

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Penelitian Penelitian ini bertujuan untuk mengeksplorasi penerapan mikrokontroler ESP32 dan sensor ultrasonik dalam sistem . ESP32 dipilih sebagai komponen utama karena kemampuannya dalam menyediakan konektivitas Wi-Fi dan Bluetooth, menjadikannya sangat cocok untuk aplikasi berbasis Internet of Things (IoT).

Metode penelitian yang dijelaskan oleh Sabdo wiguno (2022) akan menjadi pedoman penting dalam merancang dan mengevaluasi sistem yang dikembangkan dalam penelitian ini. Dengan mengintegrasikan teknologi ESP32 bersama sensor ultrasonik, penelitian ini bertujuan untuk menyumbangkan inovasi yang lebih canggih dan efisien dalam pengelolaan sistem yang berbasis teknologi cerdas, yang dapat meningkatkan kenyamanan dan efisiensi secara keseluruhankeseluruhan[3].

Penelitian oleh Abdul Wafi dkk. (2024) ini mengembangkan sistem tempat sampah pintar berbasis mikrokontroler yang dilengkapi sensor *ultrasonik* untuk mendeteksi volume sampah, sensor *loadcell* untuk berat, serta modul GPS untuk mengetahui lokasi. Data yang dihasilkan dikirim ke platform *Thingspeak* dan ditampilkan dalam bentuk web yang terintegrasi dengan Google Maps. Sistem ini bertujuan membantu petugas kebersihan dalam mengelola tempat sampah secara efisien dengan mengetahui status

penyedia lokasi secara *real-time*, serta memiliki akurasi yang cukup tinggi dalam pengukuran volume dan lokasi[4].

Pada penelitian lain yang dilakukan oleh Abdur Rauf dkk. (2022) ,ini merancang alat tempat sampah pintar yang mampu memisahkan sampah berdasarkan material logam dan non-logam dengan memanfaatkan sensor dan *motor servo*. Selain pemisahan otomatis, sistem ini mengirimkan informasi status tempat sampah ke antarmuka berbasis web serta memberikan notifikasi WhatsApp ketika tempat sampah hampir penuh. Teknologi *IoT* memungkinkan monitoring jarak jauh dan bertujuan meningkatkan efisiensi pemilahan dan pengelolaan sampah secara digital dan *real-time*[5].

Penelitian lain juga oleh R. Wulandari dkk.(2023) ini berfokus pada pengembangan sistem tempat sampah otomatis yang dilengkapi dengan motor servo *MG996R* dan sensor *ultrasonik*, yang bertujuan untuk meningkatkan efisiensi dalam pemilahan sampah organik dan anorganik. Dalam penelitian ini, dilakukan analisis terhadap sudut pergerakan *motor servo* dalam kondisi standby dan saat membuka, serta pengukuran nilai error yang dihasilkan untuk masing-masing *motor servo*. Hasil penelitian menunjukkan bahwa *motor servo MG996R* (1) memiliki nilai error terendah sebesar 3%, sedangkan *motor servo MG996R* (3) menunjukkan nilai error tertinggi sebesar 33%. Selain itu, sistem ini juga mampu mendeteksi berbagai jenis sampah dengan persentase respon yang bervariasi, sehingga dapat diharapkan dapat berkontribusi dalam pengelolaan sampah yang lebih baik[6].

Penelitian ini yang dilakukan oleh B. A. Ramadhan dkk.(2022) berfokus pada perancangan dan pengembangan tempat sampah otomatis berbasis *mikrokontroler Arduino* yang dilengkapi dengan berbagai sensor, seperti sensor *ultrasonik* dan sensor *PIR (Passive Infrared)*, serta modul *GSM SIM 900* untuk meningkatkan efisiensi pengelolaan sampah. Dengan memanfaatkan teknologi modern, alat ini dirancang untuk mendeteksi kedatangan pengguna dan kapasitas tempat sampah, serta mengirimkan notifikasi melalui SMS kepada petugas kebersihan ketika tempat sampah sudah penuh. Inovasi ini diharapkan dapat meningkatkan kesadaran masyarakat dalam membuang sampah pada tempatnya dan menciptakan lingkungan yang lebih bersih dan sehat[7].

