

## BAB II TINJAUAN PUSTAKA

Bab ini menyajikan tinjauan pustaka yang bertujuan untuk memberikan landasan teoritis dan latar belakang ilmiah yang mendasari penelitian yang dilakukan. Tinjauan pustaka meliputi tiga komponen utama, yaitu penelitian terdahulu, landasan teori, dan literatur *review* yang relevan dengan topik yang diangkat dalam penelitian ini.

### 2.1 Penelitian Terdahulu

Penelitian mengenai sistem penyiraman otomatis berbasis IoT yang memanfaatkan algoritma *fuzzy logic* dengan monitoring melalui aplikasi *mobile* telah banyak dilakukan oleh berbagai peneliti. Ringkasan penelitian terdahulu digunakan untuk memahami sejauh mana perkembangan dan pencapaian yang telah dicapai dalam bidang tersebut.

**Tabel 2. 1 Penelitian Terdahulu**

Nama Peneliti	Pembahasan
(Sandi & Fatma, 2023)	Tujuan penelitian ini yaitu untuk mengetahui apa saja pemanfaatan IoT pada salah satu sector ekonomi di Indonesia yaitu pertanian. Sektor pertanian merupakan sumber penghasilan yang cukup memberikan dampak besar bagi perekonomian di Indonesia, contohnya seperti padi, jagung, kelapa sawit, lada, kopi, teh dan lain lain.
(Walid & Fikri, 2022)	Tujuan Penelitian ini yaitu untuk mengembangkan sistem manajemen irigasi berbasis IoT dengan fokus pada monitoring, kontrol, dan manajemen jaringan komunikasi untuk efisiensi dan jangkauan luas. Sistem dilengkapi antarmuka mobile yang user-friendly serta penyimpanan data real-time yang divisualisasikan. Menggunakan metode waterfall, sistem mencatat waktu update data rata-rata 1,8 detik pada aplikasi Blynk dan 6,4 detik pada Thingspeak, dengan selisih 4,6 detik yang tidak signifikan.
(Brigida Helvia Vien, 2023)	Tujuan Penelitian ini untuk mengembangkan sistem irigasi berbasis IoT dengan monitoring, kontrol, dan manajemen jaringan untuk komunikasi yang efisien. Sistem dilengkapi antarmuka mobile dan penyimpanan data real-time yang divisualisasikan. Menggunakan metode waterfall, sistem mencatat waktu update rata-rata 1,8 detik pada Blynk dan 6,4 detik pada Thingspeak.
(Naufal, 2022)	Tujuan penelitian ini yaitu membuat alat pemantau ini menggunakan water flow sensor untuk mengukur debit air dan viscosity sensor untuk membaca kekeruhan. ESP32 TTGO T-Display sebagai mikrokontroler terhubung Wi-Fi

Nama Peneliti	Pembahasan
	menampilkan data real-time melalui Telegram. Uji coba pada greenhouse menunjukkan sistem dapat memantau aliran dan jumlah air dengan kekeruhan 3,29–8,66 NTU. Rata-rata penggunaan air harian tercatat 11,25 liter.
(Adinda, 2023)	Penyiraman tanaman otomatis berbasis IoT dengan NodeMCU ESP8266 mempermudah penyiraman dan aktivitas pertanian. Sistem ini berfungsi real-time, mengontrol penyiraman otomatis, dan menampilkan 3 status pompa air berdasarkan sensor kelembapan tanah.
(Fadhilah dkk., 2021)	Sistem berbasis ESP32 ini menggunakan sensor YL-69 dan DHT22 untuk mengukur kelembapan tanah, dengan hasil dikirim ke smartphone via WhatsApp. Keypad memungkinkan pengaturan kelembapan sesuai jenis tanaman. Saat tanah kering, ESP32 mengaktifkan relay untuk menyalakan pompa air.

## 2.2 Landasan Dasar Teori

Kerangka teoritis dan topik-topik utama akan dibahas dalam bagian ini. Pemahaman yang lebih baik tentang subjek yang diteliti dan penguatan analisis dan interpretasi data merupakan hasil dari penjelasan teori-teori yang relevan.

### 2.2.1 Penyiraman

Dalam hal perawatan tanaman, penyiraman merupakan komponen penting untuk pertumbuhan tanaman yang sehat. Laju pertumbuhan tanaman sangat dipengaruhi oleh kebutuhan air yang cukup (Putra dkk., 2022).

### 2.2.2 Kelembapan Tanah

Tanah dianggap lembab atau basah jika memiliki kadar kelembapan tinggi. Persentase kelembapan yang dianggap lembab dapat bervariasi, tergantung pada jenis tanah, tanaman yang ditanam, dan kondisi iklim. Secara umum, tanah dianggap lembab jika kadar kelembapan berada di kisaran 50% hingga 70% dari kapasitas maksimum tanah untuk menahan air. Kadar kelembapan di atas 70% mungkin menunjukkan bahwa tanah sudah jenuh air dan terlalu basah, sementara kadar di bawah 50% dapat mengindikasikan bahwa tanah mulai mengering.

Perkembangan tanaman sangat rentan terhadap kekurangan air. Dalam pengelolaan air, ada sejumlah faktor penting yang perlu dipikirkan. Salah satu

faktor yang harus dipertimbangkan dalam sistem air adalah jenis tanah. Pengelolaan air yang tepat dapat membantu menjaga keseimbangan kadar air tanah, sehingga pupuk kandang dapat terurai. Untuk perkembangan optimal, kadar air tanah tidak boleh kurang dari 60–70% dari batas lahan oleh karena itu, sebagian besar lahan memerlukan sistem air tambahan (Sirait, 2020).

### **2.2.3 Pohon Pinus**

Indonesia hanya memiliki satu spesies pohon pinus-pinus. Dalam bahasa ilmiah pohon pinus dikenal sebagai pinus merkusi, pohon ini telah lama digunakan dalam kegiatan reboisasi di Indonesia. Reboisasi hutan yang gundul, hutan lindung, dan hutan produk merupakan tujuan utama penanaman hutan pinus. Hutan pinus, yang ditanam secara luas di Jawa sejak masa penjajahan Belanda, telah memicu perdebatan mengenai apakah hutan tersebut secara efektif mengatur kadar air (Indrajaya & Handayani, 2008).

Pendapat pertama mengatakan bahwa pinus dapat mengatasi kekeringan di musim kemarau karena pinus mengonsumsi banyak air. Pendapat kedua mengatakan pinus dapat menyimpan air di musim penghujan dan digunakan pada musim kemarau. Pujiharta (2005) menyarankan agar pinus ditanam pada daerah dengan curah hujan di atas 3.000 mm/tahun sehingga tidak perlu dikhawatirkan terjadi kekeringan atau kehilangan ketersediaan air tanah pada musim kemarau akibat konsumsi air yang tinggi oleh pinus. Tulisan ini menguraikan peranan hutan pinus dalam pengendalian fenomena tanah longsor secara vegetatif. Pinus secara genetis memiliki potensi untuk dikembangkan sebagai tanaman pengendali tanah longsor karena memiliki beberapa kelebihan, antara lain, perakaran yang dalam, intersepsi dan evapotranspirasi yang tinggi, pohonnya tidak terlalu berat atau ringan, dan produk utama yang bukan berupa kayu (Pudjiharta, 2005).

Konstruksi rumah, perabotan, wadah, korek api, pulp, tiang listrik, wol kayu, kayu lapis, resin, dan gondorukem adalah beberapa dari sekian banyak kegunaan pinus. Getah yang telah diolah memiliki beberapa kegunaan kosmetik, perekat, sabun, gondorukem, dan cat (Fadilah dkk., 2022). Pohon pinus memiliki kebutuhan nutrisi antara 850-1680 PPM. Tanaman Pinus disajikan pada Gambar 2.1



**Gambar 2. 1 Pohon Pinus**

## **2.3 Perangkat Keras**

Sistem penyiraman otomatis *Internet of things* (IoT) ini mengandalkan sejumlah perangkat keras utama yang bekerja sama untuk menjalankan algoritma *kontrol fuzzy logic*, yang dikendalikan oleh aplikasi *mobile*.

