



# INTERNET OF THINGS DALAM AGROINDUSTRI

**Disusun oleh Anang Andrianto, Achmad  
Maududie, Dwiretno, Istiyadi**

## **ABSTRACT**

Buku ini dimulai dengan menjelaskan konsep dasar IoT dan bagaimana teknologi ini diterapkan dalam konteks agroindustri. Kami membahas tentang sensor-sensor yang digunakan untuk memantau lingkungan pertanian, selanjutnya, buku ini mengeksplorasi berbagai aplikasi praktis IoT dalam agroindustri, termasuk pemantauan dan pengendalian tanaman secara real-time.

# PRAKATA

Selamat datang di buku ajar "Internet of Things (IoT) dalam Agroindustri". Buku ini dirancang untuk memberikan pemahaman komprehensif tentang bagaimana teknologi IoT dapat diterapkan dalam sektor agroindustri untuk meningkatkan efisiensi, produktivitas, dan keberlanjutan.

Perkembangan teknologi digital telah membuka peluang baru dalam berbagai bidang, termasuk pertanian. Dengan memanfaatkan sensor, aktuator, dan sistem komunikasi yang canggih, IoT memungkinkan para petani dan pemangku kepentingan lainnya untuk memantau dan mengelola kegiatan pertanian secara lebih efisien dan real-time. Implementasi IoT dalam agroindustri bukan hanya tentang mengadopsi teknologi baru, tetapi juga tentang mengubah cara kita bertani dan mengelola sumber daya.

Buku ini disusun untuk memberikan panduan langkah demi langkah tentang penggunaan IoT dalam berbagai aspek agroindustri, mulai dari pemantauan lingkungan, irigasi cerdas, manajemen nutrisi, hingga analitik dan prediksi. Setiap bab disertai dengan contoh kasus nyata dan studi kasus yang relevan untuk membantu pembaca memahami konsep dan aplikasi praktis IoT.

Harapan kami, buku ini tidak hanya menjadi sumber referensi yang berguna bagi para mahasiswa dan peneliti, tetapi juga bagi praktisi industri yang ingin memahami dan mengimplementasikan teknologi IoT dalam kegiatan pertanian mereka. Dengan pemahaman yang lebih baik tentang potensi dan tantangan IoT, kami berharap dapat mendorong lebih banyak inovasi dan kolaborasi untuk masa depan agroindustri yang lebih cerdas dan berkelanjutan.

Kami mengucapkan terima kasih kepada semua kontributor, peneliti, dan praktisi yang telah berbagi pengetahuan dan pengalaman mereka, sehingga buku ini dapat disusun dengan baik. Semoga buku ini bermanfaat dan menginspirasi banyak pihak dalam perjalanan mereka mengintegrasikan teknologi IoT ke dalam dunia pertanian.

Selamat membaca!

Jember, Juni 2024

Penulis

# DAFTAR ISI

PRAKATA .....	i
DAFTAR ISI .....	ii
DAFTAR TABEL .....	vii
DAFTAR GAMBAR .....	viii
TINJAUAN MATAKULIAH .....	x
BAB 1. PENGENALAN IOT DALAM AGROINDUSTRI.....	11
1.1    Pendahuluan.....	11
1.2    Pengertian Internet Of Things.....	11
1.2.1    Cara Kerja IoT.....	12
1.2.2    Keuntungan Menggunakan IoT.....	12
1.2.3    Tantangan IoT .....	13
1.3    Implementasi IoT di Sektor Agroindustri .....	14
1.3.1    Penggunaan Field Monitoring System (FMS) untuk Agroindustri .....	14
1.3.2    Penggunaan drone untuk Agroindustri.....	15
1.3.3    Smart greenhouse .....	15
1.3.4    Livestock management.....	16
1.3.5    Aplikasi pendukung penjualan dan edukasi .....	17
1.4    Keuntungan Penggunaan IoT dalam Agroindustri.....	17
1.4.1    Efisiensi produksi.....	17
1.4.2    Efisiensi distribusi.....	18
1.4.3    Tantangan Penerapan Smart Farming di Indonesia.....	18
1.5    Rangkuman .....	19
1.6    Latihan Soal .....	20
1.7    Kunci Jawaban Singkat:.....	21
1.8    Bahan Diskusi .....	22
1.9    Daftar Rujukan.....	22
BAB 2. PENGENALAN PERANGKAT IOT .....	23

2.1	Pendahuluan.....	23
2.2	Jenis-jenis Perangkat IoT .....	23
2.2.1	Sensor.....	24
2.2.2	Aktuator .....	41
2.2.3	Perangkat Edge ( <i>Edge Devices</i> ):.....	44
2.2.4	Perangkat Komunikasi: .....	46
2.2.5	Perangkat pemrosesan .....	48
2.2.6	Perangkat Penyimpanan Data.....	58
2.2.7	Perangkat Antarmuka Pengguna (User Interface Devices): 58	
2.2.8	Perangkat Keamanan:.....	58
2.2.9	Perangkat Pendukung Khusus:.....	59
2.3	Rangkuman .....	59
2.4	Latihan Soal Pengenalan Sensor .....	59
2.5	Bahan Diskusi .....	61
2.6	Rujukan .....	62
BAB 3.	PEMROGRAMAN SENSOR DAN AKTUATOR .....	63
3.1	Pendahuluan.....	63
3.1.1	Latar belakang .....	63
3.1.2	Tujuan .....	63
3.1.3	Langkah-langkah Pemrograman .....	63
3.2	Pemrograman sensor suhu udara pada Arduino.....	64
3.2.1	Bahan yang Dibutuhkan .....	64
3.2.2	Langkah-langkah Pemrograman .....	64
3.3	Pemrograman sensor kelembaban tanah pada arduino .....	67
3.3.1	Bahan yang Dibutuhkan .....	67
3.3.2	Langkah-langkah Pemrograman .....	68
3.4	Pemrograman sensor suhu tanah.....	70
3.4.1	Bahan yang Dibutuhkan .....	70
3.4.2	Langkah-langkah Pemrograman .....	70

3.5	Pemrograman sensor tekanan .....	73
3.5.1	Bahan yang Dibutuhkan .....	73
3.5.2	Langkah-langkah Pemrograman .....	74
3.6	Pemrograman sensor gerak .....	77
3.6.1	Bahan yang Dibutuhkan .....	77
3.6.2	Langkah-langkah Pemrograman .....	77
3.7	Pemrograman Aktuator Relay .....	80
3.7.1	Bahan yang Dibutuhkan .....	80
3.7.2	Langkah-langkah Pemrograman .....	80
3.8	Pemrograman aktuator motor stepper .....	83
3.8.1	Bahan yang Dibutuhkan .....	83
3.8.2	Langkah-langkah Pemrograman .....	83
3.9	Rangkuman .....	87
3.10	Latihan soal .....	88
3.11	Bahan Diskusi .....	88
3.12	Rujukan .....	89
BAB 4.	Protokol komunikasi dalam iot .....	90
4.1	Pendahuluan .....	90
4.2	Contoh Implementasi Protokol .....	93
4.2.1	Implementasi HTTP pada ESP32 .....	93
4.2.2	Implementasi MQTT pada ESP32 .....	97
4.2.3	Implementasi Lora pada ESP32 .....	100
4.2.4	Implementasi Celluler pada ESP32 .....	102
4.2.5	Implementasi <i>Bluetooth Low Energy</i> pada ESP32 .....	105
4.3	Rangkuman .....	107
4.4	Latihan Soal .....	107
4.5	Bahan Diskusi .....	108
4.6	Rujukan .....	108
BAB 5.	Perancangan dan Pemrograman Database .....	109
5.1	Pendahuluan .....	109

5.1.1	Pemilihan Database yang Sesuai .....	109
5.1.2	Integrasi Database dengan Sistem Embedded .....	109
5.1.3	Interaksi dengan Database .....	109
5.1.4	Perlakuan Terhadap Keterbatasan Hardware .....	110
5.1.5	Manajemen Kesalahan .....	110
5.1.6	Keamanan .....	110
5.2	Pemrograman Database dalam IOT .....	110
5.3	Pemrograman Database SQL menggunakan protokol HTTP 112	
5.4	Pemrograman Database menggunakan Broker Blynk .....	114
5.5	Pemrograman Database menggunakan Broker MQTT .....	117
5.6	Rangkuman .....	122
5.7	Latihan Soal .....	122
5.8	Bahan Diskusi .....	125
5.9	Daftar Rujukan .....	126
BAB 6.	Integrasi Sistem IoT dan Sistem informasi .....	127
6.1	Pendahuluan .....	127
6.2	Sistem Monitoring Hidroponik Berbasis IoT Dengan Sensor Suhu, pH, dan Ketinggian Air Menggunakan ESP8266 .....	129
BAB 7.	Perancangan UI dan pemrograman Web, desktop, atau mobile 137	
7.1	Pendahuluan .....	137
7.2	Pemrograman Front end dan Back end .....	138
7.3	Pemrograman Desktop .....	139
7.4	Pemrograman Mobile .....	140
7.5	Rangkuman .....	142
7.6	Bahan Diskusi .....	147
7.7	Daftar Rujukan .....	147
BAB 8.	Implementasi IoT dalam Agroindustri .....	149
8.1	Pengembangan Smart Fogponic System Pada Budidaya Tanaman Pakcoy di Pertanian Indoor .....	149

