

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Aplikasi *Blynk* digunakan dalam proyek penelitian “Rancang Bangun Sistem *Monitoring* Pemilahan Sampah Otomatis Berbasis *Internet of Things*” oleh Ginas Alvianingsih, Tri Wahyu Oktaviana Putri, dan Pratiwi Maharani pada tahun 2023. Masing-masing sensor *ultrasonik HC-SR04* ditempatkan di dalamnya. wadah limbah logam, organik, dan anorganik yang membentuk sistem pemantauan ini. *Mikrokontroler NodeMCU ESP8266* yang mampu terhubung ke internet bertanggung jawab atas sistem ini. Dengan perangkat yang menjalankan aplikasi *Blynk*, seseorang dapat mengawasi ketinggian tempat sampah dari lokasi mana pun. Kemampuan sistem dalam mengukur tinggi sampah secara *real-time* pada jarak 0 hingga 25 cm diketahui berdasarkan hasil pengujian. Selain itu, sistem pemantauan memiliki kemampuan untuk menyampaikan[5].

Sedangkan penelitian Yohanes Bowo Widodo, Tata Sutabri, dan Leo Faturahman pada tahun 2019 berfokus pada tempat sampah pintar dengan notifikasi berbasis IoT. Dalam penelitian ini, dikembangkan teknologi—tempat sampah pintar dengan sensor *ultrasonik* berbasis *mikrokontroler*—yang dapat digunakan dalam menjaga kebersihan. Ketinggian kotak sampah dapat dideteksi oleh WeMos D1 Mini. Tujuan dari proyek ini adalah untuk membuat prototipe kerja untuk sistem pendeteksi isi kotak sampah ultrasonik. Petugas sampah akan menerima email notifikasi dari sistem yang

memberitahukan mereka untuk mengambil sampah jika isi kotak sampah mencapai 80% atau lebih[6].

Agung Panji Sasmito, Stevania Hildegardis Bere, dan Ali Mahmudi melakukan penelitian pada tahun 2021 dan menerbitkannya di publikasi berbeda. Penelitian difokuskan pada pembuatan alat pembuka dan penutup tempat sampah otomatis yang memanfaatkan sensor jarak berbasis *Arduino*. Penelitian ini memanfaatkan teknologi masa kini, khususnya dengan membuat alat pembuka dan penutup kotak. Salah satu teknik pengelolaan sampah otomatis adalah dengan menggunakan beberapa sensor jarak ultrasonik bersama dengan pengontrol untuk mengoperasikan motor servo, yang membuka dan menutup kotak sampah[7].

Pada tahun 2020, Irawan Hadi, seorang sarjana, menerbitkan penelitiannya di jurnal bertajuk *Web Platform* sebagai Penampil Data Pemantauan Kotak Sampah Berbasis IOT. Dalam penelitian ini, kami membuat sistem pemantauan kotak sampah cerdas menggunakan integrasi perangkat keras *Arduino*, *loadcell*, *ultrasonik*, *GPS*, dan *SIM900A* dengan *platform web*. Sistem perangkat keras akan mengirimkan data pembacaan ke *database server* dari sensor *GPS*, *ultrasonik*, dan *loadcell*, yang masing-masing bertindak sebagai sensor berat dan jarak. Setelah itu, muncul peta yang menunjukkan lokasi kotak sampah dan tampilan data sensor pada antarmuka *platform web*. Manfaat sistem ini adalah petugas sampah dapat menentukan[8].

Penelitian yang dilakukan Yuli Febryanti, Fahrudin Mukti Wibowo, Anggi Zafia pada tahun 2021 yaitu Sistem Monitoring Tempat Sampah Pintar Di Pusat Penelitian Konservasi Tumbuhan Dan Kebun Raya-Lipi, Alat

yang dibuat terdiri dari sensor ultrasonic dan sensor Load Cell untuk mendeteksi kapasitas dan berat pada sampah. Data sensing kemudian dikirim ke Firebase dan dimonitoring menggunakan aplikasi android. Dari hasil pengujian rancang bangun sistem ini menghasilkan nilai rata-rata error pada sensor jarak tempat sampah pertama sebesar 8.95% dengan tingkat akurasi 91,05% dan pada tempat sampah kedua dengan error 10.66% dan tingkat akurasi 89,34% serta pada Load Cell tempat sampah pertama didapat hasil nilai error 3.98% dengan tingkat akurasi sensor 96.02%, dan tempat sampah kedua didapat hasil nilai error 4.56% dengan tingkat akurasi sensor 95.44% [9].

2.2 Landasan Teori

2.2.1 Sistem *Monitoring*

Menurut Mercy (2005), Sistem Pemantauan adalah siklus tindakan yang melibatkan pengumpulan, pemeriksaan, pelaporan, dan pengambilan tindakan yang tepat mengenai suatu proses yang sedang dipraktikkan. Pemantauan biasanya dilakukan untuk memverifikasi kinerja terhadap tujuan yang telah ditentukan. Hubungan antara pemantauan dan manajemen kinerja, suatu prosedur terpadu untuk menjamin bahwa proses berjalan sesuai rencana, juga dipertimbangkan. Untuk memutuskan bagaimana melanjutkan perbaikan berkelanjutan, pemantauan mungkin memberikan informasi tentang kelangsungan proses. Ketika suatu

proses sedang dilaksanakan, pemantauan dilakukan pada saat itu[10].