### **2.3.1 Sensor DHT22**



**Gambar 2. 2 Sensor DHT22**

**Sumber: (Souqy, 2018)**

Kelembaban dan suhu merupakan dua variabel yang berbeda namun saling terkait. Beberapa sistem menggunakan sensor terpisah untuk merekam suhu dan kelembaban. Namun, DHT22 dapat merekam kedua variabel secara bersamaan sehingga menyederhanakan proses pengukuran (Alfanugraha, 2022).

Salah satu sensor digital yang dapat mendeteksi kondisi lingkungan seperti kelembaban dan suhu adalah DHT22, seperti yang terlihat pada Gambar 2.2. Tingkat stabilitas yang tinggi, respons pembacaan data yang cepat, kemampuan

anti-interferensi, dan karakteristik kalibrasi yang presisi semuanya berkontribusi pada kualitas tinggi sensor ini. Sensor ini mungkin kecil, tetapi dapat mengirimkan sinyal hingga 20 meter (Alfanugraha, 2022). Ketika dipasangkan dengan Arduino atau jenis *mikrokontroler* lainnya, sensor DHT22 bekerja dengan baik. Sensor DHT22 didefinisikan oleh fitur-fitur berikut:

1. *Supply voltage* : +5V
2. *Temperatur range* : -40°C to 80°C (*Error* ± 0.5°C)
3. *Humidity* : 0-100% RH (*Error* ± 2% RH)
4. *Interface* : digital
5. Kabel konektor : 3 Pin

### 2.3.2 Sensor *Soil moisture*

Sensor *soil moisture* adalah modul yang digunakan untuk mendeteksi tingkat kelembapan tanah serta mengetahui keberadaan kandungan air di sekitar sensor. Penggunaannya cukup sederhana yaitu sensor dimasukkan ke dalam tanah, kemudian sensitivitasnya dapat diatur menggunakan potensiometer. Sensor ini menghasilkan keluaran berupa nilai 1 atau 0, tergantung pada tingkat kelembapan tanah-nilai ini dapat disesuaikan dengan pengaturan potensiometer. Informasi lebih lanjut mengenai spesifikasi dan koneksi sensor ini ditampilkan pada Gambar 2.3.

Sensor ini memiliki spesifikasi sebagai berikut:

1. *Comparator* menggunakan LM393
2. Hanya menggunakan 2 plat kecil sebagai sensor
3. *Supply* Tegangan 3.3-5 VDC
4. Digital *output* D0 dapat secara langsung dikoneksikan dengan MCU dengan mudah.

Selain itu, sensor ini sangat cocok untuk digunakan dalam aplikasi berbasis *mikrokontroler*. Dengan sensitivitas yang dapat diatur, sensor ini memberikan fleksibilitas dalam menyesuaikan tingkat kelembapan yang ingin dipantau. Fitur ini menjadikannya ideal untuk berbagai aplikasi, terutama dalam sistem irigasi otomatis atau pemantauan tanaman.



**Gambar 2. 3 Sensor *Soil moisture***

### 2.3.3 *Real Time Clock* (RTC DS3231)

*Real-time Clock* (RTC) adalah chip yang berfungsi untuk menyimpan waktu dan tanggal. DS3231 merupakan salah satu contoh RTC, modul ini menggunakan serial I2C sebagai atarmukanya dan dilengkapi sinyal keluaran gelombang kotak yang dapat diprogram. DS3231 memiliki sejumlah fitur unggulan, seperti deteksi otomatis saat terjadi kegagalan daya dan rangkaian switch untuk mempertahankan data waktu meskipun listrik padam. Dalam mode baterai cadangan, chip ini hanya membutuhkan daya kurang dari 500nA, sambil tetap menjaga osilatornya aktif. DS3231 juga tahan terhadap suhu ekstrim, mulai dari  $-40^{\circ}\text{C}$  hingga  $+85^{\circ}\text{C}$ , dan tersedia dalam kemasan 8-pin DIP atau SOIC.



**Gambar 2. 4 *Real Time Clock* (RTC DS3231)  
(Sumber: [www.nyebarilmu.com](http://www.nyebarilmu.com))**

Berikut Penjelasan Pin-Pin Pada IC DS3231:

1. X1 dan X2 Pada DS3231 memiliki kristal internal yang sudah terintegrasi, sehingga tidak memerlukan pin X1 dan X2 untuk dihubungkan ke kristal eksternal seperti pada DS1307. Hal ini membuat DS3231 lebih akurat dan stabil tanpa memerlukan kristal tambahan.

2. Dalam hal pengoperasian fungsi waktu dan tanggal, VBAT seperti catu daya cadangan untuk RTC. Didukung oleh baterai Sel Litium atau sumber energi bentuk lain, tegangan yang diperlukan adalah 3V. Koneksi ke *Ground* diperlukan untuk pin ini setiap kali tidak digunakan. Pada suhu sekitar 25°C, RTC dapat berfungsi sebagai cadangan energi selama hampir satu dekade dengan kapasitas baterai 48mAh atau lebih tinggi.
3. GND merupakan *Ground* atau referensi tegangan rangkaian.
4. SDA adalah I/O untuk komunikasi serial I2C yang dilayani oleh pin ini. Sebuah resistor *pull-up* tambahan diperlukan untuk pin ini, karena merupakan *open drain*, mirip dengan DS1307.
5. SCL bersifat *open drain* dan bertindak sebagai *clock input* I2C, dibutuhkan resistor *pull-up* tambahan agar transfer data antarmuka serial tetap sinkron.
6. *SWQ/OUT* berfungsi sebagai penggerak keluaran untuk gelombang persegi. Saat dinyalakan, akan ada perubahan menjadi empat frekuensi gelombang persegi yang berbeda yaitu 1 Hz, 4 kHz, 8 kHz, dan 32 kHz. Pin ini memerlukan resistor *pull-up* eksternal karena sifatnya identik dengan pin SDA dan SCL. Gunakan VCC atau VBAT untuk mengoperasikannya.
7. Tegangan ini sebagian besar dipasok melalui VCC. Akses data dan pembacaan yang akurat bergantung pada sumber daya yang terhubung dengan benar. Akses data tidak dimungkinkan jika pasokan cadangan juga terhubung ke VCC tetapi nilai VCC lebih rendah dari VTP.

#### **2.3.4 Total Dissolved Solid (TDS)**

Secara sederhana, *Total Dissolved Solids* (TDS) adalah padatan terlarut yang ada didalam air, termasuk mineral, garam, logam, serta kation dan anion. Itu mencakup semua zat terlarut yang bukan molekul air murni (H<sub>2</sub>O). Sebagai aturan praktis, konsentrasi padatan terlarut sama dengan total kation dan anion dalam air.

TDS terukur dalam satuan *Parts per Million* (ppm) atau perbandingan rasio berat ion terhadap air. Benda-benda padat di dalam air tersebut berasal dari banyak sumber organik seperti lumpur, plankton, serta limbah industri dan kotoran. Sumber lainnya bisa berasal dan limbah rumah tangga, pestisida, dan banyak lainnya. Sedangkan, sumber anorganik berasal dari batuan dan udara yang mengandung

kalsium bikarbonat, nitrogen, besi fosfor, sulfur, dan mineral lain. Semua benda ini berbentuk garam, yang merupakan kandungannya perpaduan antara logam dan non logam. Garam-garam ini biasanya terlarut di dalam air dalam bentuk ion, yang merupakan partikel yang memiliki kandungan positif dan negatif. Air juga mengangkut logam seperti timah dan tembaga saat perjalanannya di dalam pipa distribusi air minum.

#### **2.3.5.1 TDS Meter**

*Total Dissolved Solid* (TDS) Meter adalah alat untuk mengetes jumlah zat padat yang terlarut dalam air. Hasil pengukuran TDS Meter mempunyai satuan *Part Per Million* (PPM) yaitu bagian per satu juta. PPM dapat diartikan suatu jumlah ion dalam suatu larutan. Sebagai contoh, Misalnya, satu juta ion natrium akan hadir dalam larutan dengan konsentrasi 1 PPM ion Na<sup>+</sup>. TDS sering digunakan untuk mengukur tingkat pencemaran air. Menurut World Health Organization (WHO) batasan zat padat terlarut pada air minum di bawah 1000 (Gaznur dkk., 2017).

