

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Penelitian Andi Muhamad Muuslim tahun 2020 berjudul Prototipe Got Totemusing (GREAT SORTING METAL MONITORING SYSTEM USING INTERNET OF THINGS). Tujuan dari penelitian ini adalah untuk membuat prototipe Got Totemusing. Proses pembuatan prototipe digunakan dalam metodologi penelitian. Ini dimulai dengan pengumpulan persyaratan, membangun prototipe, menilai prototipe, mengkodekan sistem, menguji sistem, dan mengevaluasi sistem. Proxymitas logam mendeteksi sampah logam, Proxymitas induktif mendeteksi sampah non-logam, Module DF-Player yang dapat memproses suara/MP3 dan WEMOS D1 sebagai prosesnya, dan sensor *ultrasonic* untuk mengetahui apakah sampahnya sudah terisi penuh. Web UBIDOTS menyediakan data sampah yang lengkap, dan speaker menunjukkan jenis sampah. Hasil pengujian menunjukkan bahwa prototipe telah dibuat dengan baik dan dapat mendeteksi sampah logam dan nonlogam. Jika prototipe terisi sepenuhnya, itu juga dapat memberikan informasi ke UBIDOTS [3].

Youla Gessel (2023) kemudian melakukan penelitian berjudul Sistem Pemilah Logam, Anorganik, dan Organik yang menggunakan Conveyor dan menggunakan *Internet of Things* untuk Memantau Ketinggian Sampah. Studi ini menggunakan NodeMCU ESP32 sebagai pengendali sistem utama.

Motor servo untuk pemilahan, sensor deteksi jenis sampah induktif, kapasitif, dan optik, dan sensor ultrasonik HC-SR04 untuk memantau ketinggian sampah. Antarmuka *web* digunakan untuk memantau tempat sampah dari jarak jauh. Selain itu, LED dan buzzer memberikan peringatan saat tempat sampah dalam keadaan penuh. Ketiga sensor ultrasonik HC-SR04 diuji dengan akurasi keberhasilan 96,67%. Nilai error untuk sensor HC-SR04 1 sebesar 2,517%, HC-SR04 2 sebesar 1,9315%, dan HC-SR04 3 sebesar 2,176%. Hasil pengujian sistem menunjukkan bahwa sampah yang terkumpul sudah sesuai dengan jenisnya dan membuatnya lebih mudah bagi pengguna untuk memantau tempat sampah [4].

Ali Wafi melakukan pengembangan tambahan pada tahun 2020 dengan mengembangkan prototipe sistem sampah pintar yang menggunakan aplikasi Android yang berbasis IoT (Internet of Things). Sampah menumpuk dan mengeluarkan bau busuk yang dapat menjadi sumber penularan penyakit karena tata kelola sampah yang buruk dan kurangnya kesadaran masyarakat tentang pentingnya membuang sampah pada tempatnya. Penanganan sampah menjadi lebih lambat karena petugas kebersihan tidak memiliki teknologi informasi yang diperlukan untuk mengelola sampah. Untuk menyelesaikan masalah ini, tempat sampah otomatis harus memilah sampah organik dan anorganik sesuai dengan jenisnya. Selain itu, perlu ditambahkan fitur IoT, atau *Internet of Things*, yang dapat mengetahui kapan tempat sampah penuh untuk cepat diproses. Mikrokontroler ESP 32 berfungsi sebagai pengontrol sistem untuk protokol tempat sampah pintar.

Sensor dekat induktif dan kapasitif yang dapat mengidentifikasi jenis sampah organik dan anorganik. Servo untuk memilah sampah dan mengontrol pintu tempat sampah Sensor ultrasonik untuk mengidentifikasi kehadiran orang dan ketinggian sampah Pembuatan alat ini menghasilkan keberhasilan deteksi sampah organik sebesar 95%, deteksi sampah anorganik sebesar 97,5%, keberhasilan sensor jarak buka tutup 99,26%, keberhasilan sensor jarak organik sebesar 99,07%, dan keberhasilan sensor jarak anorganik sebesar 99,21%. Selain itu, alat ini dapat mengirimkan hasil *monitoring* secara real time dan memberi tahu aplikasi Android jika tempat sampah telah penuh.[5].

2.2 Landasan Teori

2.2.1 Sistem monitoring

Sistem pemantauan pemilah tempat sampah adalah sistem yang menggunakan komputer untuk mengawasi hasil pemilah sampah. Dilihat dari perspektif manajemen kinerja, *monitoring* adalah proses terintegrasi yang memastikan bahwa proses berjalan sesuai rencana. *Monitoring* dapat memberikan informasi tentang keberlangsungan proses untuk menetapkan tindakan untuk perbaikan yang berkelanjutan. Pada pelaksanaannya, proses diawasi. Level kajian sistem *monitoring* mengacu pada kegiatan per kegiatan dalam suatu bagian (Wrihatnolo, 2008). Salah satu contohnya adalah bagian pembelian yang memesan barang kepada supplier. *Output* per proses

atau kegiatan adalah acuan untuk pengawasan.