8.1.1	Rancangan Sistem Smart Fogponic.....	150
8.1.2	Rangkaian arsitektur perangkat keras sistem smart fogponic	151
8.1.3	Kode Program sistem smart fogponics.....	151
8.2	Implementasi Wireles Serial Master-Slave pada Sistem Penyemprotan Otomatis serta Monitoring Suhu dan Kelembapan Udara Budidaya Jamur Janggel.....	160
8.2.1	Desain Sistem.....	161
8.2.2	Pemrograman system .....	164
8.3	Bahan Diskusi .....	167
8.4	Daftar Rujukan.....	167

## **DAFTAR TABEL**



## DAFTAR GAMBAR

Gambar 1-1 Konsep dasar kerja IoT (Annisa, 2021) .....	12
Gambar 1-2 Field Monitoring System dalam Agroindustri (Manager, 2018).....	14
Gambar 1-3 Penggunaan drone dalam Agroindustri (Chusna, 2023) .....	15
Gambar 1-4 Green house dengan menerapkan IoT (Yulianti, 2022) .....	16
Gambar 1-5 Penerapan IoT dalam Live Stock Management ternak (Chusna, 2023) .....	16
Gambar 2-1 Modul sensor suhu dengan thermocouple .....	25
Gambar 2-2 Modul sensor temperature Sensor Rs485 dengan RTD .....	26
Gambar 2-3 Modul sensor temperature dengan thermistor .....	26
Gambar 2-4 Sht30-D Modul I2C Sensor Suhu Dan Kelembaban.....	27
Gambar 2-5 Modul sensor sushudengan infrared.....	28
Gambar 2-6 Modul sensor hygrometer .....	29
Gambar 2-7 Modul sensor DHT 11 .....	30
Gambar 2-8 Modul Sensor LDR Modul Sensor Cahaya.....	31
Gambar 2-9 Modul sel photovoltaik.....	32
Gambar 2-10 Modul Sensor Infrared Receiving KY-022 .....	33
Gambar 2-11 modul photodiode.....	33
Gambar 2-12 Modul sensor suhu dengan infrared .....	34
Gambar 2-13 Modul sensor PIR.....	34
Gambar 2-14 Sensor Tekanan .....	35
Gambar 2-15Modul sensor gas metana .....	35
Gambar 2-16 Modul sensor sentuh .....	36
Gambar 2-17 Modul sensor proximity .....	36
Gambar 2-18 Sensor posisi dengan resistensi .....	37
Gambar 2-19 Modul sensor aliran udara .....	37
Gambar 2-20 Modul Sensor PH air .....	38
Gambar 2-21 Sesor PH Tanah.....	38
Gambar 2-22 Modul sensor kelembaban tanah .....	39
Gambar 2-23 Modul Humidity sensor DHT11 .....	40
Gambar 2-24 Sensor suhu tanah DS18B20 .....	41
Gambar 2-25 Modul Aktuator linier .....	42
Gambar 2-26 Aktutor rotari.....	42
Gambar 2-27 Modul solenoid.....	43
Gambar 2-28 Motor servo .....	43
Gambar 2-29 Modul stepper.....	44
Gambar 2-30 Modul arduino uno .....	49

Gambar 2-31 Modul ESP32 .....	51
Gambar 2-32 Modul Raspberry pi.....	53
Gambar 2-33 Modul Beaglebone Black .....	54
Gambar 3-1 Rangkaian sensor DTH11 pada Arduino.....	65
Gambar 3-2 Rangkaian sensor kelembaban pada arduino.....	68
Gambar 3-3 Rangkaian sensor DS18B20 pada arduino .....	71
Gambar 3-4 Rangkaian sensor tekanan pada arduino .....	74
Gambar 3-5 Rangkaian sensor gerak PIR pada Arduino.....	78
Gambar 3-6 Rangkaian relay pada ESP32 .....	81
Gambar 3-7 Rangkaian motor stepper pada ESP32 .....	84
Gambar 6-1 Diagram alir system .....	131
Gambar 6-2 Rangkaian sistem .....	132
Gambar 8-1 Rancangan sistem fogponic.....	151

# **TINJAUAN MATAKULIAH**

Diskripsi matakuliah

Buku ajar ini mempelajari tentang komponen-komponen IoT, pemrograman, dan integrasinya dengan suatu sistem informasi untuk menyelesaikan permasalahan di bidang agroindustri. Materi yang akan dibahas dalam buku ini meliputi:

Pengenalan IoT dalam Agroindustri

Perangkat perangkat dalam sistem iot.

Pemrograman sensor dan actuator

Protokol komunikasi dalam iot

Perancangan dan pemrograman database

Integrasi sistem IoT dan sistem informasi

Perancangan UI dan pemrograman Web, desktop, atau mobile.

Implementasi IoT dalam Agroindustri

# **BAB 1. PENGENALAN IOT DALAM AGROINDUSTRI**

## **Capaian Pembelajaran Matakuliah (CPMK)**

Mahasiswa mampu memahami tentang peranan IoT, cara kerja keuntungan penggunaan IoT

### **1.1 Pendahuluan**

Materi pengenalan IoT bertujuan memberikan pembelajaran "Mahasiswa mampu memahami tentang peranan IoT, cara kerja keuntungan penggunaan IoT", Pemahaman yang diharapkan meliputi:

- a) konsep dasar Internet of Things (IoT) termasuk arsitektur, protokol, dan komponen utama dari sistem IoT,
- b) spesifikasi teknis dari berbagai mikrokontroler,
- c) teknologi dan protokol komunikasi yang digunakan dalam IoT seperti MQTT, CoAP, dan HTTP.

### **1.2 Pengertian Internet Of Things**

Internet of Things adalah sebuah konsep yang terhubung dengan perangkat sebagai media komunikasi berbasis internet. Dengan adanya IoT, seorang user dapat saling terhubung dan berkomunikasi untuk melakukan aktivitas tertentu, mencari, mengolah, dan mengirimkan informasi secara otomatis. Konsep ini sepintas hampir serupa dengan M2M (Machine-to-Machine). Akan tetapi, sebenarnya kedua konsep ini memiliki perbedaan dari segi skala dan lingkup penggunaannya.

M2M di sini merujuk pada teknologi yang memungkinkan komunikasi antara mesin-mesin tanpa melibatkan campur tangan manusia. Dengan kata lain, M2M lebih berfokus pada sistem kerja mesin untuk menjalankan sebuah program. Contoh paling mudah dilihat adalah pada pengoperasiannya mesin di sebuah pabrik. Di pabrik, mesin-mesin berjalan otomatis dan berkomunikasi antar-sesama mesin saja. Jadi, mereka bisa mengatur sendiri jalannya produksi tanpa perlu diintervensi oleh manusia.

### 1.2.1 Cara Kerja IoT

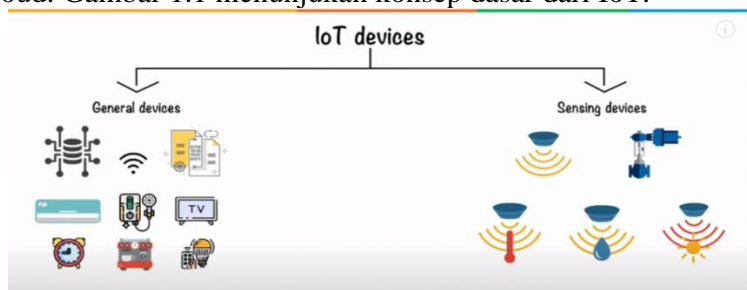
Pada dasarnya, IoT beroperasi dengan cara menghubungkan berbagai jenis perangkat seperti software atau hardware ke jaringan internet. Ada 3 komponen utama yang berperan penting dalam proses kerja IoT, yaitu sensor, gateway, dan cloud.

Sensor yang digunakan pada konsep ini dapat berupa sensor gerakan, sensor cahaya, dan jenis sensor lainnya. Tujuan dari penggunaan perangkat ini adalah untuk mengumpulkan data dari objek-objek fisik yang terhubung dengan jaringan internet.