2.2.2 Sampah Organik

Sampah organik merupakan sampah yang dihasilkan dari kegiatan pertanian, perikanan, dan kegiatan lainnya. Oleh karena itu, Sebagian besar sampah rumah tangga merupakan sampah organik. Seperti kulit buah, sisa sayuran, dan lain-lain[11].

2.2.3 Sampah Anorganik

Sampah anorganik atau sering disebut dengan sampah kering adalah sampah yang tidak mudah membusuk dan sangat susah terurai oleh alam, sehingga jika jumlah sampah tersebut menumpuk dalam tanah maka akan mengakibatkan pencemaran tanah dan lingkungan. Contoh dari sampah anorganik adalah plastik, kertas, kaleng, botol, seng, logam, besi dan bahan lainnya[12].

2.2.4 *Internet Of Things*

Istilah "*Internet of Things*" mengacu pada gagasan bahwa segala sesuatu yang memiliki akses internet dapat berkomunikasi dengan benda-benda terdekat dan satu sama lain. *Internet of Things* (IoT) secara luas dianggap sebagai "*the next big thing*" di era digital. Hal ini disebabkan besarnya potensi yang dimiliki oleh teknologi *Internet of Things* (IoT)[13].

2.2.5 Website

Website adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam *World Wide Web* (WWW) di Internet. Sebuah halaman *web* adalah dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*), yang hampir selalu bisa diakses melalui HTTP, yaitu protokol yang menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser*[14].

2.2.6 My SQL



Gambar 2. 1 *My SQL*

Tipe *database* yang populer untuk membangun aplikasi berbasis *web* dinamis adalah *My SQL*. Salah satu jenis sistem manajemen basis data relasional (RDBMS) adalah *MySQL*. Bahasa pemrograman PHP didukung oleh *MySQL*. Selain itu, *MySQL* menawarkan kueri SQL (*Structured Query Language*) dasar yang menggunakan karakter *escape* yang sama seperti PHP[15].

2.2.7 Database

Basis data adalah susunan data terkait yang disimpan secara hati-hati secara berlebihan untuk mendukung satu atau lebih aplikasi dengan sebaik-baiknya. Data disimpan secara terpisah dari program yang menggunakannya, dan merupakan hal yang biasa dan diatur untuk menambah, mengubah, dan mengambil data yang sudah ada[16].

2.2.8 PHP (*Personal Home Page*)



Gambar 2. 2 PHP (*Personal Home Page*)

Personal Home Pages, atau PHP Salah satu jenis bahasa pemrograman adalah PHP. PHP dirancang terutama untuk pengembangan aplikasi *web*. Bahasa pemrograman yang paling banyak dicari di seluruh dunia oleh programmer *web* adalah PHP. Ini karena bahasa pemrograman ini bersifat *open source* dan biasanya mudah dipelajari[17].

2.2.9 Hosting

Hosting merupakan tempat penyimpanan data *website* dimana didalamnya meliputi kapasitas penyimpanan, *bandwith* yang merupakan sebuah kapasitas yang di gunakan untuk mengukur jumlah pengunjung *website* serta *database*[18].

2.2.10 *Visual Studio Code*



Gambar 2. 3 *Visual Studio Code*

Microsoft membuat editor kode sumber terbuka *Visual Studio Code* untuk sistem operasi *Windows*, *Linux*, dan *MacOS*. Penulisan kode yang mendukung banyak bahasa pemrograman, termasuk *C++*, *C#*, *Java*, *Python*, *PHP*, dan *GO*, menjadi sederhana dengan *Visual Code*. *Visual Code* dapat menentukan jenis bahasa pemrograman yang digunakan dan dapat mengubah warna suatu segmen kode berdasarkan fungsinya. *Github* juga telah dikaitkan dengan *Visual Studio Code*. Selain itu, pengembang dapat menambahkan ekstensi untuk menambah fungsionalitas yang tidak disertakan dalam *Visual Studio Code*. Ini adalah fitur lainnya[19].

2.2.11 *Bootstrap*



Gambar 2. 4 Bootstrap

Bootstrap merupakan sebuah *framework* yang dapat menyelesaikan permasalahan dalam mendesain *web*. Slogan dari *framework* ini adalah “*Sleek, intuitive, and powerful front-end framework for faster and easier web development*”, yang berarti kita dapat mendesain sebuah *website* dengan lebih rapi, cepat dan mudah[20].

2.2.12 Xampp



Gambar 2. 5 Xampp

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP *Server*, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan *Perl*[21].


