

BAB 2

DASAR TEORI

2.1 KAJIAN PUSTAKA

Penelitian pertama oleh Muhammad Arif Maulana Nabil yang berjudul “Kotak Sampah Pintar Menggunakan Sensor Ultrasonik Berbasis Mikrokontroler Arduino Uno”. Pada sistem ini peneliti menggunakan sensor ultrasonik untuk mendeteksi ketinggian sampah yang kemudian akan diproses pada mikrokontroler *Arduino Uno R3*, melalui modul GSM Sim800L untuk mengirimkan pemberitahuan berupa SMS apabila tempat sampah penuh [3].

Penelitian kedua oleh Muhammad Mukrim Al Maburr (2016) yang berjudul “Rancang Bangun Sistem Smart Trash Can Berbasis Android”. Pada sistem ini peneliti menggunakan sensor ultrasonik dan sensor berat untuk mendeteksi kapasitas sampah, GPS untuk mengetahui posisi tempat sampah yang kemudian akan diproses pada mikrokontroler *Arduino Uno R3*. Dengan modul GSM SIM 900 yang akan mengirimkan pemberitahuan ke *web server* apabila tempat sampah penuh dan menampilkan posisi tempat sampah [4].

Penelitian ketiga diambil dari jurnal oleh Muhamad Syaifudin, Faqih Rofi, Anis Qustoniah yang berjudul “Rancang Bangun Sistem Monitoring Tempat Sampah Rumah Tangga dan Penerangan Jalan Berbasis Wireless Sensor Network (WSN)”. Pada sistem ini peneliti memanfaatkan sensor ultrasonik untuk mendeteksi ketinggian sampah dan *photodiode* untuk mendeteksi perubahan cahaya pada lampu penerangan jalan yang kemudian diproses pada mikrokontroler *Arduino Uno R3*. Sistem ini menggunakan konsep *Wireless Sensor Network* (WSN) dengan memanfaatkan *Xbee* untuk mengirimkan informasi dari *transmitter* jika terjadi perubahan pada ketinggian tempat sampah dan lampu penerangan jalan ke ruang petugas (*receiver*) [5].

Penelitian keempat diambil dari jurnal oleh Yudha Elasya, Didik Notosudjono, Evyta Wismiana yang berjudul “Aplikasi Sensor Ultrasonik Berbasis Mikrokontroler Atmega328 Untuk Merancang Tempat Sampah Pintar”. Pada sistem ini peneliti menggunakan sensor ultrasonik untuk mendeteksi

ketinggian sampah yang kemudian akan diproses pada mikrokontroler *Arduino Uno R3*, melalui modul GSM Sim800L untuk mengirimkan pemberitahuan berupa SMS dan motor DC akan mengeluarkan bak sampah apabila tempat sampah penuh [6].

Penelitian kelima diambil dari jurnal oleh Almuchlisin, Agung Nugroho, Umar All Ahmad yang berjudul “Perancangan Dan Implementasi Sistem Monitoring Pelaporan Tempat Sampah Berbasis Teknologi Embedded”. Pada sistem ini peneliti memanfaatkan sensor ultrasonik untuk mendeteksi ketinggian sampah yang kemudian diproses pada mikrokontroler *Arduino Uno R3* menggunakan konsep *Wireless Sensor Network* (WSN) dengan nRF agar dapat mengirim hasil ketinggian sampah dari setiap sensor yang terpasang pada TPS (Tempat Pembuangan Sampah), menggunakan modul GSM SIM 900 yang akan mengirimkan pemberitahuan ke *web server* apabila tempat sampah penuh [7].

Hubungan lima penelitian diatas dengan penelitian penulis yaitu pada alat yang dibuat, berupa tempat sampah yang digunakan untuk mengatasi masalah penumpukan sampah pada tempat sampah akibat dari pengangkutan sampah yang tidak teratur. Dari lima penelitian tersebut untuk dapat membuat tempat sampah yang dapat mengatasi masalah penumpukan sampah pada tempat sampah, diperlukan adanya tempat sampah pintar yang dapat memantau ketinggian sampah menggunakan sensor ultrasonik dan sensor berat yang berintegrasi dengan mikrokontroler *Arduino Uno R3*.