Setelah sensor berhasil mengumpulkan data tersebut, komponen gateway berfungsi untuk mentransmisikan data itu ke cloud atau internet yang terhubung. Gateway di sini juga dapat memproses serta melakukan tindakan otomatis terhadap data yang ada, seperti mematikan atau menyalakan perangkat yang terhubung. Di sini, AI dapat membantu IoT untuk mengoptimalkan fungsi perangkat.

Terakhir, data yang sudah ditransmisikan tersebut kemudian dikirimkan ke server cloud. Cloud yang sudah terkoneksi dengan internet ini juga akan memberikan layanan dan aplikasi yang diperlukan untuk mengelola IoT.

Dengan begitu, user bisa langsung memberikan perintah kepada sebuah perangkat untuk melakukan sesuatu dengan mengakses data dari cloud. Gambar 1.1 menunjukkan konsep dasar dari IoT.



Gambar 1-1 Konsep dasar kerja IoT (Annisa, 2021)

### 1.2.2 Keuntungan Menggunakan IoT

Kemudahan yang ditawarkan oleh IoT tentu memberikan manfaat bagi penggunanya. Berikut adalah keuntungan menggunakan IoT yang bisa Anda rasakan:

#### 1. Efisiensi energi

Konsep IoT bisa digunakan pada berbagai aspek hidup manusia. Mulai dari Agrindustri, pendidikan, kesehatan, industri, hingga rumah tangga. Tujuan dari penggunaan konsep ini pada

bidang-bidang tertentu salah satunya adalah untuk efisiensi energi. Hal ini lantaran IoT bisa meningkatkan efisiensi aktivitas pengguna serta mengurangi biaya operasional dan konsumsi sumberdaya.

Selain itu, IoT juga dapat mengoptimalkan penggunaan sumber daya dengan memberikan kontrol lebih baik atas perangkat yang digunakan. Dengan kata lain, seorang user bisa mengatur batas minimal dan maksimal penggunaan sumber daya atau perangkat agar tidak melebihi kemampuan yang ditentukan.

## 2. Hemat biaya

Keuntungan lainnya yang diberikan oleh konsep IoT adalah bisa mengurangi biaya operasional sebuah perusahaan atau bisnis. Konsep ini juga memungkinkan adanya pemeliharaan perangkat dengan memantau dan menganalisis data secara real-time.

## 3. Produktivitas meningkat

Dengan adanya sistem kerja yang kompleks seperti penggunaan sensor, konsep ini akan memudahkan user untuk memberikan perintah dan melakukan aktivitas. Proses akses yang diterima dan dihasilkan oleh IoT bekerja dengan cepat dan tepat sehingga user bisa lebih praktis dalam penggunaannya.

### 1.2.3 Tantangan IoT

Segala kelebihan yang ditawarkan oleh IoT tentunya tidak bisa menghindar dari adanya tantangan. Tantangan ini bisa datang dari dalam atau luar konsep itu sendiri. Berikut adalah beberapa tantangannya:

#### 1. Keamanan data privasi

Seperti yang sudah dibahas pada bagian sebelumnya, sistem IoT terhubung dengan jaringan internet dan perangkat-perangkat lainnya. Hubungan ini tentunya bisa memberikan celah bagi para pelaku kejahatan siber untuk melancarkan aksi peretasan IoT.

Perangkat IoT terkadang memiliki sistem keamanan yang lemah dan tidak memadai. Hal inilah yang menyebabkan maraknya kasus pencurian data sensitif oleh pihak tidak bertanggung jawab. Untuk itu, diperlukan upaya perlindungan data dengan sistem keamanan enkripsi atau yang lainnya.

#### 2. Regulasi rendah

Tidak hanya dari sistem IoT yang memungkinkan terjadinya tantangan, dari pihak luar pun bisa memengaruhi kinerja sebuah sistem.

Di setiap negara, penggunaan teknologi terbaru yang

menggunakan konsep IoT memerlukan regulasi dari pemerintah. Tujuan regulasi ini adalah untuk menjamin adanya keterikatan hukum tertentu apabila terjadi permasalahan di masa mendatang.

Kendati demikian, tidak semua perangkat IoT memiliki regulasi yang kuat di sebuah negara karena adanya ketidaksetujuan atau masih perlu peninjauan kembali. Jadi, penerapan IoT pun menjadi terhambat.

### 1.3 Implementasi IoT di Sektor Agroindustri

Berikut ini adalah beberapa contoh penerapan Internet of Things (IoT) di bidang Agroindustri:

#### 1.3.1 Penggunaan Field Monitoring System (FMS) untuk Agroindustri

Field monitoring system (FMS) adalah teknologi yang digunakan untuk memantau kondisi tanaman dan kondisi iklim di sawah. Kondisi tanaman dan iklim tersebut direkam dan diolah menggunakan sensor secara real-time, gambar 1.2 menunjukkan gambaran FMS.



Gambar 1-2 Field Monitoring System dalam Agroindustri (Manager, 2018)

Hasil data dari sensor ini, seperti tingkat sinar matahari, tingkat hujan, kecepatan angin, dan lain sebagainya akan direkam di cloud dan disampaikan kepada petani dalam bentuk data yang muncul di aplikasi. Dengan demikian, petani bisa mengamati kondisi sawah meskipun sedang tidak berada di rumah.

Harapannya adalah dengan data-data tersebut serta pengetahuan yang diperoleh petani selama bekerja dapat digabungkan menjadi satu untuk menghasilkan keputusan terkait Agroindustri yang lebih tepat.

### 1.3.2 Penggunaan drone untuk Agroindustri

Drone juga merupakan alat yang berguna untuk membantu petani. Pesawat terbang mini ini dapat digunakan untuk memetakan kondisi tanaman, melacak hewan, menyebarkan irigasi dan menyebarkan pestisida.



Gambar 1-3 Penggunaan drone dalam Agroindustri (Chusna, 2023)

Drone tersebut dikontrol menggunakan aplikasi khusus yang juga dapat digunakan untuk menerima data sensor. Dengan demikian, petani tidak perlu mengelilingi sawah atau kebun untuk membasmi hama, atau menyemprot tanaman.

### 1.3.3 Smart greenhouse

Saat ini produksi Agroindustri tidak hanya berpusat di sawah maupun kebun, tetapi juga berpusat pada ruang khusus yang disebut dengan greenhouse (greenhouse farming). Greenhouse farming adalah pengembangan tanaman di ruang khusus yang dilingkupi dengan material transparan. Tujuannya adalah supaya tanaman terhindar dari perubahan cuaca yang berlebihan dan hama.





Gambar 1-4 Green house dengan menerapkan IoT (Yulianti, 2022)

Penerapan inovasi Internet of Things (IoT) di sektor ini adalah dengan menggunakan satu aplikasi terintegrasi untuk mengendalikan pencahayaan, suhu, kelembaban udara dan berbagai kebutuhan Agroindustri lainnya di area ini. Dengan demikian, petani tidak perlu mengeliling area untuk memeriksa kondisi tanaman maupun memberikan pupuk dan pestisida.

#### 1.3.4 Livestock management

Tidak hanya pada tanaman, teknologi IoT juga dapat diterapkan pada livestock management atau pengelolaan peternakan. Contohnya adalah penanaman chip pada sapi yang dilakukan oleh sebuah perusahaan peternakan di Tel Aviv, Israel.



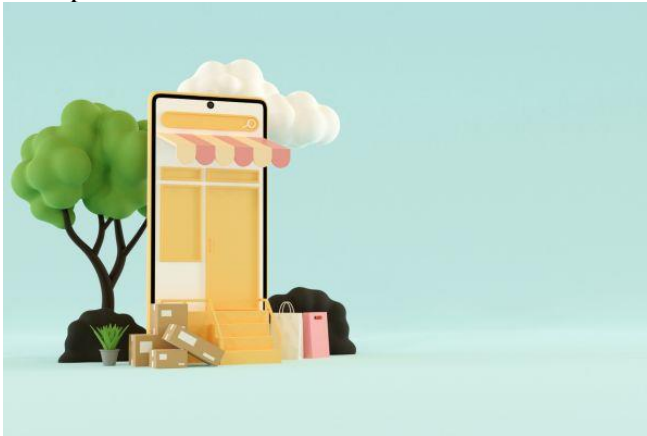
Gambar 1-5 Penerapan IoT dalam Live Stock Management ternak (Chusna, 2023)

Dengan adanya chip pada tubuh sapi tersebut, perusahaan bisa memantau kondisi dan perilaku hewan ternak tersebut melalui sistem komputer, sehingga ketika perilaku aneh maupun kondisi yang tidak

wajar, perusahaan bisa segera memisahkan sapi tersebut dari sapi lainnya.