Penelitian yang dilakukan Muhammad Akbar dkk. (2023) merancang sistem pemantauan tempat sampah khusus untuk rumah sakit, menggunakan kombinasi *Arduino* dan Wemos D1 Mini untuk mengontrol sensor jarak yang mengukur ketinggian sampah. Data dikirimkan ke sistem berbasis web yang menampilkan status penuh atau kosong dari tiap tempat sampah. Monitoring dilakukan secara otomatis setiap lima detik, memberikan solusi cepat dan tepat untuk penanganan sampah medis agar lingkungan rumah sakit tetap bersih dan terkontrol[8].

Penelitian ini mengembangkan sistem tempat sampah pintar dengan sensor *ultrasonik* untuk mendeteksi tingkat kepenuhan tempat sampah. NodeMCU ESP8266 digunakan untuk mengirimkan data secara *real-time* ke *database Google Firebase*. Data kemudian ditampilkan pada *dashboard*

berbasis *website* yang mudah diakses oleh pengguna. Sistem ini berhasil menunjukkan bahwa penggunaan *Firebase* memberikan kecepatan dan keandalan tinggi dalam menyimpan dan menampilkan data monitoring tempat sampah secara efisien[9].

Penelitian oleh Visa Bili Nur dkk. (2021) ini bertujuan menciptakan sistem pemantauan tempat penampungan sampah (TPS) yang mampu memberikan informasi kondisi dan lokasi TPS secara *real-time*. Sistem menggunakan sensor *ultrasonik* untuk mengukur tinggi tumpukan sampah dan modul GPS untuk pelacakan lokasi. Data dikirim melalui mikrokontroler ke sistem web yang menampilkan status tiap TPS dalam bentuk visual. Dengan sistem ini, pengelola dapat mengetahui TPS yang sudah penuh dan membutuhkan pengangkutan, sehingga meningkatkan efisiensi operasional[10].

2.2 Landasan Teori

2.2.1 Tempat Sampah

Tempat sampah adalah wadah yang digunakan untuk menampung sampah sebelum dibuang atau didaur ulang. Biasanya, tempat sampah terbuat dari bahan seperti plastik, logam, atau kayu dan tersedia dalam berbagai ukuran serta desain sesuai kebutuhan. Berdasarkan jenis sampah yang ditampung, tempat sampah dibagi menjadi tiga kategori utama: tempat sampah organik untuk sampah yang dapat terurai alami seperti sisa makanan dan dedaunan, tempat

sampah anorganik untuk sampah yang tidak mudah terurai seperti plastik dan kaca, serta tempat sampah B3 (Bahan Berbahaya dan Beracun) yang digunakan untuk limbah berbahaya seperti baterai dan bahan kimia. Fungsi utama tempat sampah adalah menjaga kebersihan lingkungan, mencegah penyebaran penyakit, serta membantu dalam pemilahan dan daur ulang sampah agar lebih efektif. Penggunaan tempat sampah yang tepat dapat meningkatkan kesadaran masyarakat akan kebersihan, mengurangi pencemaran lingkungan, serta mendukung pengelolaan sampah secara efisien.

Selain itu, regulasi dan kebijakan terkait tempat sampah diatur dalam berbagai peraturan, seperti Undang-Undang No. 18 Tahun 2008 tentang Pengelolaan Sampah dan peraturan daerah yang mendukung kebersihan lingkungan. Pemerintah juga menjalankan kampanye dan program edukasi untuk meningkatkan pemilahan sampah yang lebih baik. Seiring perkembangan teknologi, kini terdapat tempat sampah modern yang dilengkapi dengan fitur otomatis seperti sensor untuk membuka tutupnya, sistem pemilahan otomatis, dan sistem kompos yang mengolah sampah organik menjadi pupuk. Dengan memahami landasan teori ini, diharapkan masyarakat lebih sadar akan pentingnya penggunaan tempat sampah dalam menjaga kebersihan dan keberlanjutan lingkungan[11].

2.2.2 Tempat Sampah Pintar

Kemajuan teknologi di era modern telah memberikan pengaruh besar dalam berbagai aspek kehidupan, termasuk dalam pengelolaan sampah dan limbah. Salah satu inovasi yang berkembang adalah tempat sampah pintar, yang memanfaatkan teknologi terkini seperti *Internet of Things (IoT)* untuk menciptakan solusi pengelolaan limbah yang lebih efektif dan ramah lingkungan. Tempat sampah pintar dirancang untuk memudahkan proses pengelolaan sampah dengan memanfaatkan sensor dan sistem otomatis untuk mendeteksi, mengelola, serta mengirimkan data mengenai kondisi sampah secara efisien.