#### **2.3.5.1 Prinsip dasar TDS**

Salah satu ukuran kualitas air adalah *Total Dissolved Solids* (TDS), yang dapat mengungkapkan konsentrasi kontaminan termasuk limbah organik dan logam berat dalam air.

#### **2.3.5.2 Elektroda *Total Dissolved Solid* (TDS)**

*Total Dissolved Solids* (TDS) merupakan matrik penting untuk mengevaluasi kualitas air dan efektivitas metode pemurnian. TDS mengukur konsentrasi partikel terlarut dalam cairan. Menggunakan elektroda baja tahan karat merupakan cara umum untuk menguji TDS. Elektroda ini mengukur konsentrasi partikel terlarut dengan mendeteksi konduktivitas larutan, seperti yang ditunjukkan pada Gambar 2.5.

Mengukur tingkat konduktivitas antara dua elektroda merupakan tugas elektroda TDS Meter. Ion dengan muatan positif (Na<sup>+</sup>, Mg<sup>++</sup>, H<sup>+</sup>, dll.) akan bermigrasi ke elektroda dengan muatan negatif, sedangkan ion dengan muatan negatif (Cl<sup>-</sup>, SO<sub>4</sub><sup>-</sup>, HCO<sub>3</sub><sup>-</sup>, dll.) akan bermigrasi ke elektroda dengan muatan



positif. Elektroda selanjutnya akan menentukan fluks ion total untuk menentukan TDS. Ion yang bergerak adalah satu-satunya yang dapat dideteksi dan diukur oleh elektroda.



**Gambar 2. 5 Elektroda TDS Meter**  
(Sumber: <https://www.elecdesignworks.com>)

### 2.3.5 Relay

Relay adalah perangkat elektromagnetik yang digunakan untuk mengoperasikan sejumlah kontaktor atau saklar elektronik. Perangkat ini dikendalikan oleh rangkaian elektronik lain dengan menggunakan energi listrik sebagai sumbernya. Berbeda dengan saklar manual yang dioperasikan secara langsung tanpa arus listrik, relay menggunakan elektromekanis sehingga memungkinkan pengendalian otomatis. Definisi dasar relay elektromekanis adalah perangkat yang menutup atau membuka kontak sakelar dengan menggunakan gaya elektromagnetik. Struktur dan prinsip kerja relay dijelaskan lebih lanjut pada Gambar 2.6.

Tiga komponen utama relay adalah:

1. *Common*, adalah bagian yang terhubung ke *Normally Close* ketika semuanya baik-baik saja.
2. *Coil* (kumparan), adalah bagian utama relay yang menghasilkan medan magnet.
3. Kontak, adalah bagian yang terdiri dari *Normally Close* dan *Normally Open*.

Pin Konfigurasi:

1. VCC : 5VDC
2. COM : 5VDC

- 3. IN1 : *High/ Low Output*
- 4. IN2 : *High/ Low Output*
- 5. GND : *Ground*



**Gambar 2. 6 Relay**  
(Sumber: Industri306, 2016)

### **2.3.6 Pompa Air**

Pompa adalah alat yang digunakan untuk mengangkat cairan dari permukaan yang lebih rendah ke permukaan yang lebih tinggi, atau untuk memindahkan cairan dari area dengan tekanan rendah ke area dengan tekanan lebih tinggi. Pompa bekerja berdasarkan prinsip perbedaan tekanan antara bagian masuk (*suction*) dan bagian keluar (*discharge*). Dengan kata lain, pompa mengubah tenaga mekanis (penggerak) menjadi tenaga kinetis (kecepatan), yang berguna untuk mengalirkan cairan dan mengatasi hambatan sepanjang jalur aliran. Komponen utama pompa dapat dilihat lebih lanjut pada Gambar 2.7.

Pompa memainkan peran penting dalam berbagai aplikasi, seperti sistem irigasi, pengolahan air, dan industri kimia, di mana aliran cairan perlu dikendalikan dan dipindahkan secara efisien. Dengan menggunakan pompa yang sesuai dengan kebutuhan, proses pemindahan cairan dapat dilakukan dengan lebih optimal, mengurangi pemborosan energi, dan meningkatkan kinerja sistem secara keseluruhan.



**Gambar 2. 7 Pompa Air Celup (*Submersible*)**

### 2.3.7 ESP32

ESP32 DevKit adalah *mikrokontroler* Espresif yang mengikuti jejak ESP8266. Kemampuan Wi-Fi dan *Bluetooth* 4.2 bawaan ESP32 ini hanyalah dua dari sekian banyak manfaat yang ditawarkannya dibandingkan Arduino. Selain itu, CPU *Dual-Core* 32-bit 160/240MHz milik ESP32 sangat cepat.



**Gambar 2. 8 ESP32 DevKit**

ESP32 *DevKit* menghilangkan kebutuhan akan modul tambahan untuk Wi-Fi atau *Bluetooth*, menjadikan ESP32 *DevKit* sebagai opsi yang menarik untuk pemrograman berbasis *Internet of things* (IoT). Selain itu terlihat seperti yang ditunjukkan pada Gambar 2.8, ESP32 memiliki 36 pin GPIO, yang merupakan pin *input* dan *output* analog dan digital yang secara kolektif dikenal sebagai *General Purpose Input Output*. Melihat perbandingan ESP8266 dan ESP32 dalam hal kemampuan dan spesifikasi lengkap pada Tabel 2.2.

**Tabel 2. 2 Perbandingan ESP8266 dengan ESP32**

Spesifikasi	Board	
	ESP8266	ESP32
MCU	Xtensa Single-core 32-bit L106	Xtensa Dual-Core 32-bit LX6 with 600DMIPS
Wi-Fi	802.11 b/g/n tipe HT20	802.11 b/g/n tipe HT40
<i>Bluetooth</i>	Tidak Ada	<i>Bluetooth</i> 4.2 dan BLE
Frekuensi	80 MHz	160 MHz
SRAM	Tidak Ada	Ada
Total GPIO	17 Pin	36 Pin
Total ADC pin	1 Pin	15 Pin
Total Digital pin	9 Pin	2 Pin
Tegangan <i>Output</i>	3.3 – 5 Volt	3.3 – 5 Volt
Total SPI-UART-I2C-I2S	2-2-1-2	4-2-2-2
Resolusi ADC	10 Bit	12 Bit
Suhu operasional kerja	-40°C hingga 125°C	-40°C hingga 125°C
Sensor dalam modul	Tidak ada	Touch Sensor, <i>Temperature</i> Sensor, <i>Hall Effect</i> Sensor
Harga di pasaran	Rp. 30.000 – 350.000	Rp. 70.000 – 650.000

Tabel 2.2 menunjukkan dengan jelas bahwa ESP32 adalah pilihan yang lebih baik, berkat prosesornya yang lebih besar, yang memungkinkan pemrosesan data lebih cepat. ESP32 juga memiliki lebih banyak pin analog-ke-digital (ADC) daripada ESP8266, yang memungkinkannya menangani tugas pemrograman yang lebih rumit.

#### 2.4 Perangkat Lunak

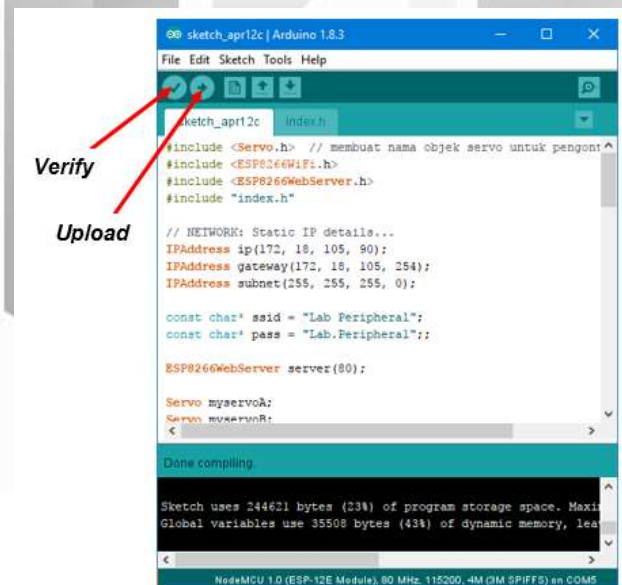
Perangkat lunak, atau yang lebih dikenal dengan sebutan *software*, adalah kumpulan data elektronik yang disusun dan disimpan di dalam komputer berupa program atau instruksi yang menjalankan perintah tertentu. *Software* bertindak sebagai penerjemah perintah yang diberikan oleh *user* atau pengguna, kemudian diteruskan dan diproses oleh *hardware* atau perangkat keras. Ketiadaan perangkat lunak, sistem komputer tidak dapat menjalankan perintah dan fungsinya secara efektif.