### 1.3.5 Aplikasi pendukung penjualan dan edukasi

Salah satu tantangan terbesar dari sektor Agroindustri adalah masalah supply chain, mulai dari pembelian bibit dan pupuk, sampai akses ke konsumen akhir. Petani, khususnya di Indonesia, umumnya menjual hasil Agroindustri mereka kepada pengepul dan dari pengepul kepada pengecer. Ditambah dengan harga bibit dan pupuk yang semakin mahal, jumlah keuntungan bersih yang diperoleh petani semakin tipis.



Oleh sebab itu, dua aplikasi penting yang dibutuhkan oleh petani saat ini adalah aplikasi pendukung penjualan yang bisa membantu mereka memotong rantai pasok hingga ke konsumen akhir dan aplikasi edukasi untuk membantu mereka lebih memahami cara penggunaan teknologi di bidang Agroindustri.

## 1.4 Keuntungan Penggunaan IoT dalam Agroindustri

Adapun beberapa keuntungan penggunaan Internet of Things (IoT) dalam bidang Agroindustri adalah sebagai berikut:

### 1.4.1 Efisiensi produksi

Penggunaan teknologi IoT di sektor Agroindustri dapat meningkatkan efisiensi produksi, karena:

Petani dan peternak tidak perlu berkeliling lahan untuk menyemprot pestisida, mengalirkan air dan memberi makan ternak, sehingga lebih hemat waktu dan tenaga. Sebaliknya, waktu

dan tenaga tersebut dapat dialihkan untuk menambah pengetahuan di bidang Agroindustri dan manajemen, atau untuk mencari pekerjaan lain sambil menunggu panen.

Agroindustri dapat menyesuaikan dengan perubahan cuaca secara real time. Dengan semakin tidak menentunya musim di Indonesia, penggunaan IoT di bidang Agroindustri semakin dibutuhkan untuk memantau cuaca secara real time.

Petani dan peternak tidak perlu khawatir soal kesehatan tanaman dan ternak. Pemantauan tanaman secara teratur dan presisi serta penanaman chip pada ternak membantu petani dan peternak untuk segera mengetahui jika ada hama yang menyerang tanaman mereka.

Ketiga manfaat di atas kemudian merujuk pada satu manfaat utama, yaitu jumlah hasil Agroindustri yang memadai dan berkualitas bagus.

#### 1.4.2 Efisiensi distribusi

Dengan bantuan Internet of Things (IoT), petani bisa menjangkau konsumen akhir secara langsung dengan tanpa melewati pengepul dan pedagang. Di satu sisi, hal ini akan meningkatkan pendapatan petani, dan di sisi lain konsumen berpeluang untuk membeli produk Agroindustri dengan harga yang lebih terjangkau karena langsung dari petani.

#### 1.4.3 Tantangan Penerapan Smart Farming di Indonesia

Meskipun smart farming menawarkan berbagai manfaat yang menguntungkan bagi sektor agrikultur di Indonesia, namun penerapan teknologi ini harus menghadapi berbagai tantangan, diantaranya:

##### Skala ekonomi

Mayoritas petani dan peternak di Indonesia adalah petani dan peternak skala mikro (gurem), yang mengelola lahan atau ternak dalam jumlah yang sedikit. Bahkan, banyak juga yang hanya merupakan petani pengelola, sementara lahan yang diolahnya dimiliki oleh orang lain yang lebih mampu secara ekonomi. Akibatnya penggunaan berbagai alat teknologi canggih, seperti sensor atau drone yang cukup mahal, terbilang kurang efektif karena mahal dan hanya digunakan untuk area yang sempit.

##### Edukasi dan regenerasi

Seperti yang telah disinggung pada paragraf pembuka di atas bahwasanya petani adalah pekerjaan yang membutuhkan usaha keras, namun memiliki hasil yang sedikit. Hal ini berakibat pada

minimnya generasi muda dan dominasi generasi paruh baya yang menggeluti bidang ini.

Masalah regenerasi ini kemudian mengembang pada masalah edukasi dan literasi, yaitu:

Petani tentu akan kesulitan merumuskan strategi produksi dan distribusi yang tepat apabila mereka tidak bisa membaca dan menganalisis data yang dipresentasikan dalam bentuk kurva.

Aplikasi online marketplace merupakan solusi dari permasalahan nomor 1 di atas. Dengan aplikasi ini, petani bisa menjual produk mereka secara langsung kepada pelanggan. Akan tetapi, pembentukan aplikasi ini sendiri juga memiliki tantangan. Tantangan yang pertama adalah tidak semua petani memiliki literasi digital yang tinggi, oleh sebab itu dibutuhkan adanya edukasi intensif kepada petani dan keluarganya mengenai tata cara menggunakan online marketplace. Tantangan yang kedua adalah tersedianya dedicated internet yang memadai untuk operasional dan edukasi petani.

## **1.5 Rangkuman**

Internet of Things adalah sebuah konsep yang terhubung dengan perangkat sebagai media komunikasi berbasis internet. Konsep ini sepintas hampir serupa dengan M2M. M2M di sini merujuk pada teknologi yang memungkinkan komunikasi antara mesin-mesin tanpa melibatkan campur tangan manusia. Dengan kata lain, M2M lebih berfokus pada sistem kerja mesin untuk menjalankan sebuah program.

Selain itu, IoT juga dapat mengoptimalkan penggunaan sumber daya dengan memberikan kontrol lebih baik atas perangkat yang digunakan. Dengan kata lain, seorang user bisa mengatur batas minimal dan maksimal penggunaan sumber daya atau perangkat agar tidak melebihi kemampuan yang ditentukan. Keuntungan lainnya yang diberikan oleh konsep IoT adalah bisa mengurangi biaya operasional sebuah perusahaan atau bisnis.

Segala kelebihan yang ditawarkan oleh IoT diantaranya untuk:

Field monitoring system adalah teknologi yang digunakan untuk memantau kondisi tanaman dan kondisi iklim di sawah.

Drone juga merupakan alat yang berguna untuk membantu petani. Drone tersebut dikontrol menggunakan aplikasi khusus yang juga dapat digunakan untuk menerima data sensor.

Livestock management, contohnya adalah penanaman chip pada sapi yang dilakukan oleh sebuah perusahaan peternakan di Tel Aviv, Israel.

Agroindustri dapat menyesuaikan dengan perubahan cuaca secara real time. Dengan semakin tidak menentunya musim di Indonesia, penggunaan IoT di bidang Agroindustri semakin dibutuhkan untuk memantau cuaca secara real time. Pemantauan tanaman secara teratur dan presisi serta penanaman chip pada ternak membantu petani dan peternak untuk segera mengetahui jika ada hama yang menyerang tanaman mereka.

## 1.6 Latihan Soal

Soal-soal ini dirancang untuk menguji pemahaman mahasiswa tentang konsep, cara kerja, keuntungan, dan tantangan IoT serta aplikasinya dalam agroindustri. Mahasiswa diharapkan dapat menjawab dengan mengacu pada materi yang telah dipelajari

1. **Konsep Dasar IoT:** a. Jelaskan secara singkat apa yang dimaksud dengan Internet of Things (IoT). b. Sebutkan dan jelaskan tiga komponen utama dalam sistem IoT. c. Apa perbedaan utama antara konsep IoT dan M2M (Machine-to-Machine)?

**Spesifikasi Teknis Mikrokontroler:** a. Sebutkan tiga contoh mikrokontroler yang sering digunakan dalam proyek IoT dan jelaskan spesifikasi teknis dasar dari masing-masing. b. Mengapa pemilihan mikrokontroler yang tepat sangat penting dalam proyek IoT?

**Teknologi dan Protokol Komunikasi IoT:** a. Jelaskan apa yang dimaksud dengan protokol komunikasi MQTT dan bagaimana cara kerjanya dalam IoT. b. Apa perbedaan antara protokol CoAP dan HTTP dalam konteks IoT?

**Cara Kerja IoT:** a. Bagaimana sensor, gateway, dan cloud berinteraksi dalam sebuah sistem IoT? b. Berikan contoh aplikasi IoT di rumah tangga dan jelaskan cara kerjanya.

**Keuntungan Penggunaan IoT:** a. Jelaskan tiga keuntungan utama dari penggunaan IoT dalam kehidupan sehari-hari. b. Bagaimana IoT dapat membantu dalam efisiensi energi?

**Tantangan IoT:** a. Sebutkan dan jelaskan dua tantangan utama dalam implementasi IoT. b. Bagaimana keamanan data dan privasi dapat ditingkatkan dalam sistem IoT?

**Implementasi IoT dalam Agroindustri:** a. Jelaskan bagaimana Field Monitoring System (FMS) dapat membantu petani. b. Bagaimana drone digunakan dalam agroindustri untuk meningkatkan efisiensi? c. Apa itu

smart greenhouse dan bagaimana IoT berperan dalam sistem ini? d. Berikan contoh bagaimana IoT digunakan dalam livestock management.