Sistem ini terdiri dari beberapa komponen utama, termasuk sensor ultrasonik, *mikrokontroler*, dan modul komunikasi. Sensor ultrasonik bertugas mengukur tingkat sampah berdasarkan jarak antara sensor dengan permukaan sampah. Data yang diperoleh kemudian diproses oleh mikrokontroler, seperti *Arduino* atau *Raspberry Pi*, yang bertindak sebagai pusat pengendalian. Informasi ini selanjutnya dikirimkan ke server melalui koneksi IoT untuk diolah lebih lanjut dan ditampilkan dalam aplikasi berbasis web atau perangkat lainnya[12].

Kelebihan lain dari tempat sampah pintar adalah kemampuannya untuk mengotomatisasi beberapa proses, seperti membuka dan menutup pintu tempat sampah menggunakan *motor servo* atau *aktuator*. Fitur ini tidak hanya memberikan kenyamanan bagi pengguna tetapi juga menjaga kebersihan dengan meminimalkan kontak langsung

antara pengguna dan tempat sampah. Selain itu, sistem ini dilengkapi dengan notifikasi ketika tempat sampah penuh, memungkinkan pengelola untuk segera mengambil tindakan sehingga tumpukan sampah dapat dihindari[13].

Konsep tempat sampah pintar sejalan dengan visi *smart city* yang memanfaatkan teknologi untuk meningkatkan kualitas hidup masyarakat. Johnson et al. dalam penelitiannya menemukan bahwa sistem pengelolaan sampah berbasis *IoT* mampu mengurangi biaya operasional hingga 30% dengan mengoptimalkan pengangkutan hanya di lokasi tempat sampah yang sudah penuh. Ini juga membantu mengurangi emisi karbon dari kendaraan pengangkut sampah yang tidak perlu mengunjungi lokasi yang belum memerlukan pengosongan[14].

Dengan teknologi informasi dan komunikasi sebagai pendukung utama, tempat sampah pintar berpotensi menjadi solusi inovatif dalam pengelolaan limbah modern. Sistem ini tidak hanya memberikan manfaat praktis dalam mengatasi masalah sampah yang berserakan tetapi juga mendorong masyarakat untuk lebih peduli terhadap lingkungan. Dengan demikian, inovasi ini dapat berkontribusi dalam mewujudkan lingkungan yang bersih, sehat, dan berkelanjutan[2].

2.2.3 Website

Website adalah aplikasi berbasis web yang berjalan di atas protokol HTTP/HTTPS dan diakses melalui alamat domain tertentu. *Website* biasanya terdiri dari komponen *frontend* dan *backend*. Bagian *frontend* adalah tampilan antarmuka yang dilihat oleh pengguna, sedangkan *backend* bertanggung jawab mengelola logika aplikasi, proses, serta interaksi dengan *database*.

2.2.4 Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS) adalah bahasa desain yang berfungsi untuk mengatur tampilan visual dan tata letak pada halaman web. *CSS* digunakan bersama *HTML* untuk memberikan gaya pada berbagai elemen web, seperti mengatur warna, ukuran, font, margin, tata letak, animasi, hingga responsivitas. Dengan *CSS*, halaman web dapat dirancang agar lebih menarik secara estetika, terorganisir, dan nyaman diakses oleh pengguna di berbagai perangkat, baik desktop maupun seluler. Kehadiran *CSS* memungkinkan pengembang untuk memisahkan konten dan desain, sehingga mempermudah proses pengelolaan dan pembaruan situs web.

2.2.5 HTML

HyperText Markup Language (HTML) adalah bahasa markup standar yang digunakan untuk membangun dan menentukan struktur konten pada halaman web. *HTML* berfungsi memberikan instruksi kepada browser tentang cara menampilkan berbagai elemen seperti

teks, gambar, video, dan komponen lainnya. Meskipun *HTML* tidak bertanggung jawab terhadap aspek visual, melainkan lebih fokus pada pengaturan struktur dan isi halaman, tampilan dan tata letak visualnya dapat dikustomisasi dengan menggunakan *CSS* untuk menciptakan desain yang lebih menarik.