### 2.4.1 Software Mikrokontroler Arduino Uno

Software arduino yang digunakan adalah *driver* dan IDE, walaupun masih ada beberapa *software* lain yang sangat berguna selama pengembangan arduino. *Integrated Development Environment* (IDE), suatu program khusus untuk suatu komputer agar dapat membuat suatu rancangan atau sketsa program untuk papan Arduino. IDE arduino merupakan *software* yang sangat canggih ditulis dengan menggunakan java.

### 2.4.2 Perangkat Lunak Arduino IDE

IDE merupakan kependekan dari *Integrated Development Enviroenment*. IDE merupakan program yang digunakan untuk membuat program pada Arduino Uno. Program yang ditulis dengan menggunakan *Software* Arduino IDE disebut sebagai *sketch*. *Sketch* ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi.ino, seperti pada Gambar 2.9.



**Gambar 2. 9 Arduino IDE**  
(Sumber: Arranda Ferdian D.)

Papan yang dikonfigurasi dan *port* COM yang digunakan ditampilkan di kanan bawah program Arduino IDE (Arranda Ferdian D, 2017).

- a. *Verify/Compile*, berfungsi untuk mengecek apakah sintaks program dalam *sketch* sudah sesuai atau belum. Jika tidak ada kesalahan maka sintaks yang dibuat akan dicompile kedalam bahasa mesin.
- b. *Upload*, berfungsi mengirimkan program yang sudah dikompilasi ke *Arduino Board*.

### 2.4.3 Logika *Fuzzy* (*Fuzzy logic*)

Logika *Fuzzy* adalah suatu cara pandang yang tepat untuk memetakan suatu ruang *input* ke dalam suatu ruang *output*. Ada beberapa cara untuk memetakan *input* ke *output* diantaranya:

1. Sistem *Fuzzy*
2. Sistem linear
3. Sistem pakar
4. Jaringan syaraf
5. Persamaan differensial
6. Tabel interpolasi multi-dimensi

#### 2.4.3.1 Klasifikasi *Fuzzy*

Sistem *Fuzzy* mencakup berbagai jenis aplikasi, seperti sistem Klasifikasi *Fuzzy* dan sistem diagnosis *Fuzzy*. Dalam cakupan yang lebih luas, terdapat pula sistem lain yang telah terbukti sukses, seperti sistem pakar *Fuzzy*, sistem analisis data *Fuzzy*, sistem pengolahan citra *Fuzzy*, serta berbagai aplikasi lainnya (Richard, 2006).

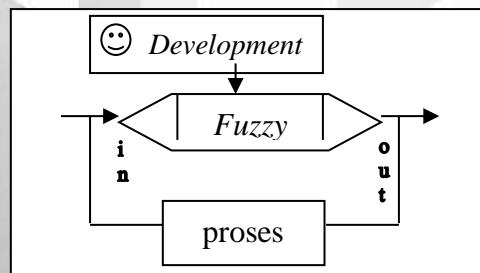
Selain itu, teknologi *Fuzzy* telah diterapkan di berbagai bidang, termasuk sistem kontrol *Fuzzy* yang digunakan untuk pengaturan otomatis dalam industri manufaktur, sistem HVAC, dan kendaraan otonom. Teknologi ini memungkinkan pengambilan keputusan yang lebih adaptif dan fleksibel, terutama di lingkungan yang tidak pasti atau tidak terstruktur.

Dengan kemampuannya mengatasi ketidakpastian dalam data dan mengaburkan batasan yang kaku, sistem *Fuzzy* juga berpotensi digunakan dalam

analisis prediksi di berbagai bidang, seperti keuangan, peramalan cuaca, dan diagnostik medis di mana variabilitas data dan ketidakpastian sering menjadi tantangan utama.

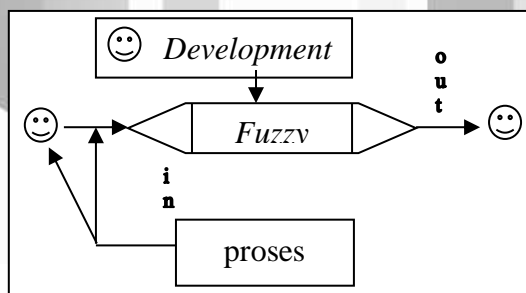
### 1. Sistem Kendali *Fuzzy*

*Fuzzy* memiliki beberapa jenis Klasifikasi yaitu, sistem kendali. Sistem kendali *Fuzzy* yang diGambarkan, Gambar 2.10 merupakan suatu sistem lingkaran tertutup, yang tidak terdapat operator yang menjadi bagian dari sistem lingkaran kendali (*control loop*) (Ross, 2014). Contoh dari sistem ini adalah *vacuum cleaner*, sistem pada alat ini mengatur daya motor penghisap tergantung dari banyaknya debu.



**Gambar 2. 10 Sistem Kendali *Fuzzy***

### 2. Sistem Klasifikasi *Fuzzy*

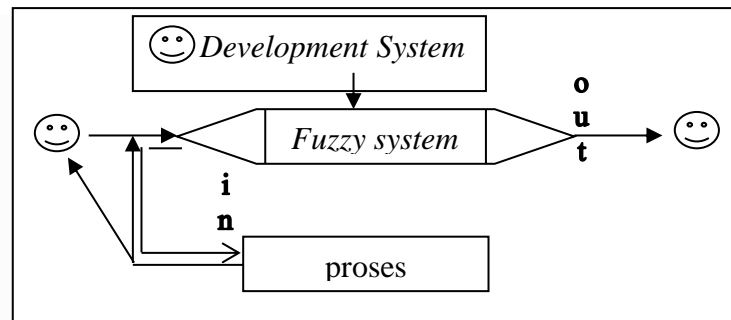


**Gambar 2. 11 Sistem Klasifikasi *Fuzzy***

Sistem Klasifikasi *Fuzzy* pada Gambar 2.11 bukan termasuk sitem lingkaran tertutup (*close loop*), sistem ini hanya menerima masukan dan meberikan keluaran dari proses untuk selanjutnya memberikan informasi berupa (*state*), dari proses tadi informasi kondisi ini digunakan untuk mengendalikan sistem atau memberikan tanggung jawab kendali kepada operator. Contoh dari sistem ini adalah mesin cuci yang sistem pada alat ini akan menjalankan motor dari aturan *slector* yang

diberikan. Keputusan yang dihasilkan sistem *Fuzzy* ini berbasis pada aturan-aturan yang telah ditetapkan sebelumnya untuk merespons kondisi masukan yang berubah-ubah.

### 3. Sistem Diagnosis *Fuzzy*



**Gambar 2. 12 Sistem Diagnosis *Fuzzy***

Pada sistem ini, Gambar 2.12 peranan manusia atau operator lebih dominant. Ketika sistem memerlukan data tambahan maka, pengiriman data akan dilaksanakan oleh operator, selain itu operator dapat meminta atau menanyakan informasi dari sistem diagnosis berupa hasil koklusi diagnosis atau prosedur detail hasil diagnosis oleh sistem. Dilihat dari sifatnya, sistem diagnosis *Fuzzy* dapat digolongkan ke dalam sistem pakar *Fuzzy*. Sistem pakar *Fuzzy* menggunakan notasi *Fuzzy* pada aturan-aturan dan proses inferensi (logika keputusan).