**Keuntungan Penggunaan IoT dalam Agroindustri:** a. Jelaskan bagaimana IoT dapat meningkatkan efisiensi produksi di sektor agroindustri. b. Bagaimana IoT dapat membantu dalam efisiensi distribusi hasil pertanian?

**Tantangan Penerapan Smart Farming di Indonesia:** a. Sebutkan dua tantangan utama dalam penerapan smart farming di Indonesia dan jelaskan mengapa tantangan tersebut signifikan. b. Bagaimana aplikasi online marketplace dapat membantu petani dalam menjual produk mereka?

## 1.7 Kunci Jawaban Singkat:

1. Konsep Dasar IoT: a. IoT adalah jaringan perangkat fisik yang terhubung ke internet, memungkinkan mereka untuk mengumpulkan dan bertukar data. b. Sensor, Gateway, dan Cloud. c. IoT mencakup lebih banyak perangkat dan aplikasi yang lebih luas dibandingkan M2M, yang lebih berfokus pada komunikasi antar mesin tanpa intervensi manusia.

Spesifikasi Teknis Mikrokontroler: a. Contoh: Arduino (ATmega328P), Raspberry Pi (BCM2837), ESP8266 (Tensilica L106). b. Mikrokontroler yang tepat memastikan efisiensi, kompatibilitas, dan kinerja yang optimal dalam proyek IoT.

Teknologi dan Protokol Komunikasi IoT: a. MQTT adalah protokol komunikasi yang ringan, menggunakan model publish-subscribe untuk mengirim pesan. b. CoAP lebih efisien dan cocok untuk perangkat terbatas dibandingkan HTTP yang lebih berat.

Cara Kerja IoT: a. Sensor mengumpulkan data, gateway mentransmisikan data ke cloud, cloud mengolah dan menyediakan layanan. b. Contoh: Smart thermostat yang mengatur suhu rumah berdasarkan data sensor dan kontrol dari aplikasi.

Keuntungan Penggunaan IoT: a. Efisiensi energi, hemat biaya, dan peningkatan produktivitas. b. IoT mengoptimalkan penggunaan perangkat dan sumber daya melalui kontrol otomatis dan pemantauan real-time.

Tantangan IoT: a. Keamanan data privasi dan regulasi rendah. b. Dengan enkripsi data, otentikasi yang kuat, dan pemantauan keamanan yang berkelanjutan.

Implementasi IoT dalam Agroindustri: a. FMS memantau kondisi tanaman dan iklim secara real-time, membantu petani dalam pengambilan keputusan. b. Drone memetakan lahan, melacak hewan, dan menyebarkan pestisida secara efisien. c. Smart greenhouse menggunakan IoT untuk

mengontrol lingkungan tanaman secara otomatis. d. Chip pada ternak untuk memantau kesehatan dan perilaku hewan secara real-time.

Keuntungan Penggunaan IoT dalam Agroindustri: a. Meningkatkan efisiensi produksi dengan otomatisasi dan pemantauan real-time. b. Menghubungkan petani langsung dengan konsumen akhir, mengurangi biaya distribusi.

Tantangan Penerapan Smart Farming di Indonesia: a. Skala ekonomi dan edukasi petani. b. Membantu petani menjual produk langsung ke konsumen dan meningkatkan literasi digital petani.

## 1.8 Bahan Diskusi

Carilah topik terkait Agroindustri yang dimungkinkan untuk penerapan IoT, dapat dikelompokkan di sisi proses produksi, monitoring, control atau disisi pemasaran dan supply chainnya. Diskusikan dan buat dalam bentuk narasi gambaran teknologi yang akan dilakukan.

## 1.9 Daftar Rujukan

- Annisa, S. (2021, November 15). *Apa itu IoT? Cara Kerja, Tujuan dan Manfaatnya di Beberapa Industri*. Retrieved from Niaga Hoster: <https://www.niagahoster.co.id/blog/iot-adalah/>
- Chusna, F. (2023, Mei 10). <https://www.linknet.id>. Retrieved from contoh-iot-dalam-bidang-pertanian: <https://www.linknet.id/article/contoh-iot-dalam-bidang-pertanian>
- Manager, T. C. (2018, Oktober 3). <https://www.topcropmanager.com>. Retrieved from New field monitoring system from Hoskins Scientific: <https://www.topcropmanager.com/new-field-monitoring-system-from-hoskins-scientific-21485/>
- Yulianti, S. (2022, Februari 23). <https://www.indoniaga.co.id>. Retrieved from pertanian-green-house-berbasis-iot: <https://www.indoniaga.co.id/2022/02/pertanian-green-house-berbasis-iot.html>

## **BAB 2.      PENGENALAN PERANGKAT IOT**

### **Capaian Pembelajaran Matakuliah (CPMK)**

Mahasiswa mampu memahami perangkat perangkat yang digunakan dalam sistem iot, spesifikasi perangkat, serta penggunaannya.

### **2.1    Pendahuluan**

Internet of Things (IoT) mengacu pada jaringan kolektif perangkat yang terhubung sekaligus teknologi yang terdapat komunikasi antara perangkat dan cloud, dan antar perangkat. Ini mencakup berbagai jenis perangkat, mulai dari sensor kecil hingga perangkat pintar yang canggih. Perangkat IoT dirancang untuk mengumpulkan data dari lingkungan mereka, berkomunikasi satu sama lain, dan melakukan tindakan berdasarkan data yang dikumpulkan.

**Karakteristik Utama Perangkat IoT**

1. **Koneksi ke Internet:** Perangkat IoT memiliki kemampuan untuk terhubung ke internet, yang memungkinkan mereka untuk mengirim dan menerima data.

**Sensors:** Sebagian besar perangkat IoT dilengkapi dengan berbagai jenis sensor untuk mengumpulkan data dari dunia nyata. Ini termasuk sensor suhu, kelembaban, gerakan, cahaya, dan banyak lagi.

**Kemampuan Pemrosesan:** Beberapa perangkat IoT memiliki kemampuan pemrosesan yang terbatas, sementara yang lain bergantung pada cloud computing untuk analisis data.

**Komunikasi:** Perangkat IoT dapat berkomunikasi dengan perangkat lain melalui jaringan yang terhubung.

**Kemampuan Pemantauan dan Kontrol:** Perangkat IoT dapat digunakan untuk memantau lingkungan, mengontrol perangkat lain, atau memberikan notifikasi kepada pengguna.

### **2.2    Jenis-jenis Perangkat IoT**

Dalam sebuah sistem IoT (Internet of Things), terdapat berbagai jenis perangkat yang berperan penting untuk memastikan fungsi dan integrasi yang efektif. Berikut adalah beberapa jenis perangkat yang



umumnya ada dalam sistem IoT:

### 2.2.1 Sensor

Sensor adalah perangkat yang digunakan untuk mendeteksi dan mengukur perubahan dalam lingkungan fisik atau kimia dan mengubah informasi tersebut menjadi sinyal yang dapat dibaca oleh pengguna atau perangkat lain. Sensor berfungsi sebagai "indera" dari perangkat atau sistem, memungkinkan mereka untuk merespons perubahan dalam lingkungan.

Fungsi dan Jenis Sensor

Fungsi Sensor:

1. Deteksi: Sensor mendeteksi perubahan dalam lingkungan fisik atau kimia, seperti suhu, tekanan, cahaya, atau keberadaan objek.
2. Pengukuran: Sensor mengukur besaran dari perubahan yang terdeteksi dan mengubahnya menjadi sinyal yang dapat diproses lebih lanjut.
3. Transmisi Data: Sensor mengirimkan data yang dikumpulkan ke perangkat lain, seperti mikrokontroler, komputer, atau sistem cloud, untuk dianalisis dan diinterpretasikan.

Jenis-jenis Sensor:

#### 1. Sensor Suhu

Mengukur suhu lingkungan atau objek. Contohnya adalah termokopel, RTD (*Resistance Temperature Detector*), thermistor, sensor berbasis IC dan Infrared.

Berbagai jenis sensor suhu telah dikembangkan untuk memenuhi kebutuhan spesifik dari aplikasi yang berbeda. Pemilihan sensor suhu yang tepat dapat menentukan keberhasilan dalam memonitor dan mengontrol proses penting. Berikut ini adalah penjelasan mendalam mengenai jenis sensor suhu yang paling umum digunakan.