2.2.6 PHPMyAdmin

PhpMyAdmin adalah aplikasi berbasis web yang dirancang untuk mempermudah pengelolaan basis data *MySQL* atau *MariaDB*. Alat ini memungkinkan pengguna untuk berinteraksi dengan *database* melalui antarmuka grafis yang intuitif, sehingga mengurangi kebutuhan untuk menulis perintah *SQL* secara manual. *PhpMyAdmin* sangat populer berkat kemudahannya dalam digunakan, fitur yang lengkap, dan aksesibilitasnya melalui *browser web*, menjadikannya pilihan utama bagi pengembang untuk mengelola *database* secara efisien.

2.2.7 Internet of Things (IoT)

Internet of Things (IoT) merupakan suatu sistem yang memungkinkan berbagai objek dan individu terhubung dalam suatu jaringan, sehingga dapat bertukar data secara otomatis tanpa memerlukan interaksi langsung antar manusia yaitu sumber ke tujuan atau interaksi manusia ke komputer. *IoT* merupakan perkembangan teknologi yang menjanjikan, *IoT* dapat mengoptimalkan kehidupan dengan sensor sensor cerdas dan benda yang memiliki jaringan dan bekerjasama dalam internet [15].

2.2.8 Visual Studio Code

Visual Studio Code adalah editor teks sumber terbuka yang disediakan secara gratis oleh Microsoft. Aplikasi ini bersifat lintas platform, sehingga dapat digunakan di berbagai sistem operasi seperti Windows, Linux, dan macOS. Visual Studio Code adalah aplikasi yang sangat kuat yang, meskipun ringan, menawarkan banyak kemampuan. Editor ini mendukung berbagai bahasa pemrograman, termasuk Java, JavaScript, Go, C++, dan banyak lainnya. Visual Studio Code juga dilengkapi dengan berbagai fitur berguna seperti penyorotan sintaks, penyelesaian kode, potongan kode, refaktorisasi, debugging, dan integrasi Git. Seperti editor kode lainnya, misalnya sublime atau atom. Visual studio code dapat mengenali berbagai bahasa pemrograman dan secara otomatis menyesuaikan skema warna sesuai dengan fungsi dalam kode yang sedang ditulis. Ini menjadikannya aplikasi yang sangat

efisien bagi para pengembang *website* software, maupun aplikasi lainnya yang kompleks [16].

2.2.9 MySQL

MySQL *MySQL* adalah sistem manajemen basis data relasional (RDBMS) yang bersifat *open-source* dan banyak digunakan untuk mengelola data dalam berbagai aplikasi, baik skala kecil maupun besar. MySQL menggunakan *Structured Query Language (SQL)* sebagai bahasa utama dalam pengelolaan data, termasuk proses penyimpanan, pengambilan, dan manipulasi informasi dalam *database*. Keunggulan MySQL terletak pada kecepatan, kestabilan, dan fleksibilitasnya, serta kompatibilitasnya dengan berbagai sistem operasi seperti Windows, Linux, dan macOS. Selain itu, MySQL sering digunakan dalam pengembangan aplikasi berbasis web, seperti sistem manajemen konten (CMS), *e-commerce*, dan aplikasi berbasis cloud, karena kemampuannya dalam menangani transaksi yang kompleks dengan integritas data yang tinggi[17].

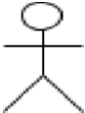
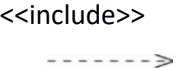
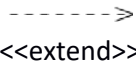
2.2.10 UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah bahasa pemodelan visual yang berfungsi untuk mendeskripsikan, merancang, dan mendokumentasikan sistem perangkat lunak. *UML* menyediakan berbagai jenis diagram yang digunakan untuk menggambarkan struktur, perilaku, serta interaksi dalam sistem. Bahasa ini sering digunakan dalam pengembangan perangkat lunak berbasis objek.