2.2.2 Visual Studio Code

Pembuatan kode program membutuhkan sebuah aplikasi yang baik. Anda dapat menggunakan Visual Studio Code dalam hal ini, karena *software* ini sangat ringan tetapi memiliki editor kode sumber yang kuat yang dapat digunakan dari desktop. Ada dukungan built-in untuk JavaScript, naskah, dan Node.js, dan ada banyak ekstensi untuk bahasa lain seperti C ++, C #, Python, dan PHP [6].



Gambar 2. 1 Visual Studio Code

2.2.3 Database MySQL

Database adalah kumpulan data yang terdiri dari kumpulan data yang disusun sedemikian rupa sehingga menghasilkan informasi berguna. Data seperti nama, nomor, dan alamat adalah contohnya. Semua data ini digabungkan menjadi kelompok data baru yang disebut data informasi.



Gambar 2. 2 Mysql

2.2.4 Chrome

Google Chrome adalah browser situs lintas platform yang dikembangkan oleh Google. Ini pertama kali dirilis untuk Microsoft Windows pada tahun 2008 dan kemudian dikembangkan untuk Linux, macOS, iOS, dan Android. Versi yang digunakan dalam penelitian ini adalah Google Chrome 79.0. Beberapa produk Google Chrome yang dapat diakses termasuk Extension, Aplikasi Chrome, dan Tema.



Gambar 2. 3 Chrome

2.2.5 PhpMyAdmin

PhpMyAdmin adalah aplikasi *web* yang berasal dari subdomain `phpmyadmin.net` dan digunakan untuk mengelola database MySQL. Untuk menggunakan `phpmyadmin` pada layar utama WAMP, pilih `phpMyAdmin[7]`.



Gambar 2. 4 PhpMyAdmin

2.2.6 Bootstrap

Bootstrap pertama kali dibuat oleh Mark Otto dan Jacob

Thornton di kantor Twitter untuk memastikan bahwa langkah-langkah pengembangan *interface* tetap sama selama pembuatan *website*. Ini adalah salah satu jenis *framework* yang memungkinkan integrasi CSS dan JavaScript sebagai alternatif untuk *framework* lainnya. Bootstrap saat ini merupakan aplikasi *open-source* yang mendukung platform seperti HTML5 dan CSS3. Dengan menyediakan template untuk font, typography, tombol, navigasi, dan elemen lainnya yang dapat diakses melalui aplikasi, tujuannya adalah untuk membantu pembangun *web* membuat *interface* yang responsif[8].

2.2.7 Website

Website adalah media informasi yang berbasis jaringan komputer yang dapat diakses kapan saja dan di mana saja melalui internet. *Website* dapat menampung berbagai jenis informasi, seperti teks, suara, gambar, dan animasi, dan dapat dikelola dan diperluas secara *online*.

2.2.8 PHP (*Hypertext Preprocessor*)

Bahasa pemrograman ini berfungsi sebagai penerjemah data di server *web*. Basis data server *web* mengambil dan menyimpan informasi yang dikirim oleh klien. Dengan cara ini, informasi dapat diperbarui setiap kali pengguna mengaksesnya. Proses mentransfer data atau file dari komputer klien ke server *web* sehingga PHP dapat dijalankan dikenal sebagai pengunggahan file [9].

Sebuah program yang dapat mengumpulkan data dari komputer

klien atau server sehingga dapat dengan mudah dilihat di browser sangat diperlukan untuk membuat situs *web* yang selalu diperbarui dan mudah dikelola setiap hari di browser. Salah satu aplikasi yang dapat dijalankan di server adalah PHP. PHP bekerja dengan dokumen *Hypertext Markup Language* (HTML), memungkinkan Anda untuk memodifikasi situs *web* sesuai keinginan. Berkat PHP, kita dapat mengubah situs *web* kita menjadi aplikasi *web*, bukan hanya sekumpulan halaman statis yang sering diperbarui.



Gambar 2. 5 PHP

2.2.9 Esp8266

Akhir-akhir ini, para pengembang *hardware* semakin menyukai modul WiFi ESP8266. Ini bukan hanya murah tetapi juga bersifat SOC (*System on Chip*), yang berarti kita dapat memprogram ESP8266 secara langsung tanpa memerlukan mikrokontroler tambahan. Selain itu, ESP8266 memiliki kemampuan untuk berfungsi sebagai akses poin adhoc dan klien sekaligus [10].

“Espressif” adalah nama pengembang asal Tiongkok yang mengembangkan ESP8266. Seri produk ESP8266 saat ini masih



Gambar 2. 7 XAMPP

2.2.10 UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah bahasa yang telah menjadi standar industri untuk visualisasi, desain, dan dokumentasi sistem piranti perangkat lunak. Ini adalah salah satu alat yang paling umum yang kami gunakan untuk membuat dokumentasi piranti perangkat lunak.



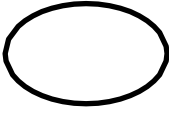



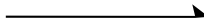

Gambar 2. 8 UML

1. Use Case Diagram

Diagram ini memperlihatkan himpunan usecase dan aktor-aktor, diagram ini juga sangat penting untuk mengorganisir dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan manusia.