##### a) Thermokopel

Termokopel adalah sensor suhu yang bekerja berdasarkan efek Seebeck dengan menghasilkan tegangan listrik ketika ada perbedaan suhu antara dua logam berbeda. Kelebihanannya adalah dapat mengukur rentang suhu yang sangat luas dengan membuatnya ideal untuk aplikasi industri. Termokopel relatif murah dan responsif, tetapi akurasi lebih rendah dibandingkan dengan beberapa jenis sensor suhu lainnya. Modul sensor suhu dengan termokopel ditunjukkan

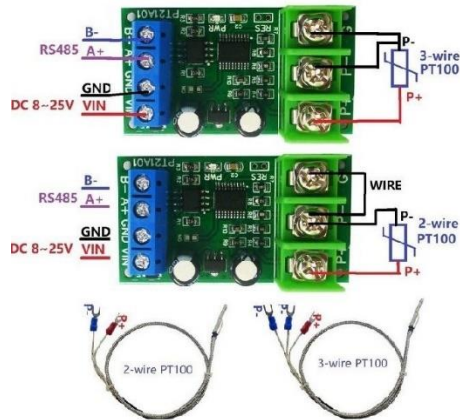
dalam gambar dibawah ini merupakan K Type Thermocouple Sensor MAX6675 High Temperatur merupakan module sensor suhu dengan kemampuan pengukuran suhu sangat tinggi, dimana probe nya mampu mengukur suhu dengan rentang 0C - 400C. probe terbuat dari stainless steel.



Gambar 2-1 Modul sensor suhu dengan thermocouple

b) RTD (*Resistance Temperature Detector*)

RTD memanfaatkan perubahan resistansi logam murni sebagai fungsi dari suhu. Keakuratan dan keandalan RTD sangat tinggi, terutama pada rentang suhu yang lebih rendah dibanding termokopel. RTD memerlukan sumber arus eksternal untuk operasinya dan biasanya lebih mahal daripada termokopel. Namun, kestabilan jangka panjang dan linearitas responsnya menjadikan RTD pilihan yang sangat diandalkan. Modul sensor suhu dengan RTD ditunjukkan dalam gambar 2- Rtd Pt100 Transmitter Rs485



Gambar 2-2 Modul sensor temperature Sensor Rs485 dengan RTD

#### Fitur Modul

- 1) Tegangan Kerja : DC 6-25V (12V recommended)
- 2) Komunikasi : RS485 (modbus protokol 03 dan 06)
- 3) Kompatibel Sensor : Sensor PT100 3 wire atau 2 wire (wiring ada di gambar kedua)
- 4) Pembacaan temperature :  $-20^{\circ} \sim +400^{\circ}\text{C}$
- 5) Suport parallel :247 modul

#### c) Termistor

Termistor adalah jenis resistor yang resistansinya berubah secara signifikan dengan perubahan suhu. Termistor menawarkan respon yang sangat sensitif terhadap perubahan suhu tetapi pada rentang yang lebih terbatas. Termistor tersedia dalam dua jenis: NTC (*Negative Temperature Coefficient*) dan PTC (*Positive Temperature Coefficient*). Oleh karena sensitivitas kecil, termistor ideal untuk aplikasi pengukuran suhu di lingkungan terkontrol. Modul sensor dengan thermistor ditunjukkan dalam gambar 2-1.



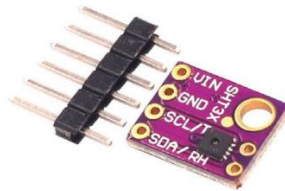
Gambar 2-3 Modul sensor temperature dengan thermistor

#### Fitur Modul:

- 1) menggunakan sensor termistor NTC, sensitivitas yang baik
- 2) sinyal keluaran komparator bersih, bentuk gelombang bagus, kemampuan mengemudi, lebih dari 15mA.
- 3) sesuaikan ambang deteksi posisi distribusi suhu dengan potensiometer
- 4) tegangan kerja 3.3V-5V
- 5) format keluaran: keluaran peralihan digital (0 dan 1)
- 6) ukuran papan PCB kecil: 3.2cm x 1.4cm
- 7) menggunakan komparator LM393 tegangan lebar

d) Sensor berbasis IC (*Integrated Circuit*)

Jenis sensor suhu berbasis IC menggunakan sirkuit terintegrasi yang dirancang khusus untuk pengukuran suhu dengan output digital atau analog. Sensor suhu berbasis IC sangat cocok untuk monitoring suhu di data center karena menawarkan akurasi yang tinggi, output yang stabil, dan kemudahan integrasi dengan sistem pemantauan. Sensor ini menghasilkan output digital yang dapat diproses oleh sistem manajemen data center (DCIM). Modul sensor suhu berbasis IC umumnya juga sebagai sensor kelembaban. (Sht30-D Modul ).Gambar 2- merupakan sensor suhu dan kelembaban dengan IC.



Gambar 2-4 Sht30-D Modul I2C Sensor Suhu Dan Kelembaban

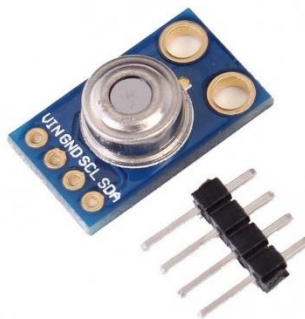
Spesifikasi modul

- Pin Daya: Vin – daya masukan. Chip ini dapat menggunakan 2,5-5VDC untuk daya. Untuk memberi daya pada modul, berikan daya yang sama dengan tingkat logika mikrokontroler - mis. untuk mikro 5V seperti Arduino, gunakan 5V. Untuk pengontrol 3.3V seperti Raspberry Pi, sambungkan ke 3.3V
- GND - landasan bersama untuk kekuatan dan logika
- Pin Logika I2C:
- SCL - Pin jam I2C, sambungkan ke jalur jam I2C mikrokontroler Anda.

- Pin ini memiliki resistor pullup 10K ke Vin
- SDA - pin data I2C, sambungkan ke jalur data I2C mikrokontroler Anda.
- Pin ini memiliki resistor pullup 10K ke Vin
- Pin Lainnya:
- ADR - Ini adalah pin pemilihan alamat I2C. Pin ini memiliki resistor pull down 10K untuk membuat alamat I2C default 0x44. Anda dapat mengikat pin ini ke Vin untuk membuat alamat 0x45
- RST - Pint penyetelan ulang perangkat keras. Memiliki pullup 10K untuk membuat chip aktif secara default. Hubungkan ke ground untuk melakukan reset perangkat keras!
- ALR - Keluaran peringatan/interrupt. Anda dapat mengatur sensor untuk memperingatkan Anda ketika suatu peristiwa telah terjadi.

#### e) Inframerah (IR)

Selanjutnya adalah sensor suhu inframerah yang menawarkan kemampuan unik untuk pemantauan suhu tanpa kontak. Dalam data center, keseimbangan distribusi udara dingin dan panas sangat penting. Sensor IR dapat memberikan gambaran instan tentang efektivitas konfigurasi cooling. Meskipun sangat berguna untuk analisis dan pemecahan masalah, sensor IR lebih sering digunakan sebagai pelengkap sistem monitoring suhu utama. Gambar 2- merupakan modul sensor suhu dengan infrared MLX90614.



Gambar 2-5 Modul sensor suhu dengan infrared

MLX90614 adalah sebuah modul termometer inframerah yang didesain supaya dapat melakukan pengukuran suhu obyek tanpa harus menyentuh permukaan obyek secara langsung. Disertai DSP dengan resolusi ADC 17-bit membuatnya dapat membaca suhu benda dengan sangat akurat.

Untuk output datanya sendiri dapat diakses melalui mikrokontroler melalui antarmuka I2C

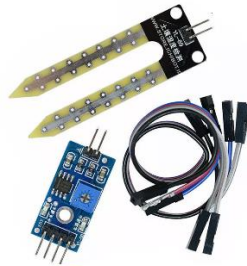
Spesifikasi modul

- Catu Daya: 3-5V
- Dimensi: 16x12mm
- Resolusi Pembacaan: 0.02°C
- ADC: 17-bit
- Jarak Pembacaan Suhu: 5 - 10 cm
- Antarmuka : I2C
- Jangkauan Pengukuran Suhu: -40° ~ 125°C untuk pengukuran suhu dan -70° ~ 380°C untuk pengukuran suhu obyek

## 2. Sensor Kelembaban

Mengukur tingkat kelembaban tanah atau kelembaban udara. Contohnya adalah hygrometer atau sensor kapasitif untuk mengukur kelembaban tanah, sedangkan untuk mengukur kelembaban udara menggunakan *DHT22 Temperature and Humidity module*.