Dengan *UML*, pengembang, arsitek perangkat lunak, dan analis sistem dapat merancang serta memahami sistem yang sedang dibangun dengan cara yang lebih sistematis dan mudah dipahami. Terdapat beberapa diagram *UML* yang sering digunakan dalam pengembangan sebuah sistem yaitu: [18]

1. *Use Case* adalah representasi visual dari fungsionalitas yang diharapkan dari suatu sistem, menggambarkan interaksi antara aktor dan sistem tersebut. Dalam *Use Case*, terdapat aktor yang berperan sebagai entitas, baik pengguna maupun sistem lain, yang berinteraksi dengan sistem untuk menjalankan tugas tertentu. Berikut pada Tabel 2.1.

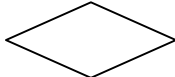
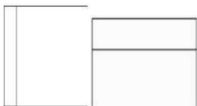
Tabel 2. 1 *Use Case Diagram*

No.	Gambar	Nama	Keterangan
1		<i>Actor</i>	Orang, proses atau sistem yang berinteraksi dengan sistem lain.
2		<i>Dependency</i>	Hubungan perubahan yang terjadi pada suatu elemen mandiri
3		<i>Generalization</i>	Hubungan generalisasi (umum-khusus) antar dua buah use case
4		<i>Include</i>	Relasi use case tambahan ke sebuah use case, dimana use case yang ditambahkan dapat berdiri sendiri.
5		<i>Extend</i>	Relasi tambahan yang Menunjukkan use case lain memerlukan use case. Syarat untuk

No.	Gambar	Nama	Keterangan
			agar fungsinya dapat berjalan dengan baik.
6		<i>Association</i>	Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

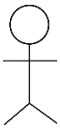
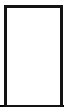
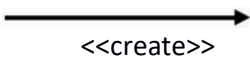
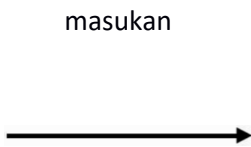
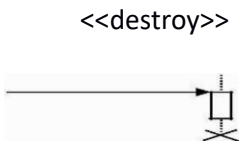
2. *Activity Diagram* adalah jenis diagram dalam *UML* yang menggambarkan alur dinamis dari suatu sistem atau bagian tertentu dari sistem, melalui urutan kontrol yang menghubungkan berbagai aksi atau kegiatan yang dilakukan oleh sistem. Lihat pada Tabel 2.2.

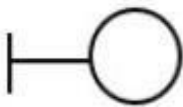
Tabel 2. 2 Activity Diagram

No.	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana kelas saling berinteraksi
2		<i>Decision</i>	Asosiasi percabangan jika ada pilihan yang lebih dari satu
3		<i>Initial Node</i>	Bagaimana objek diawali. Hanya ada satu initial dalam satu diagram.
4		<i>Activity Final Node</i>	Bagaimana objek diakhiri. Dalam satu diagram minimal ada satu final state
5		<i>Fork Node</i>	Satu aliran pada tahap yang berubah menjadi beberapa aliran
6		<i>Control Flow</i>	Menunjukkan urutan aliran aktivitas.
7		<i>Partition</i>	Simbol yang membatasi aktivitas antar orang, organisasi, sistem atau kelompok

3. *Sequence Diagram* adalah diagram yang menggambarkan interaksi antara objek-objek di dalam dan sekitar sistem (seperti pengguna, tampilan, dan lainnya) dalam bentuk pesan yang diatur berdasarkan waktu. Diagram ini sering digunakan untuk menggambarkan skenario atau serangkaian langkah yang diambil sebagai respons terhadap sebuah peristiwa untuk menghasilkan hasil tertentu[19]. Lihat Pada Tabel 2.3.

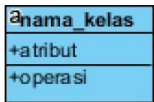
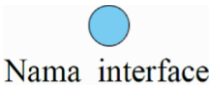
Tabel 2. 3 Sequence Diagram

No.	Gambar	Nama	Keterangan
1		Actor	Orang , proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat
2		Lifeline	Menyatakan kehidupan suatu objek
3		Object	Menyatakan objek yang berinteraksi pesan
4		Timelife	Menyatakan objek dalam berinteraksi pesan
5		Pesan tipe Create	Menyatakan objek dalam keadaan aktif dan berinteraksi
6		Pesan tipe Call	Menyatakan suatu objek memanggil operasi/ metode yang ada pada objek lain atau dirinya sendiri
7		Pesan tipe Send	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lain
8		Pesan tipe Return	Menyatakan bahwa suatu objek telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu
9		Pesan tipe Destroy	Menyatakan suatu objek telah menjalankan suatu operasi

