		<i>Decision</i>	<i>Decision</i> , merupakan membuat suatu keputusan <i>false or true</i> .
5.		<i>Line Conecctor</i>	Digunakan untuk menghubungkan simbol