Sensor hygrometer merupakan sensor kelembaban yang dapat digunakan untuk mendeteksi kelembaban tanah. Gambar 2-



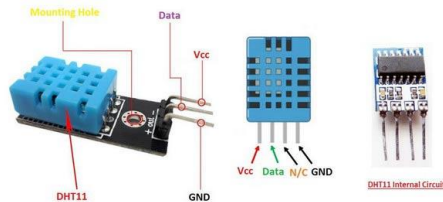
Gambar 2-6 Modul sensor hygrometer

### SPESIFIKASI

- Adjustable sensitivity (shown in blue digital potentiometer adjustment)
- Operating Voltage 3.3V-5V

- module dual output mode, the digital output is simple, analog output is more accurate.
- with fixed bolt hole for easy installation
- small PCB board size: 3cm 1.6cm
- Small plate interface specification (4-wire)
- VCC 3.3V-5V external
- GND GND External
- DO small board digital output interface (0 and 1)
- AO small plates analog output interface

Sedangkan untuk membaca besaran kelembaban udara, kita memakai alat Higrometer, dan pada aplikasi Arduino biasanya memakai sensor DHT11, SHT11 atau DHT22. DHT11 selain dapat membaca kelembaban, DHT11 juga mampu membaca temperatur lingkungan. Gambar 2- merupakan DHT11



Gambar 2-7 Modul sensor DHT 11

Spesifikasi sensor suhu kelembaban DHT11 :

- Tegangan input : 3,5 – 5 VDC
- Sistem komunikasi : Serial (single – Wire Two way)
- Range suhu : 0 °C – 50°C
- Range kelembaban : 20% – 90% RH
- Akurasi :  $\pm 20^{\circ}\text{C}$  (temperature)  $\pm 5\%$  RH (humidity)

Datasheet selengkapnya bisa dilihat disini. Sensor suhu dan kelembaban DHT11 terdiri dari 4 kaki/pin, tetapi yang dipakai hanya 3 pin saja. Biasanya kalau kita membeli dalam bentuk modul jumlah pin-nya menjadi 3 :

- VCC(+) : tegangan input (5V)
- GND(-) : Ground
- DOUT : Data output serial

### 3. Sensor Cahaya

Sensor cahaya bekerja dengan memanfaatkan prinsip arus dan hambatan listrik. Ketika cahaya mengenai sensor, maka hambatan tersebut akan meningkat atau menurun sesuai posisinya sehingga mengatur arus. Berdasarkan jenisnya, prinsip cahaya diklasifikasikan menjadi dua jenis yakni: Sensor pasif, sensor yang akan menerima cahaya kemudian akan meningkatkan atau mengurangi resistensinya. Sehingga arus listrik akan mengalir atau terputus yang kemudian disambungkan pada lampu atau alarm. Contoh sensor pasif yaitu *Light Dependent Resistor* (LDR) Sensor aktif, ketika terpapar cahaya, sensor ini akan mengubah cahaya menjadi energi lain yang lebih berguna, contoh sensor ini adalah solar cell (Yusro & Diamah, 2019) (Sulistiyanti & Purwiyanti, 2020).

Jenis sensor cahaya

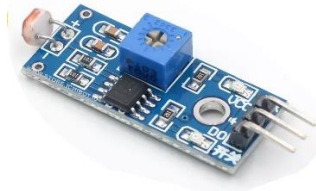
Sensor cahaya sangat luas penggunaannya, adapun jenis-jenis sensor cahaya, antara lain:

a) Detektor kimiawi

seperti pelat fotografis, di mana molekul silver halida dibagi menjadi sebuah atom perak metalik dan atom halogen.

b) Fotoresistor atau *Light Dependent Resistor* (LDR)

Sensor Cahaya LDR (*Light Dependent Resistor*) adalah salah satu jenis resistor yang dapat mengalami perubahan resistansinya apabila mengalami perubahan penerimaan cahaya. Modul sensor cahaya bekerja menghasilkan output yang mendeteksi nilai intensitas cahaya. Perangkat ini sangat cocok digunakan untuk project yang berhubungan dengan cahaya seperti nyala mati lampu. Gambar 2- menunjukkan modul sensor Cahaya LDR.



Gambar 2-8 Modul Sensor LDR Modul Sensor Cahaya

Spesifikasi singkat: Supply : 3.3 V – 5 V (arduino available)  
Output Type: Digital Output (0 and 1) – Inverse output – Include



IC LM393 voltage comparator Sensitivitasnya dapat diatur  
Dimensi PCB size: 3.2 cm x 1.4 cm

c) Sel fotovoltaiik

Panel surya, yang juga dikenal sebagai modul surya atau photovoltaic, merupakan perangkat yang mampu mengubah energi dari sinar matahari menjadi energi listrik. Panel surya terdiri dari sejumlah sel surya yang dihubungkan secara seri. Setiap sel surya umumnya mampu menghasilkan tegangan keluaran sekitar 0,5 hingga 0,6 volt. Ketika 32 hingga 36 sel surya dihubungkan secara seri, ini akan menghasilkan tegangan sekitar 16 volt, yang cukup untuk mengisi akumulator 12 volt. Sejumlah modul photovoltaic yang diatur secara seri disebut sebagai array, yang memiliki kapasitas untuk menghasilkan daya yang lebih besar (Harmini & Nurhayati, 2020)



Gambar 2-9 Modul sel photovoltaik

d) Fotolistrik

Sensor yang berprinsip kerja berdasarkan pantulan karena perubahan posisi/jarak suatu sumber sinar (inframerah atau laser). Modul penerimaan inframerah mengadopsi 1838 cahaya inframerah, interferensi elektromagnetik yang kuat, built-in IC khusus inframerah, dapat bekerja di bawah intensitas cahaya 500 lux. Banyak digunakan dalam: stereo, TV, mesin video, mesin disk, kotak set-top, bingkai foto digital, stereo mobil, mainan remote control, penerima satelit, pemutar hard disk, AC, pemanas, kipas angin listrik, pencahayaan dan peralatan rumah lainnya



Gambar 2-10 Modul Sensor Infrared Receiving KY-022

#### SPESIFIKASI :

Dimensi: 6,4 x 7,4 x 5,1 mm

Menerima sudut: 90 ° nTegangan kerja: 2.7 ~ 5.5V

Frekuensi: 37,9 KHz

Kisaran penerimaan: 18m

#### e) Fotodioda,

berfungsi untuk mengubah intensitas cahaya menjadi konduktivitas dioda.



Gambar 2-11 modul photodioda

#### f) Detektor optis

yang berlaku seperti termometer, secara murni tanggap terhadap pengaruh panas dari radiasi yang masuk, seperti detektor piroelektrik, sel Golay, termokopel dan termistor, tetapi kedua yang terakhir kurang sensitif. Modul sensor Inframerah Detektor Api suhu Flame Module Detection ditunjukkan dam gambar 2-



Gambar 2-12 Modul sensor suhu dengan infrared

Flame sensor atau Sensor Api merupakan sensor yang mempunyai fungsi sebagai pendeteksi nyala api yang dimana api tersebut memiliki panjang gelombang antara 760nm – 1100nm. Sensor ini menggunakan infrared sebagai transduser dalam mendeteksi kondisi nyala api

Spesifikasi:

- Supply Voltage: 3.3-5V
- Spectrum range: 760nm ~ 1100nm
- Detection angle: 0 - 60 degree
- Output: Digital
- Sensitivity: Adjustable: Yes
- Operating temperature: -25 degrees Celsius ~ 85 degrees Celsius

#### 4. Sensor Gerak

Mendeteksi pergerakan objek. Contohnya adalah sensor PIR (*Passive Infrared*) dan sensor ultrasonik. Sensor PIR adalah sebuah sensor yang menangkap pancaran sinyal inframerah yang dikeluarkan oleh tubuh manusia maupun hewan. Sensor PIR dapat merespon perubahan-perubahan pancaran sinyal inframerah yang dipancarkan oleh tubuh manusia.



Gambar 2-13 Modul sensor PIR

## 5. Sensor Tekanan

Mengukur tekanan dalam berbagai aplikasi seperti tekanan udara atau tekanan air. Contohnya adalah barometer dan manometer.

Sensor Tekanan Arduino, Ada banyak jenis sensor tekanan, seperti sensor MPX5700, atau bisa juga di gunakan load cell, atau sensor strain gauge untuk mengukur berat, begitu juga dengan sensor GY-BMP28, namun semua sensor ini tidak cocok di gunakan untuk mengukur tekanan air. misalnya tekanan pada tank air. Jika menggunakan sensor sensor di atas mungkin membutuhkan konstruksi dan rangkaian sebagai penguat sinyalnya.



Gambar 2-14 Sensor Tekanan

## 6. Sensor Gas

Mendeteksi keberadaan dan konsentrasi gas tertentu di lingkungan. Contohnya adalah sensor CO2 dan sensor metana.

Sensor MQ-6 Gas LPG Sensor Module MQ 6 ditunjukan dalam gambar 2-



Gambar 2-15 Modul sensor gas metana

## 7. Sensor Sentuhan

Mendeteksi sentuhan fisik atau tekanan pada permukaan. Contohnya adalah layar sentuh kapasitif dan resistif.