10		Boundary Class	Berupan tepi dari sistem, seperti <i>user interface</i> atau alat yang berinteraksi dengan sistem lain
11		Control Class	Mengatur aliran dari informasi untuk sebuah skenario

4. *Class Diagram* adalah diagram yang menggambarkan struktur dan deskripsi dari kelas, paket, dan objek yang saling terhubung, termasuk hubungan seperti pewarisan, asosiasi, dan hubungan lainnya. Diagram ini digunakan untuk memodelkan komponen-komponen utama dalam sistem perangkat lunak dan interaksi antar elemen tersebut. Berikut pada Tabel 2.4.

Tabel 2. 4 *Class Diagram*

No.	Gambar	Nama	Keterangan
1		Kelas	Kelas pada struktur sistem
2		Antarmuka/ Interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3		Asosiasi/ Association	Relasi antar kelas dengan makna umum
4		Asosiasi berarah/ Directed Association	Relasi antar kelas dengan makna kelas

5		Generalisasi	Relasi antar kelas dengan makna generalisasi
6		Kebergantungan / <i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas
7		Agregasi / <i>Aggregation</i>	Relasi antar kelas dengan makna whole-part

2.2.11 Arduino IDE

Arduino IDE (Integrated Development Environment) merupakan suatu aplikasi *open source* sebagai *text editor* untuk membuat, membuka, mengedit, dan mevalidasi kode serta melakukan upload ke board *Arduino* atau mikrokontroler lainnya. *Arduino* memanfaatkan bahasa pemrograman yang mirip dengan C, tetapi dirancang khusus untuk memudahkan pengembangan perangkat keras sehingga mudah dipelajari, terutama bagi pemula. *Arduino IDE* juga menyediakan banyak contoh kode dan library yang siap pakai, memudahkan pengguna dalam membuat proyek elektronik [20].

2.2.12 Database

Database adalah sekumpulan data dan informasi yang disusun secara teratur dalam suatu tempat penyimpanan, memungkinkan kita mengaksesnya dengan mudah melalui program komputer. *Database* hadir untuk membantu kita mengelola data dengan lebih efektif, memeriksa, memproses, atau mengolahnya agar dapat menghasilkan

informasi yang bermanfaat. Pada intinya, *database* adalah kumpulan data yang saling terhubung, dirancang untuk memudahkan kehidupan kita dalam mengatur dan memahami informasi secara lebih terstruktur dan efisien [21].

2.2.13 Bootstrap

Bootstrap adalah *framework* CSS yang dirancang untuk membantu pengembang dalam membuat *website* yang responsif dan memiliki tampilan menarik dengan lebih mudah. Ketidakonsistenan terhadap aplikasi individual membuat pengembangan dan pemeliharannya menjadi sulit, sehingga penggunaan *framework* seperti Bootstrap menjadi solusi yang efektif. *Bootstrap* adalah CSS tetapi dibentuk dengan LESS, sebuah *pre-prosesor* yang memberi fleksibilitas dari CSS biasa. Selain itu, Bootstrap juga menyediakan komponen-komponen antarmuka siap pakai seperti tombol, form, dan navigasi yang dapat langsung digunakan dan dikustomisasi.

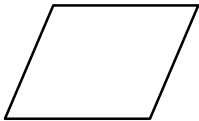
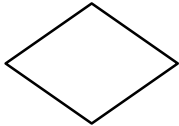
2.2.14 Laragon

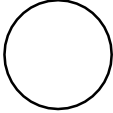
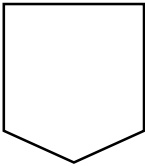
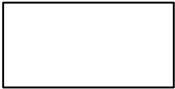
Laragon adalah perangkat lunak yang dibuat untuk menyederhanakan proses pembuatan dan pemeliharaan server lokal. Laragon menyediakan solusi untuk web developer yang ingin memiliki environment pembangunan yang efisien, stabil, dan fleksibel. Selain itu, Laragon juga mendukung berbagai stack teknologi seperti *PHP*, *Apache*, *Nginx*, *MySQL*, dan *Node.js*, serta memudahkan pengguna untuk memasang dan mengelola berbagai *framework* seperti *Laravel*, *Symfony*, dan *WordPress*. Dengan antarmuka pengguna yang sederhana dan beberapa fitur otomatisasi seperti auto *virtual hosts* dan SSL tombol Sendiri, *Laragon* adalah pilihan sempurna untuk membangun *website* lokal.