Sensor berat mempunyai kelemahan yaitu hanya bisa mendeteksi ketinggian sampah dari segi beratnya, sedangkan sampah pasti mempunyai berat yang berbeda untuk setiap sampah. Tidak mesti sampah dengan berat 5 kg mempunyai tinggi 20 cm, kemudian juga sebaliknya sampah dengan tinggi 20 cm belum tentu mempunyai berat 5 kg dengan ukuran tempat sampah yang sama [8]. Pada penelitian ini penulis menggunakan sensor ultrasonik untuk mendeteksi ketinggian tempat sampah karena mempunyai kelebihan dibanding dengan sensor berat. Kelebihannya yaitu sensor ultrasonik dapat mendeteksi jarak benda yang ada didepanya secara langsung, yaitu dengan jarak deteksi minimal 2 cm dan maksimal 4 m [9].

Istilah *Internet of Things* pertama kali disarankan oleh Kevin Ashton pada tahun 1999. Sistem *Internet of Things* memungkinkan pengguna untuk bisa mencapai lebih dalam terhadap otomisasi, analisis, dan integrasi sistem. *Internet of Things* memanfaatkan teknologi penginderaan, jaringan, dan robotika yang sudah ada. Dengan adanya *Internet of Things* akan mempermudah kegiatan manusia dalam melakukan berbagai aktifitas sehari-hari. Semua kegiatan dapat dilakukan dengan sangat praktis dan dengan adanya sistem kontrol dengan perangkat yang terhubung menyebabkan kehidupan akan lebih praktis dan efisien [11].

2.2.2 PLATFORM IOT (INTERNET OF THINGS)

Platform IoT adalah perangkat lunak pendukung yang menghubungkan segala sesuatu dalam sistem *IoT*. *Platform IoT* memfasilitasi komunikasi, aliran data, manajemen perangkat, dan fungsionalitas aplikasi. Dengan semua jenis perangkat keras dan opsi konektivitas yang berbeda, perlu adanya cara untuk membuat semuanya bekerja sama. *Platform IoT* mengatasi masalah tersebut.

Fungsi *Platform IoT* :

1. Menghubungkan perangkat keras, seperti sensor dan perangkat.
2. Menangani berbagai protokol komunikasi perangkat keras dan perangkat lunak.
3. Memberikan keamanan dan otentikasi untuk perangkat dengan pengguna.
4. Mengumpulkan, memvisualisasikan, dan menganalisis data yang dikumpulkan sensor dan perangkat [12].

Salah satu *Platform IoT* yang bisa dimanfaatkan untuk mendukung sistem *IoT* yaitu IFTTT (*If This Then That*). IFTTT merupakan sebuah aplikasi yang dapat menghubungkan dua aplikasi atau *web* menjadi satu. Aplikasi ini mempunyai visi untuk memungkinkan data digital seperti data fisik, dimana pengguna dapat menggabungkan beberapa hal untuk membuat hal baru dengan mudah. Setiap aktifitas yang akan dilakukan secara *online* dapat diatur untuk memiliki reaksi otomatis pada sebuah aplikasi *web* tertentu. IFTTT merupakan rantai yang dapat menggabungkan beberapa aplikasi *web* sehingga dapat saling menunjang. Untuk dapat bekerja IFTTT membutuhkan pemicu (*trigger*) dan

reaksi (*action*). Untuk dapat menggunakan IFTTT maka klik “this” untuk memilih program yang ingin digunakan sebagai *trigger*, setelah itu klik “that” untuk memilih program yang digunakan sebagai *action* [13].

Aplikasi pemicu (*trigger*) yang dapat digunakan di IFTTT saat ini yaitu *Webhooks*. *Webhooks* merupakan cara bagi suatu aplikasi untuk menyediakan aplikasi lain dengan informasi *real time*. *Webhooks* adalah *link URL* yang ditambahkan agar data yang dikirim dapat langsung diterima di waktu sama dengan *link URL* yang sudah ditentukan. *Webhooks* juga bisa disebut sebagai “Reverse API” karena harus merancang API agar *Webhooks* dapat digunakan. Untuk dapat menggunakan *Webhooks* maka berikan URL kepada *provider Webhooks* untuk mengirimkan permintaan. Kemudian atur URL di aplikasi yang digunakan agar dapat diakses dari *web publik* [14].