Tabel 2. 1 Use Case Diagram

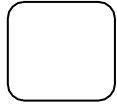





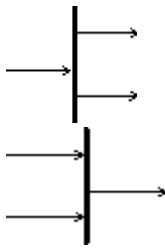

No	Simbol	Keterangan
----	--------	------------


No	Simbol	Keterangan
1.		<i>Use case</i> merupakan deskripsi fungsional yang telah disediakan oleh sistem sebagai entitas yang menghasilkan hasil yang terukur untuk suatu <i>actor</i> .
2.		<i>Actor</i> merupakan himpunan peran untuk berinteraksi dengan <i>usecase</i>
3.		<i>Association</i> merupakan garis yang menghubungkan objek satu dengan objek yang lain
4.	 <<include>>	<i>Include</i> merupakan gambaran jika <i>usecase</i> dipanggil oleh <i>usecase</i> lain
5.		<i>Dependecy</i> merupakan garis panah yang menunjukkan jika <i>actor</i> berinteraksi secara pasif
6.	 <<extends >>	<i>Extend</i> merupakan gambar jika memperluas <i>usecase</i> target.

2. Activity Diagram

Diagram ini adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dari suatu sistem. Diagram ini penting dalam pemodelan fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.

Tabel 2. 2 Activity Diagram

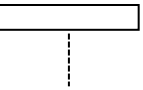
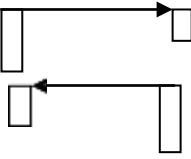

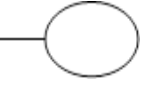
No	Gambar	Nama	Keterangan
1.		<i>Activity</i>	Menggambarkan cara masing-masing kelas antarmuka berinteraksi satu sama lain.
2.		<i>Action</i>	<i>State</i> sistem yang menunjukkan eksekusi tindakan
3.		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4.		<i>Final Node</i>	Bagaimana hal-hal dibentuk dan hancur
5.		<i>Fork Node</i>	Pada titik tertentu, satu aliran berkembang menjadi beberapa aliran.
6.		<i>Decision</i>	Pilihan untuk membuat pilihan
7		<i>Fork/Join</i>	digunakan untuk menunjukkan kegiatan yang dilakukan secara bersamaan atau menggabungkan dua kegiatan bersamaan.
8		<i>Rake</i>	menunjukkan bahwa ada fragmentasi.





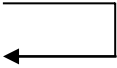
No	Gambar	Nama	Keterangan
9		<i>Send</i>	Tanda pengiriman.

3. Sequence Diagram

Diagram *sequence*, yang bersifat dinamis, menekankan pengiriman pesan dalam jangka waktu tertentu.

Tabel 2. 3 *Sequence Diagram*

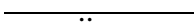
No	Gambar	Nama	Keterangan
1.		<i>LifeLine</i>	Objektif entitas, antarmuka interaktif.
2.		<i>Message</i>	Spesifikasi dari cara objek berkomunikasi satu sama lain yang mengandung informasi—informasi tentang aktifitas yang terjadi.
3		<i>Actor</i>	Menggambarkan individu yang berinteraksi dengan sistem.
4		<i>Boundary Class</i>	Menggambarkan gambar bentuk.

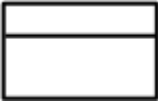
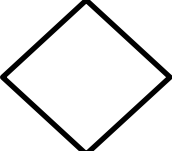




No	Gambar	Nama	Keterangan
5		<i>Entity</i> <i>Class</i>	Menjelaskan hubungan antara kegiatan yang akan dilakukan. .
6.		<i>Control</i> <i>Class</i>	menggambarkan cara batas berhubungan dengan tabel.
7		<i>Activation</i>	sebagai benda yang dapat melakukan sesuatu.
8	Message 	<i>Message</i>	Menunjukkan bahwa ada komunikasi antara objek dengan objek.
9		<i>Self</i> <i>Message</i>	Mengizinkan sebuah objek untuk berkomunikasi dengan orang lain kembali ke dalamnya

4. Class Diagram

Diagram kelas yang statis menampilkan himpunan kelas, antarmuka, kolaborasi, dan hubungan.

Tabel 2. 4 Class Diagram

No	Simbol	Keterangan
1.		<i>Generallization</i> , merupakan dimana objek <i>descendent</i> membagikan perilaku dan struktur data objek induknya

No	Simbol	Keterangan
2.		<p><i>Class</i>, adalah kumpulan objek yang saling berbagi.</p>
3.		<p><i>Nary Association</i>, digunakan untuk asosiasi terhadap dengan objek lainnya.</p>
4.		<p><i>Collaboration</i> merupakan deskripsi urutan aksi yang ditampilkan suatu sistem yang memiliki konsekuensi terukur bagi actor</p>
5.		<p>Merupakan operasi yang valid dilakukan oleh suatu objek</p>
6.		<p>Merupakan garis penah yang menunjukkan jika actor berinteraksi secara pasif</p>
7.		<p><i>Association</i> merupakan garis yang menghubungkan objek satu dengan objek yang lain</p>