#### 2.4.3.2 Himpunan Crisp Dan Himpunan *Fuzzy*

Bagian-bagian yang membentuk himpunan Crisp adalah yang mendefinisikannya. Nilai yang dikaitkan dengan elemen  $a$  dalam himpunan  $A$  adalah 1. Meskipun demikian,  $a$  tidak memiliki nilai selain nol jika bukan merupakan elemen  $A$ . Jika item  $x$  termasuk dalam  $A$  dan  $P(x)$  benar, maka  $A$  ditulis sebagai  $A = \{x P(x)\}$ . Untuk properti  $P$  dan fungsi karakteristik  $X_A$  dari  $A$ , kondisi  $P(x) = 1$  harus dipenuhi sebelum  $P(x)$  dapat dianggap benar.

Himpunan *Fuzzy* didasarkan pada gagasan untuk memperluas jangkauan fungsi karakteristik sedemikian sehingga fungsi tersebut akan mencakup bilangan real pada interval  $[0,1]$ . Nilai keanggotaannya menunjukkan bahwa suatu item dalam semesta pembicaraan tidak hanya berada pada 0 atau 1, namun juga nilai yang terletak diantaranya. Dengan kata lain, nilai kebenaran suatu item tidak hanya



bernilai benar atau salah. Nilai 0 menunjukkan salah, nilai 1 menunjukkan benar dan masih ada nilai-nilai yang terletak antara benar dan salah. Beberapa hal yang perlu diketahui dalam memahami sistem *Fuzzy* sebagai berikut:

1. Variabel *Fuzzy*

Merupakan variabel yang hendak dibahas dalam suatu sistem *Fuzzy* contoh, umur, *temperature*, permintaan dan sebagainya.

2. Himpunan *Fuzzy*

Merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *Fuzzy*, contoh:

- a. Variabel umur dibagi menjadi 3 himpunan *Fuzzy* yaitu muda, parobaya, tua
- b. Variabel *temperature* dibagi menjadi 5 himpunan *Fuzzy* yaitu dingin, sejuk, normal, hangat dan panas.

3. Semesta Pembicaraan

Keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *Fuzzy*, contohnya semesta pembicaraan untuk variabel *temperature*.

4. *Domain*

Keseluruhan nilai yang diinginkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *Fuzzy*.

5. Nilai Ambang *Alfa-Cut*

Merupakan nilai ambang batas *domain* yang didasarkan pada nilai keanggotaan untuk tiap-tiap *domain* dimana  $\alpha$  - *cut* memiliki 2 kondisi.

a.  $\alpha$  -*cut* lemah dapat dinyatakan sebagai:  $\mu(x) \geq \alpha$  .....(2.1)

b.  $\alpha$  -*cut* kuat dapat dinyatakan sebagai:  $\mu(x) > \alpha$  .....(2.2)

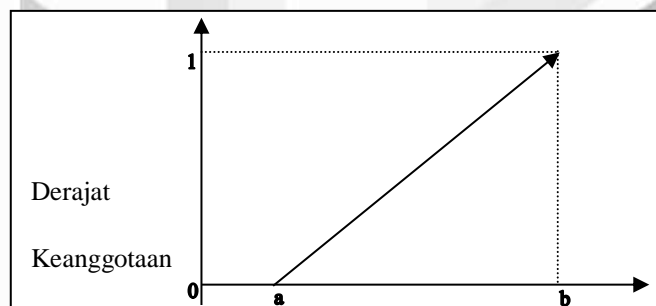
**2.4.3.3 Fungsi Keanggotaan**

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data kedalam nilai keanggotaannya (sering

disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Ada beberapa bentuk kurva yang dapat diambil oleh fungsi keanggotaan, seperti:

### 1. Representasi Linier

Garis lurus merepresentasikan permukaan dalam bentuk linier. Jika mencoba memahami ide yang kurang jelas, ini adalah metode yang paling mudah digunakan. Dalam himpunan *Fuzzy* linier, ada dua kemungkinan hasil. Dimulai dengan nilai dominan dengan derajat keanggotaan nol [0], himpunan meningkat seperti yang ditunjukkan pada Gambar 2.13, bergerak ke kanan menuju nilai *domain* dengan derajat keanggotaan yang lebih tinggi.

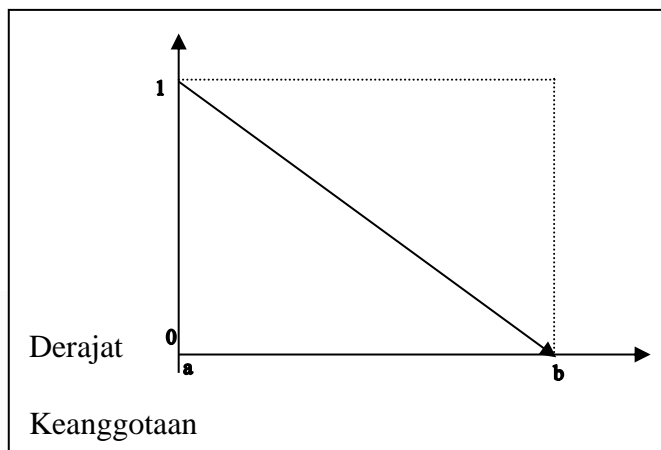


**Gambar 2. 13 Representasi Linear Naik**  
**Sumber:(Ross, 2014)**

Fungsi keanggotaan:

$$\mu(x) = \begin{cases} 0; & x \leq a \\ (x - a)/(b - a); & a \leq x \leq b \\ 1; & x \geq b \end{cases} \dots\dots\dots(2.3)$$

Bagian kedua adalah menunjukkan bahwa hal ini bertentangan dengan Gambar pertama. Gambar 2.14 menunjukkan garis lurus yang dimulai pada nilai *domain* dengan derajat keanggotaan terbesar di sebelah kiri dan menurun ke nilai *domain* dengan derajat keanggotaan terendah. Kondisi ini mencerminkan fungsi keanggotaan dengan pola penurunan linear, yang sering digunakan untuk merepresentasikan konsep "semakin kecil, semakin baik" dalam logika *Fuzzy*. Pola ini memungkinkan sistem untuk mengidentifikasi tingkat penurunan nilai secara bertahap, sehingga lebih sensitive terhadap perubahan kecil pada *domain input*. Dengan demikian, fungsi ini memberikan fleksibilitas dalam menentukan keputusan berbasis aturan *Fuzzy* yang membutuhkan tingkat ketelitian tinggi.



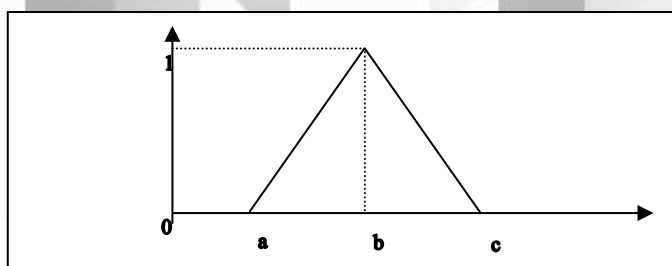
**Gambar 2. 14 Representasi Linear Turun**  
**Sumber:(Ross, 2014)**

Fungsi keanggotaan:

$$\mu(x) = \begin{cases} (b-x)/(b-a); & a \leq x \leq b \\ 0; & x \geq b \end{cases} \dots\dots\dots(2.4)$$

2. Representasi Kurva Segitiga

Gambar 2.15 menunjukkan bahwa kurva segitiga pada dasarnya merupakan kombinasi dua garis linier.