Pada IFTTT terdapat banyak aplikasi yang dapat digunakan sebagai reaksi (*action*), sebagai contoh yaitu aplikasi *Line* dan *Facebook Messenger*. *Line* merupakan suatu aplikasi yang digunakan untuk kegiatan berkiriman pesan (*messenger*) secara gratis diperangkat *smartphone*. Aplikasi *line* juga bisa disebut sebagai aplikasi jejaring sosial karena mempunyai fitur *timeline* sebagai wadah untuk berbagi status, pesan suara, *video* foto, kontak, dan informasi lokasi. Dengan aplikasi *line* bisa melakukan *voice call* maupun *video call* secara *real time* dan gratis. *Line* disediakan di semua perangkat *smartphone* dan di semua *operasi mobile* seperti *Android, iOS, Windows Phone, Blackberry*, dan *PC* [15].

Kemudian *Facebook Messenger* merupakan aplikasi resmi dari *Facebook* untuk melakukan percakapan teks dengan orang lain di jejaring social. *Facebook Messenger* dapat digunakan untuk berbagi gambar dan lokasi, dalam pesan teks bisa menambahkan beberapa penerima dan membuka jendela obrolan dengan beberapa orang secara bersamaan. Setiap percakapan kemudian disimpan dalam gelembung yang kemudian bisa dipindahkan kemanapun dalam layar *smartphone*. *Facebook messenger* juga bisa menerima tanda suara dan getar setiap kali menerima pesan [16].

2.2.3 EMBEDDED SYSTEM

Merupakan sistem komputer yang secara khusus dirancang untuk menjalankan tugas tertentu, biasanya tertanam pada satu kesatuan sistem. Sistem ini menjadi bagian dari seluruh sistem yang terdiri dari mekanik dan perangkat keras lainnya. Sistem *Embedded* merupakan sebuah sistem yang lebih besar, yang biasanya bukan berupa sistem elektronika. Kata *Embedded* menunjukkan bagian yang tidak dapat berdiri sendiri. *Embedded System* biasanya diimplementasikan dengan menggunakan mikrokontroler, sistem ini dapat memberikan respon yang sifatnya *real time* dan banyak digunakan pada peralatan digital.

Mikrokontroler merupakan sebuah komputer kecil didalam satu sirkuit yang berisi inti prosesor, memori, *input* atau *output peripheral* yang dapat diprogram, dan didesain khusus untuk *Embedded System*. Mikrokontroler berukuran sangat kecil berbeda dengan komputer yang besar. Hal ini dikarenakan mikrokontroler didesain untuk suatu tujuan khusus, bukan untuk melakukan pekerjaan rumit seperti yang ada pada komputer. *Embedded System* merupakan sistem dengan ciri-ciri sebagai berikut :

1. Mempunyai *computing power* atau dilengkapi dengan sebuah *processor*.
2. Mempunyai tugas yang spesifik, berbeda dengan komputer atau *server* yang relatif lebih *multi purpose*.

Kategori sistem *Embedded* :

1. Mandiri (*Standalone*)

Perangkat ini dapat berfungsi secara independen dari perangkat keras lainnya. Tidak terintegrasi dengan perangkat lain.

2. *Real Time*

Sistem *Embedded* dengan tugas-tugas spesifik yang dilakukan dalam periode waktu spesifik disebut dengan sistem *Real Time*. Sistem ini terdiri dari *Hard Real Time* dan *Soft Real Time*.

3. *Hard Real Time*

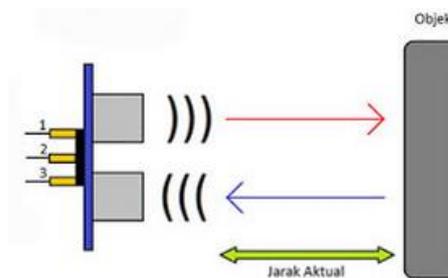
Merupakan sistem yang harus melaksanakan tugas dengan *deadline* yang tepat. Sistem ini sering digunakan sebagai alat pengontrol untuk aplikasi yang *dedicated*, mempunyai batasan waktu yang tetap yang sudah didefinisikan

dengan baik. Pemrosesan harus selesai dalam batasan-batasan yang telah didefinisikan, jika tidak sistem akan gagal.

4. *Soft Real Time*

Merupakan sistem yang tidak memerlukan *deadline*. Seperti pada *DVD player*, jika diberikan suatu perintah dari *remote control* maka akan mengalami *delay* selama beberapa milidetik untuk dapat menjalankan perintah tersebut. *Delay* ini tidak akan berakibat sesuatuyang serius karena sistem *Soft Real Time* mempunyai lebih sedikit batasan waktu yang keras dan tidak mendukung *deadline* dengan batas akhir [17].