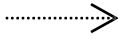

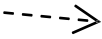
2.2.13 UML (*Unified Modeling Language*)







Teknik pemodelan visual yang disebut UML (*Unified Modeling Language*) digunakan untuk merancang dan/atau membuat perangkat lunak berorientasi objek. Semua diagram dan elemen dalam UML didasarkan pada paradigma berorientasi objek karena merupakan bahasa *visual* untuk pemodelan berorientasi objek. UML sendiri menawarkan pedoman untuk membuat sistem cetak biru yang terdiri dari konsep proses bisnis dan kelas bahasa pemrograman. Penjelasan berikut ini menggunakan beberapa diagram UML (*Unified Modeling Language*) sebagai contoh :

1. Use Case Diagram

Kasus Pemanfaatan Fungsionalitas yang diharapkan dari suatu sistem dijelaskan dalam diagram. Ini bukan tentang "bagaimana" tetapi tentang "apa" yang dilakukan Sistem. Interaksi aktor-sistem diwakili oleh *use case*. *Use case* adalah tugas tertentu, seperti membuat daftar belanjaan atau login ke sistem. Suatu entitas, baik manusia atau buatan, yang berkomunikasi dengan sistem untuk melakukan aktivitas tertentu disebut aktor atau orang. Entitas manusia atau mekanik yang berkomunikasi dengan sistem untuk menjalankan fungsi tertentu.

Tabel 2. 1 Use Case Diagram

No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<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 (<i>independent</i>).
3.		<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>).
4.		<i>Include</i>	Menspesifikasikan bahwa use case sumber secara eksplisit.

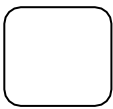




No	Gambar	Nama	Keterangan
5.		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7.		<i>System</i>	Menspesifikasikan paket yang menampilkan Sistem secara terbatas.
8.		<i>Use Case</i>	Deskripsi dari urutan aksi – aksi yang ditampilkan Sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
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.


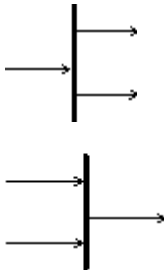

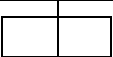

2. Activity Diagram

Activity Diagram menunjukkan aliran aktivitas berbeda di dalam sistem yang sedang dirancang, beserta titik awal, hasil potensial, dan akhir. Diagram aktivitas juga dapat menunjukkan proses yang berjalan secara bersamaan dalam banyak eksekusi. Akibatnya, diagram aktivitas umumnya mewakili proses dan jalur aktivitas dari tingkat atas daripada menggambarkan secara tepat perilaku mendasar suatu sistem (termasuk interaksi antar

subsistem). *Use case* dapat digunakan untuk merealisasikan suatu aktivitas. Kasus penggunaan merinci bagaimana aktor menggunakan sistem untuk melaksanakan aktivitas, sedangkan aktivitas menjelaskan proses yang sedang berlangsung. Keputusan digunakan untuk mengkarakterisasi tindakan dalam skenario tertentu. Titik sinkronisasi, yang dapat berupa titik atau garis *vertikal* atau *horizontal*, digunakan untuk menunjukkan proses paralel (percabangan dan penggabungan) di dalam.

Tabel 2. 2 Keterangan *Activity Diagram*

No	Gambar	Nama	Keterangan
1.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2.		<i>Action</i>	<i>State</i> dari Sistem yang mencerminkan eksekusi suatu aksi.
3.		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4.		<i>Final Node</i>	Bagaimana objek dan dihancurkan.
5.		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

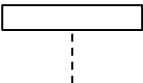
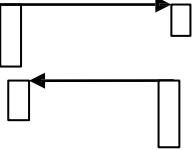



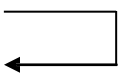
No	Gambar	Nama	Keterangan
6.		<i>Decision</i>	Pilihan untuk mengambil keputusan.
7		<i>Fork/Join</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
8		<i>Rake</i>	Menunjukkan adanya dekomposisi.
9		<i>Time</i>	Tanda waktu.
10		<i>Send</i>	Tanda pengiriman.

3. *Sequence Diagram*

Sequence Diagram menunjukkan interaksi antar objek di sekitarnya, seperti tampilan, pengguna, dan sebagainya, sebagai pesan yang ditampilkan sepanjang waktu. Dua dimensi diagram sekuen adalah objek yang terhubung pada *horizontal* dan waktu pada *vertikal*. Diagram urutan biasanya digunakan untuk menggambarkan situasi atau serangkaian tindakan yang diambil

sebagai reaksi terhadap suatu kejadian untuk menghasilkan hasil tertentu. diawali dengan peristiwa yang menjadi pemicu kegiatan, prosedur internal dan modifikasinya, serta produk akhir. Semua objek mempunyai garis hidup *vertikal*, termasuk pelakunya.

Tabel 2. 3 Keterangan *Sequence Diagram*

No	Gambar	Nama	Keterangan
1.		<i>LifeLine</i>	Objek <i>entity</i> , antar muka yang saling berinteraksi.
2.		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan Sistem.
4		<i>Activation</i>	Sebagai sebuah objek yang akan melakukan sebuah aksi.
5	Message 	<i>Message</i>	Mengindikasikan komunikasi antara objek dengan objek.
6		<i>Self Message</i>	Menginndikasikan komunikasi kembali kedalam sebuah objek itu sendiri.