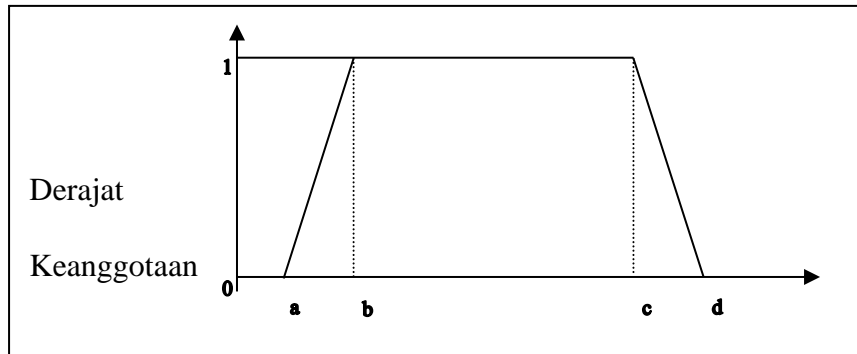
**Gambar 2. 15 Kurva Segitiga**  
**Sumber:(Ross, 2014)**

Fungsi keanggotaan:

$$\mu(x) = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ (x-a)/(b-a); & a \leq x \leq b \\ (b-x)/(c-b); & b \leq x \leq c \end{cases} \dots\dots\dots(2.5)$$

### 3. Representasi Kurva Trapesium

Seperti ditunjukkan dalam Gambar 2.16, kurva trapesium menyerupai kurva segitiga dengan pengecualian sejumlah titik yang memiliki nilai keanggotaan 1.



**Gambar 2. 16 Kurva Trapesium**  
**Sumber:(Ross, 2014)**

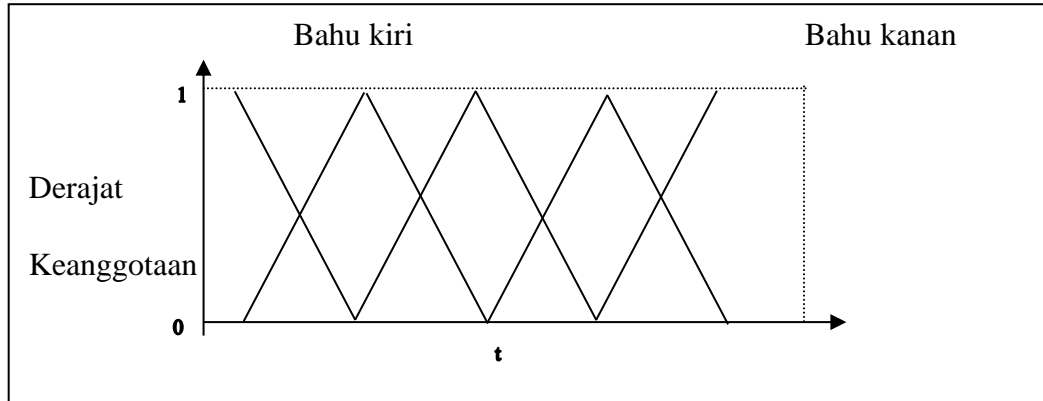
Fungsi keanggotaan:

$$\mu(x) = \begin{cases} 0; & x \leq a \text{ atau } x \geq d \\ (x-a)/(b-a); & a \leq x \leq b \\ 1; & b \leq x \leq c \\ (d-x)/(d-c); & c \leq x \leq d \end{cases} \dots\dots\dots(2.6)$$

### 4. Representasi Kurva Bentuk Bahu

Representasi Kurva Bentuk Bahu adalah salah satu teknik dalam logika *Fuzzy* yang digunakan untuk menggambarkan variabel yang memiliki karakteristik asimetris atau tidak linier. Berbeda dengan bentuk segitiga yang simetris, kurva bahu cenderung lebih fleksibel dalam mencerminkan perubahan nilai suatu variabel, terutama pada sisi kiri atau kanan yang mungkin tidak selalu mengalami perubahan yang sama. Himpunan *Fuzzy* 'bahu' digunakan untuk membatasi rentang nilai dari variabel dengan lebih adaptif. Dalam kebanyakan kasus, bahu kiri melambangkan transisi dari kebenaran ke kesalahan, sedangkan bahu kanan melambangkan kebalikannya, dari kesalahan ke benar. Penggunaan kurva bahu ini memungkinkan model *Fuzzy* untuk menggambarkan batasan variabel dengan lebih realistis, terutama dalam situasi di mana transisi antara dua kondisi tidak selalu linier atau simetris. Sebagai contoh, bahu kiri dan kanan dalam representasi variabel *Fuzzy* dapat dilihat pada Gambar 2.17, yang memperlihatkan bagaimana bahu

tersebut membatasi rentang nilai dengan lebih akurat tanpa harus mengikuti pola yang seragam. Hal ini memberikan fleksibilitas lebih dalam penggambaran sistem yang kompleks.



**Gambar 2. 17 Kurva Bentuk Bahu**  
**Sumber:(Ross, 2014)**

#### 2.4.4 Sistem Interferensi *Fuzzy*

Sistem inferensi *Fuzzy* merupakan langkah penting dalam pengambilan keputusan berbasis logika *Fuzzy*. Proses ini melibatkan penentuan aturan-aturan *Fuzzy* dan penerapan fungsi keanggotaan untuk mengubah *input* menjadi *output* yang dapat dimengerti. Dengan menggunakan mekanisme ini, suatu sistem dapat menangani ketidakpastian dan data yang tidak jelas untuk menghasilkan keputusan yang tepat.

##### 2.4.4.1 Fungsi Keanggotaan

Metodologi implikasi *Fuzzy* didasarkan pada metode penalaran monoton. Meskipun penalaran semacam ini tidak sering digunakan, terkadang digunakan untuk penskalaan *Fuzzy*. Mengingat implikasi langsung berikut antara dua *domain*:

*IF x is A THEN y is B*

Transfer fungsi:

$$y = f((x, A), B)$$

Dengan demikian, komposisi dan dekomposisi *Fuzzy* tidak diperlukan untuk pengoperasian sistem *Fuzzy*. Dapat memperkirakan nilai keluaran dengan melihat derajat keanggotaan antesedennya.

#### **2.4.4.2 Metodologi Desain Sistem *Fuzzy***

Metodologi desain sistem *Fuzzy* melibatkan beberapa langkah penting yang bertujuan untuk mengembangkan sistem berbasis logika *Fuzzy*. Metodologi desain sistem *Fuzzy* melibatkan penentuan variabel *input-output*, pembentukan fungsi keanggotaan, dan pembuatan aturan *Fuzzy*. Proses ini diakhiri dengan inferensi dan *defuzzifikasi* untuk menghasilkan *output* yang konkret dari nilai *Fuzzy*. Dengan metodologi ini, sistem *Fuzzy* menjadi lebih fleksibel dan efektif dalam menangani situasi yang kompleks dan tidak pasti.

#### **2.4.4.3 Mendefinisikan karakteristik model secara fungsional dan operasional**

Pada bagian ini perlu diperhatikan karakteristik apa saja yang dimiliki oleh sistem yang ada, kemudian dirumuskan karakteristik operasi-operasi yang akan digunakan pada model *Fuzzy*.

#### **2.4.4.4 Melakukan dekomposisi variabel model menjadi himpunan *Fuzzy***

Himpunan *Fuzzy* Terkait dibangun dari variabel-variabel yang ditetapkan dengan tetap memperhatikan *domainnya*.

#### **2.4.4.5 Membuat aturan *Fuzzy***

Aturan pada *Fuzzy* menunjukkan bagaimana suatu sistem beroperasi. Cara penulisan aturan secara umum adalah *IF* ( $X_1$  is  $A_1$ ). ... . ( $X_n$  is  $A_n$ ) *THEN*  $Y$  is  $B$  dengan ( $\cdot$ ) adalah operator (*OR* atau *AND*),  $X$  adalah scalar dan  $A$  adalah variabel linguistik. Hal yang perlu diperhatikan dalam membuat aturan adalah:

- a. Masukkan semua aturan dalam satu kategori jika semuanya berlaku untuk variabel yang sama.
- b. Pastikan aturan berada dalam urutan yang mudah dibaca.
- c. Gunakan identitas untuk mengilustrasikan struktur aturan.

- d. Gunakan penamaan yang umum untuk mengidentifikasi variabel-variabel pada kelas yang berbeda.
- e. Gunakan komentar untuk menjelaskan mengapa aturan atau serangkaian aturan ada.
- f. Pisahkan aturan dengan spasi.
- g. Gunakan huruf kapital untuk variabel dan himpunan *Fuzzy*, dan huruf kecil untuk semua hal lain dalam bahasa tersebut.