2.2.15 Flowchart

Flowchart merupakan sebuah diagram grafis yang digunakan untuk menggambarkan alur langkah-langkah dalam suatu program atau proses yang disusun secara terstruktur. Diagram ini menggunakan simbol-simbol standar yang mempresentasikan beragam tahapan atau aktivitas untuk memudahkan pembaca dalam memahami suatu program atau proses. Semua simbol dan fungsinya dipilih sesuai dengan jenis langkah atau aktifitas yang akan diilustrasikan. Berikut flowchart pada Tabel 2.5.

Tabel 2. 5 Flowchart

Simbol	Keterangan
	<i>Terminator / Terminal</i> Simbol yang digunakan untuk menunjukkan keadaan dasar dan kondisi terakhir dari suatu diagram alur program adalah lingkaran.. Dalam lingkaran tersebut biasanya dituliskan kata "Mulai" untuk state awal, dan "Selesai" atau "Akhir" untuk state akhir.
	<i>Preparation / Persiapan</i> Simbol yang digunakan untuk membedakan faktor-faktor yang akan dimanfaatkan dalam program pada <i>Flowchart</i> umumnya berbentuk persegi atau kotak.. Dalam kotak tersebut, variabel dapat diberi nama sesuai dengan tipe datanya, seperti string, numeric, Boolean, atau tanggal.
	Simbol " <i>Predefined Process</i> " atau " <i>Subroutine</i> " dalam <i>Flowchart</i> digunakan untuk menunjukkan proses yang dijelaskan secara terpisah sebagai subprogram, yang dimulai dan diakhiri dengan terminator.
	Dalam flowchat, simbol ini digunakan untuk mewakili proses input dan output. Input mencakup aktivitas seperti menerima data dari pengguna, misalnya melalui keyboard atau perangkat lainnya.
	Simbol keputusan dalam <i>Flowchart</i> digunakan untuk menentukan pilihan suatu kondisi (Ya atau Tidak). Memiliki dua keluaran arus dengan keterangan "Ya" dan "Tidak" untuk membedakannya dari simbol lain.

	Connector adalah simbol lingkaran digunakan untuk menghubungkan simbol satu dengan yang lain dalam satu halaman, memperjelas aliran program tanpa memerlukan penulisan arus yang panjang. Sedangkan untuk menghubungkan simbol antar-halaman digunakan pasangan simbol connector dengan label yang sama.
	On-Page Connector dalam <i>Flowchart</i>, yang digunakan untuk menghubungkan bagian-bagian <i>Flowchart</i> di halaman yang sama. Simbol ini berbentuk segi lima dan dilengkapi dengan identitas berupa huruf atau angka
	Arrow / Arus Simbol yang menentukan aliran dalam sebuah <i>Flowchart</i> adalah panah. Panah ini mewakili arus atau jalur dari satu simbol ke simbol lainnya, menggambarkan urutan langkah-langkah atau proses dalam program.
	Process / Proses Simbol yang digunakan untuk memasukkan nilai dan menampilkan nilai dari suatu variabel dalam <i>Flowchart</i> biasanya berbentuk panah. Simbol masukan ditandai dengan variabel yang belum mengalami operasi dari operator tertentu, seperti pemberian nilai atau penambahan nilai.

2.2.16 White box testing

White box testing adalah metode pengujian perangkat lunak yang melibatkan analisis dan pemeriksaan struktur internal serta kode sumber dari aplikasi. Berbeda dengan *black box testing*, yang hanya

menilai hasil berdasarkan input dan *output*, *white box testing* lebih menekankan pada pemahaman aliran data dan proses di dalam perangkat lunak.

2.2.17 Black box testing

Black box testing, atau yang sering disebut sebagai *Behavioral Testing*, adalah jenis pengujian yang menilai perangkat lunak berdasarkan hasil dari input dan output, tanpa memperhatikan bagaimana kode di dalamnya bekerja. Metode ini biasanya diterapkan pada tahap akhir pengembangan untuk memastikan bahwa perangkat lunak beroperasi dengan baik dan memenuhi kebutuhan yang telah ditentukan.