2.2.4 SENSOR ULTRASONIK (HC-SR04)



Gambar 2.2 Cara Kerja Sensor Ultrasonik dengan Penyulut dan Receiver

Sensor ultrasonik bekerja dengan membangkitkan gelombang ultrasonik melalui sebuah *pin* yang disebut *Trigger* atau penyulut yang akan menghasilkan gelombang ultrasonik dengan frekuensi 40 KHz. Setelah gelombang ultrasonik menyentuh permukaan target, maka target akan memantulkan kembali gelombang ultrasonik tersebut dengan sudut pantul 15 derajat. Gelombang pantulan dari target akan ditangkap oleh *pin Echo* atau *Receiver*, kemudian gelombang pantulan tersebut akan akan diproses untuk menghitung jarak benda tersebut dengan rumus :

$$S = \frac{t}{2} \times 343 \quad (2.1)$$

Dimana S merupakan jarak antara sensor ultrasonik dengan benda pantul, t merupakan waktu yang dibutuhkan untuk perjalanan gelombang ultrasonik dari

Trigger menuju ke benda dan kembali ke *Receiver*. Maka kemudian jarak target akan dapat diketahui oleh sensor ultrasonik.



Gambar 2.3 Sensor Ultrasonik (HC-SR04) [18]

2.2.5 NTP (*NETWORK TIME PROTOCOL*)

NTP adalah sebuah protocol yang digunakan untuk pensinkronan waktu (menyamakan waktu) dalam sebuah jaringan, baik LAN atau jaringan *internet*. Protokol NTP berjalan pada *port* komunikasi UDP nomor 123. Protokol ini didesain untuk dapat bekerja dengan baik meski media komunikasinya bervariasi, mulai dari waktu yang latensinya tinggi hingga rendah, mulai dari media kabel sampai udara. Proses penyamaan waktu yang dilakukan oleh NTP sangat akurat. NTP berjalan pada dua sisi yaitu *Server* dan *Client*. Di *server* NTP berfungsi sebagai *server* yang menyimpan penanggalan waktu yang akurat, dan nantinya akan di *request* oleh NTP *Client*. NTP *Server* tidak berdiri sendiri, tetapi saling bersinkronisasi dengan NTP *Server* lainya [19].

2.2.6 STANDAR DEVIASI

Standar deviasi merupakan nilai statistik yang dimanfaatkan untuk menentukan bagaimana sebaran data dalam sampel, serta seberapa dekat titik data individu ke rata-rata nilai sampel. Sebuah standar deviasi dari kumpulan data sama dengan nol menandakan bahwa semua nilai pada himpunan tersebut sama. Sedangkan nilai deviasi yang besar menunjukkan bahwa titik data individu jauh dari nilai rata-rata. Untuk menghitung varian dan standar deviasi digunakan rumus :

$$s^2 = \frac{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}{n(n-1)} \quad (2.2)$$

$$s = \sqrt{\frac{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}{n(n-1)}} \quad (2.3)$$

Penjelasan lebih jelasnya untuk menghitung standar deviasi yaitu, pertama yang dilakukan yaitu menghitung nilai rata-rata dari semua titik data terlebih dahulu. Cara menghitung rata-rata yaitu jumlah dari semua nilai dalam kumpulan data dibagi dengan jumlah total titik data tersebut. Selanjutnya menghitung penyimpangan setiap titik data dari rata-rata. Yaitu dengan mengurangkan nilai dari nilai rata-rata. Deviasi setiap titik data akan dikuadratkan dan dicari penyimpangan kuadrat individu rata-rata. Lalu nilai yang dihasilkan disebut varian. Sedangkan akar kuadrat dari varian adalah standar deviasi [20].

2.2.7 THROUGHPUT

Throughput berhubungan dengan *Bandwidth*. *Bandwidth* adalah besar maksimal jalan yang dapat dilalui oleh data, sedangkan *Throughput* adalah data sebenarnya yang mengalir pada media transmisi. Misalkan menggunakan internet dengan *Bandwidth* 5 Mbps dan ketika mendownload suatu file dari *internet* kecepatan menunjukkan 4,2 Mbps, kecepatan ini yang disebut *Throughput*. Untuk menghitung *Throughput* digunakan rumus :

$$\textit{Throughput} = \frac{\text{Jumlah data yang dikirim}}{\text{waktu pengiriman data}} \quad (2.4)$$

Bisa dipahami bahwa *Bandwidth* adalah kecepatan maksimum dalam transfer data, sedangkan *Throughput* adalah kecepatan sebenarnya yang terjadi pada saat transfer data [21].