#### **2.4.4.6 Menentukan metode *Defuzzy* untuk tiap-tiap variabel solusi**

Pada tahap *defuzzifikasi*, dipilih nilai tunggal dari variabel solusi yang mewakili konsekuensi dari daerah *Fuzzy*. Metode yang paling umum digunakan adalah metode centroid, karena memiliki konsistensi tinggi serta sensitif terhadap tinggi dan lebar total daerah *Fuzzy*. Metode ini bekerja dengan menghitung pusat area dari bentuk *Fuzzy* yang dihasilkan melalui proses inferensi, menghasilkan nilai tunggal yang menjadi solusi akhir sistem *Fuzzy*. Selain metode centroid, terdapat metode lain seperti bisektor, *mean of maximum*, dan *weighted average*. Masing-masing metode memiliki karakteristik unik dalam menghasilkan nilai *defuzzifikasi*, memberikan fleksibilitas sesuai kebutuhan sistem yang diterapkan.

#### **2.4.4.7 Menjalankan simulasi sistem**

Simulasi sistem lengkap dibuat pada tahap ini. Kenyamanan pengguna harus menjadi tujuan utama simulasi. Kita juga harus memikirkan kemungkinan terjadinya evolusi. Pada program simulasi perlu disertakan editor sebagai berikut:

- a. Perbaiki variabel model *input / output*
- b. Perbaiki himpunan *Fuzzy*
- c. Perbaiki aturan
- d. Pilihan metode *defuzzifikasi*

#### **2.4.5 *Internet of things***

*Internet of things* (IoT) adalah konsep yang bertujuan memperluas manfaat konektivitas internet agar dapat tersambung secara terus-menerus dengan berbagai perangkat dan benda di dunia nyata (Selay dkk., 2022). IoT memungkinkan berbagi

data, pengendalian jarak jauh, dan berbagai fungsi lainnya, termasuk pada objek seperti bahan pangan, peralatan elektronik, koleksi, hingga makhluk hidup. Semua ini terhubung ke jaringan lokal maupun global melalui sensor yang tertanam dan selalu aktif.

Secara sederhana, IoT mengacu pada benda yang dapat diidentifikasi secara unik sebagai representasi virtual dalam struktur berbasis internet. Ilustrasi mengenai konsep IoT dapat dilihat pada Gambar 2.18. Istilah *Internet of things* pertama kali diperkenalkan oleh Kevin Ashton pada tahun 1999 dan mulai dikenal luas melalui *Auto-ID Center* di MIT. Kini, IoT telah menjadi salah satu topik penting yang dipelajari oleh mahasiswa di perguruan tinggi.

Perkembangan IoT telah mengubah cara manusia berinteraksi dengan teknologi dan membuka peluang baru di berbagai industri, mulai dari kesehatan hingga manufaktur (Susanto dkk., 2022). Dengan kemampuan mengumpulkan dan menganalisis data secara *real-time*, IoT memungkinkan pengambilan keputusan yang lebih cepat, akurat, dan efisien.



**Gambar 2. 18 Ilustrasi dari *Internet of things***  
(Sumber : [www.meccanismocomplexo.org/en/iot-internet-of-things/](http://www.meccanismocomplexo.org/en/iot-internet-of-things/).)

#### **2.4.6 Android**

Android adalah sistem operasi, *middleware*, dan aplikasi untuk perangkat seluler yang dibangun di atas platform Linux (Raharjo dkk., 2021). Android adalah platform gratis dan sumber terbuka yang memungkinkan programmer membuat aplikasi untuk berbagai jenis perangkat seluler (Deshmukh dkk., 2018). Pada awalnya, Google Inc mengakuisisi Android Inc, perusahaan rintisan yang mengembangkan perangkat lunak telepon pintar. Google, Intel, Motorola,



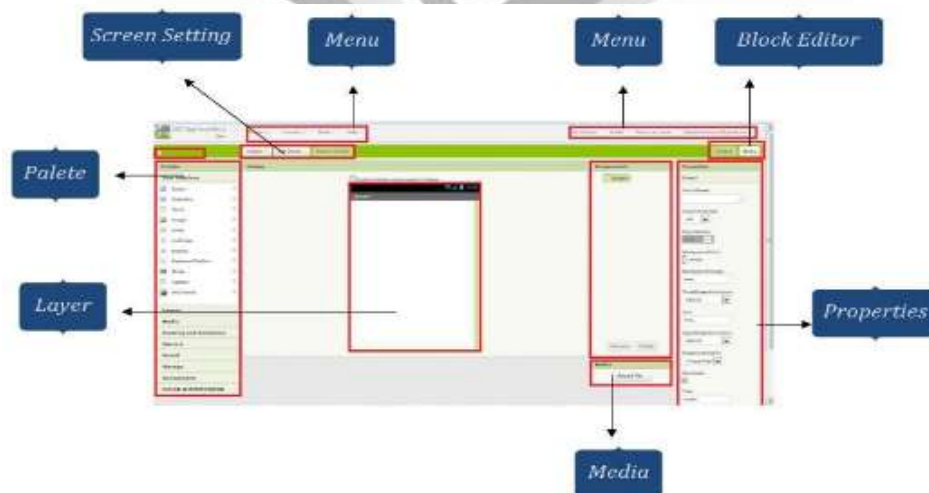
Qualcomm, T-Mobile, dan Nvidia termasuk di antara 34 bisnis perangkat keras, perangkat lunak, dan telekomunikasi yang bersatu untuk mendirikan *Open Handset Alliance*, yang kemudian mengembangkan Android pada bulan November 2007. Sebaliknya, Google mendistribusikan kode sumber Android sesuai dengan *lisensi Apache*, yang merupakan lisensi perangkat lunak dan standar terbuka perangkat seluler. Pengembang memiliki banyak keleluasaan untuk membuat perubahan dan berbagi aplikasi Android menggunakan lisensi ini.

#### 2.4.7 Aplikasi MIT App Inventor 2

Mit App Inventor *for Android* adalah aplikasi yang pada dasarnya disediakan oleh Google dan sekarang di-maintenance oleh *Massachusetts Institute of Technology* (MIT) (Abdul Karim dkk., 2020). App Inventor menggunakan bahasa *Kawa Language Framework* dan *Kawa's dialect* yang dikembangkan oleh Per Bothner. Kedua aplikasi tersebut dijadikan sebagai *compiler* dan menerjemahkan *Visual Block Programming*.

#### 2.4.8 Area kerja MIT App Inventor 2

Area kerja MIT App Inventor 2 terdiri dari dua bagian utama yaitu Designer untuk merancang antarmuka aplikasi dan *Blocks Editor* untuk memprogram logika aplikasi menggunakan blok-blok visual. Kedua area ini memungkinkan pengguna membangun aplikasi dengan mudah melalui pendekatan *drag-and-drop* seperti pada Gambar 2.19.



Gambar 2. 19 Area Kerja App Inventor 2

Komponen ruang kerja App Inventor 2 dijelaskan di sini.

1. *Screen Setting* adalah kumpulan layar yang dapat digunakan untuk mengatur, menambah, dan menghapus layar.
2. *Pallette* adalah tata letak dengan tombol terkait desain pada panel.
3. Menu adalah serangkaian menu yang dapat membantu saat membuat proyek baru, mencari tahu cara memperbaiki proses, mengubah file APK, dan banyak hal lainnya.
4. *Block Editor* menyediakan akses ke halaman kode blok proses pengkodean melalui tombol.
5. *Properties* adalah alat untuk melapisi komponen-komponen yang dikembangkan sebelumnya ke dalam suatu desain.
6. Media menawarkan lokasi untuk menyimpan berkas.
7. *Layer* mungkin memerlukan beberapa pekerjaan desain.

#### 2.4.9 Wi-Fi

Logo pada Gambar 2.20 menampilkan Wi-Fi, akronim untuk *Wireless Fidelity*, teknologi nirkabel yang dapat menyediakan akses internet dengan *bandwidth* tinggi hingga 11 Mbps untuk standar 802.11b menggunakan standar IEEE 802.11x (Kadir & Tone, 2015). Hotspot memungkinkan pengguna dengan laptop yang sudah memiliki kartu Wi-Fi untuk terhubung ke internet di tempat mana pun yang memiliki perangkat Wi-Fi.



**Gambar 2. 20 Logo Wi-Fi**

Jaringan area lokal nirkabel (Wi-Fi) memungkinkan pengguna untuk mentransmisikan data dengan aman dan cepat menggunakan gelombang radio, seperti halnya koneksi telepon seluler. Selain menghubungkan perangkat ke

internet, Wi-Fi juga memungkinkan bisnis untuk menyiapkan jaringan nirkabel. Karena alasan ini, *Freedom* merupakan asosiasi umum dengan Wi-Fi, karena teknologi ini memungkinkan pengguna mengakses internet dan transfer data tanpa batas di tempat umum seperti kampus, kafe, ruang konferensi, dan hotel.

#### 2.4.10 Spesifikasi Wi-Fi

Spesifikasi Wi-Fi dirancang berdasarkan spesifikasi IEEE 802.11. Sekarang ini ada empat variasi dari 802.11, yaitu: 802.11a, 802.11b, 802.11g, dan 802.11n. Spesifikasi b merupakan produk pertama Wi-Fi (Jannah dkk., 2015). Tabel 2.3 memperlihatkan detail spesifikasi dari setiap variasi Wi-Fi berdasarkan standar 802.11, yang menunjukkan perbedaan kecepatan dan frekuensi operasional masing-masing.

**Tabel 2. 3 Spesifikasi Wi-Fi 802.11**

Spesifikasi Band	Kecepatan	Frekuensi (GHz)	Spesifikasi Band
802.11b	11 Mbps	2,4	B
802.11a	54 Mbps	5,0	A
802.11g	54 Mbps	2,4	b, g
802.11n	100 Mbps	2,4	b, g, n

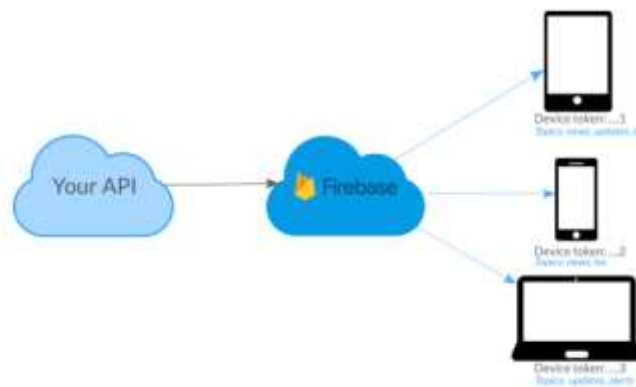
#### 2.4.11 *Firestore*

Layanan Google *Firestore* menyederhanakan proses pembuatan aplikasi bagi para pengembang. Para pengembang aplikasi dapat mengurangi pekerjaan dan lebih berkonsentrasi pada pengembangan aplikasi menggunakan *Firestore* (Ulumudin, 2023). Lihat bagaimana masing-masing aspek ini berkontribusi pada proses pengembangan aplikasi pada Gambar 2.21, yang mendukung pengembangan aplikasi yang efisien. Berikut adalah fungsi-fungsi *Firestore* yang digunakan saat mengembangkan aplikasi chat:

1. *Firestore Authentication* agar berfungsi dengan baik, sebagian besar aplikasi memerlukan autentikasi. Aplikasi dapat menyimpan data pengguna dengan aman di *cloud* dan memberikan pengalaman yang disesuaikan secara konsisten di semua perangkat pengguna saat mereka mengetahui identitasnya. Untuk membantu mengautentikasi pengguna ke aplikasi, *Firestore Authentication* menawarkan layanan *backend*, SDK sederhana, dan kerangka kerja UI yang

telah dibuat sebelumnya. Beberapa penyedia identitas terfederasi umum, termasuk Google, Facebook, dan Twitter, serta nomor telepon, kompatibel dengan *Firebase Authentication*.

2. *Firebase Real-time Database* adalah basis data yang disimpan di cloud. Setiap klien yang terhubung menerima data secara *real-time*, yang disimpan sebagai JSON. Semua klien langsung mendapatkan perubahan data terbaru saat membuat aplikasi lintas platform menggunakan SDK Android, iOS, dan JavaScript. Hal ini karena semua klien berbagi satu contoh basis data *real-time*.
3. *Firebase Cloud Storage for Firebase* adalah layanan penyimpanan objek yang dikembangkan untuk Google yang sederhana, canggih, dan terjangkau. Google *Secured File Uploads and Downloads for Firebase Apps* menggunakan *Firebase Software Development Kit (SDK)* untuk penyimpanan awan berjalan dengan sempurna di jaringan apa pun. SDK menyediakan sarana untuk menyimpan media yang dibuat oleh pengguna, termasuk foto, video, musik, dan banyak lagi. dapat mengakses file yang sama di server menggunakan *Google Cloud Storage*.
4. *Firebase Cloud Functions* dengan mengintegrasikan *Firebase* dengan permintaan dan peristiwa HTTPS yang dihasilkan oleh fungsi *firebase*, dapat mengotomatiskan eksekusi kode *backend*. Tidak perlu khawatir tentang pengelolaan atau penskalaan server sendiri karena kode disimpan di *Google cloud* dan beroperasi dalam lingkungan yang terkendali.



**Gambar 2. 21** *Firebase*

#### **2.4.12** *Quality of Service (QoS)*

*Quality of Service (QoS)* adalah metode pengukuran yang digunakan agar penentuan kualitas jaringan mendapatkan hasil yang lebih baik dan teratur,

sehingga dapat memenuhi kebutuhan dari suatu layanan (Aprianto Budiman dkk., 2020). QoS digunakan untuk menilai sekumpulan atribut kinerja yang telah ditentukan dan dihubungkan dengan layanan tersebut. Model pemantauan QoS terdiri empat kelas, yaitu *Throughput*, *Packet Lost*, *Delay*, dan *Jitter* (Bhakti, 2017).

#### 2.4.13 Wireshark

Istilah lain untuk Wireshark adalah penganalisis paket jaringan, dan ini adalah perangkat lunak yang merekam dan menampilkan informasi terperinci dari paket data yang dikirim dan diterima melalui jaringan. Menangkap dan menganalisis paket data dalam bentuk apa pun dan dalam format protokol apa pun adalah hal yang mudah. Menganalisis kinerja jaringan menjadi mudah menggunakan wireshark (Luthfansa & Rosiani, 2021).

#### 2.5 Literatur Review

Bagian ini membahas literatur yang relevan dengan topik penelitian seperti pada Tabel 2.4. Tinjauan ini memberikan dasar teoritis serta menjelaskan penelitian sebelumnya yang mendukung pengembangan proyek ini.

**Tabel 2. 4 Ringkasan Literatur Review**

Nama Peneliti	Pembahasan
(Sandi & Fatma, 2023)	Tujuan: untuk mengetahui apa saja pemanfaatan IoT pada salah satu sector ekonomi di Indonesia yaitu pertanian.
(Walid & Fikri, 2022)	Tujuan: untuk mengembangkan sistem manajemen irigasi berbasis IoT dengan fokus pada monitoring, kontrol, dan manajemen jaringan komunikasi untuk efisiensi dan jangkauan luas.
(Brigida Helvia Vien, 2023)	Tujuan: untuk mengembangkan sistem irigasi berbasis IoT dengan monitoring, kontrol, dan manajemen jaringan untuk komunikasi yang efisien.
(Naufal, 2022)	Tujuan: untuk membuat alat pemantau ini menggunakan water flow sensor untuk mengukur debit air dan viscosity sensor untuk membaca kekeruhan.
(Adinda, 2023)	Tujuan: untuk melakukan Penyiraman tanaman otomatis berbasis IoT dengan NodeMCU ESP8266 mempermudah penyiraman dan aktivitas pertanian.
(Fadhilah dkk., 2021)	Tujuan: untuk mengukur kelembapan tanah, dengan hasil dikirim ke smartphone via WhatsApp. Keypad memungkinkan pengaturan kelembapan sesuai jenis tanaman. Saat tanah kering, ESP32 mengaktifkan relay untuk menyalakan pompa air.