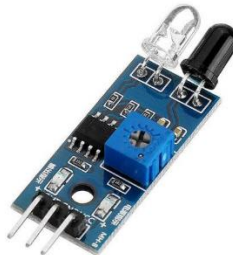
Gambar 2-16 Modul sensor sentuh

Sensor sentuh merupakan sensor elektronik yang bisa mendeteksi adanya sentuhan. Sensor tersebut akan beroperasi sebagai saklar ketika disentuh. Misalnya saja saklar pada lampu, layar sentuh pada smartphone dan lain sebagainya. Seiring perkembangan teknologi yang semakin pesat, sensor sentuh semakin banyak digunakan dan sudah menggeser peranan saklar mekanik pada perangkat elektronik.

## 8. Sensor Proximity

Mendeteksi keberadaan objek tanpa kontak fisik. Contohnya adalah sensor kapasitif dan induktif. IR OBSTACLE AVOIDANCE SENSOR MODULE adalah module yang terdiri dari 1 IR transmitter dan 1 IR receiver yang bekerja dengan prinsip pemantulan cahaya, sangat cocok untuk diaplikasikan ke project smart car / robotika untuk sensor deteksi objek / halangan / rintangan, robot line follower, dan project lainnya.

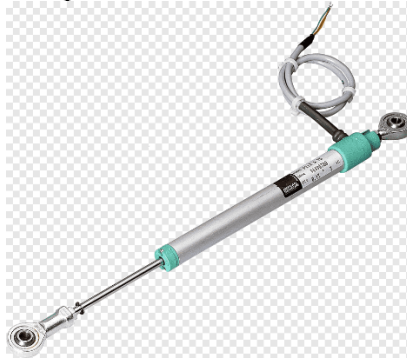
Berikut gambar sensor proximity



Gambar 2-17 Modul sensor proximity

## 9. Sensor Posisi

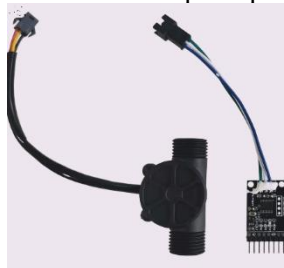
Mengukur posisi atau pergerakan linear atau rotasi. Contohnya adalah encoder dan potensiometer.



Gambar 2-18 Sensor posisi dengan resistensi

## 10. Sensor Aliran

Mengukur laju aliran cairan atau gas. Contohnya adalah flow meter dan anemometer. Modul Sensor aliran udara Sensor peniup interaktif perangkat induksi modul peniup untuk Arduino.



Gambar 2-19 Modul sensor aliran udara

## 11. Sensor PH Air

Sensor pH meter merupakan suatu sensor yang dapat melakukan pengukuran tingkat kadar keasaman atau kebasaan yang dimiliki oleh cairan/larutan. Cara bekerja dari sensor pH air yang utama berada di bagian sensor probe dengan material terbuat dari elektroda kaca, dimana pada elektroda kaca tersebut terdapat larutan HCL yang terdapat pada bagian ujung sensor probe, sensor probe tersebut akan mengukur besaran nilai ion  $H_3O^+$  pada suatu larutan sehingga dapat mengetahui kadar PH pada suatu larutan/cairan[8]. Elektroda sensor pada sensor PH air

terbentuk dari bahan lapisan kaca yang sensitif dengan impedansi yang kecil oleh sebab itu dapat mendapatkan hasil pembacaan dan penilaian yang stabil dan cepat pada suhu cairan/larutan tinggi maupun rendah. Hasil dari pembacaan nilai sensor PH bisa didapatkan oleh mikrokontroler dengan menggunakan antarmuka PH 2.0 yang sudah ada pada modul sensor PH air. Sensor PH air ini sangat baik untuk digunakan dalam melakukan pembacaan kadar PH cairan dengan interval waktu yang lama (Syafiqoh, Sunardi, & Yudhana, 2018). PH Meter Sensor Alat Ukur Asam Basa Air Cairan ditunjukkan dalam gambar 2-



Gambar 2-20 Modul Sensor PH air

## 12. Sensor PH Tanah

Sensor PH tanah ini digunakan untuk mengukur kadar PH tanah, seperti untuk pertanian dll. Sensor ini memiliki kualitas bagus, industrial grade. Bisa membaca PH tanah dengan baik. Interface menggunakan RS485 sehingga stabil dan minim interference. Sudah termasuk kabel 1m lebih.



Gambar 2-21 Sesor PH Tanah

Spesifikasi :

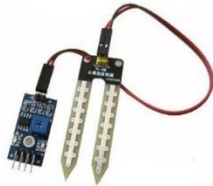
- Tegangan Kerja : 5-30v
- Range : 3-9 PH
- Interface : RS485

Bisa digunakan dengan berbagai perangkat seperti arduino,

raspberry pi, PLC dll sepanjang support RS485/modbus.

### 13. Sensor Kelembaban Tanah

Soil Moisture Sensor merupakan module untuk mendeteksi kelembaban tanah, yang dapat diakses menggunakan microcontroller seperti arduino. Sensor kelembaban tanah ini dapat dimanfaatkan pada sistem pertanian, perkebunan, maupun sistem hidroponik menggunakan hidrotan. Soil Moisture Sensor dapat digunakan untuk sistem penyiraman otomatis atau untuk memantau kelembaban tanah tanaman secara offline maupun online. Sensor yang dijual pasaran mempunyai 2 module dalam paket penjualannya, yaitu sensor untuk deteksi kelembaban, dan module elektroniknya sebagai amplifier sinyal.



Gambar 2-22 Modul sensor kelembaban tanah

#### Cara kerja sensor kelembaban tanah

Pada saat diberikan catudaya dan disensingkan pada tanah, maka nilai Output Analog akan berubah sesuai dengan kondisi kadar air dalam tanah.

Pada saat kondisi tanah :

Basah : tegangan output akan turun

Kering : tegangan output akan naik

Tegangan tersebut dapat dicek menggunakan voltmeter DC. Dengan pembacaan pada pin ADC pada microcontroller dengan tingkat ketelitian 10 bit, maka akan terbaca nilai dari range 0 – 1023. Sedangkan untuk Output Digital dapat dilihat pada nyala led Digital output menyala atau tidak dengan mensetting nilai ambang pada potensiometer.

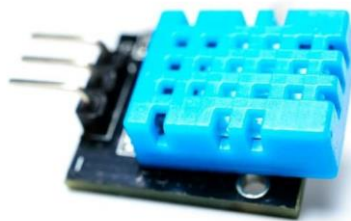
### 14. Humidity sensor

Humidity sensor adalah perangkat elektronik yang digunakan untuk mengukur jumlah uap air dalam udara atau kelembapan relatif. Kelembapan bisa diukur dalam dua bentuk, yaitu kelembapan absolut dan kelembapan relative. Sensor kelembapan memiliki aplikasi yang sangat luas. Sensor tersebut



digunakan dalam berbagai industri seperti pertanian, farmasi, penyimpanan makanan, industri otomotif, dan HVAC (Heating, Ventilation, and Air Conditioning) untuk memantau dan mengatur tingkat kelembapan yang tepat. Selain itu, sensor kelembapan juga digunakan dalam perangkat rumah tangga seperti pengering pakaian, pemanas ruangan, dan pendingin udara.

DHT11 adalah sensor suhu dan kelembapan udara yang mudah untuk ditambahkan pada project rangkaian elektronika berbasis microcontroller. Modul ini berkomunikasi dengan mikrokontroler menggunakan protokol 1-Wire. Sensor ini adalah salah satu sensor paling sering digunakan pada proyek elektronika berbasis Arduino. Membuat programnya pun sangat mudah karena library sudah tersedia. Gambar 2- menunjukkan sensor DHT 11,



Gambar 2-23 Modul Humidity sensor DHT11

## 15. Sensor suhu tanah

Sensor Suhu Tanah DS18B20 Sensor DS18B20 waterproof merupakan sensor temperatur digital yang dapat dihubungkan dengan mikrokontroler lewat antarmuka 1-Wire. Sensor DS18B20 memiliki keluaran digital sehingga tidak membutuhkan rangkaian ADC, serta akurasi nilai suhu dan kecepatan pengukuran memiliki kestabilan yang jauh lebih baik dari sensor suhu lainnya. Bentuk fisik dan deskripsi pin dari sensor DS18B20 waterproof ditunjukkan pada Gambar 2- dan Gambar 4. Sensor DS18B20 waterproof Keterangan antarmuka (interface) • Kabel Merah = VDD • Kabel Hitam = GND • Kabel kuning = DQ



Gambar 2-24 Sensor suhu tanah DS18B20

### 2.2.2 Aktuator

Aktuator merupakan perangkat yang mengubah sinyal kontrol (biasanya berupa sinyal listrik) menjadi tindakan fisik. Aktuator berfungsi sebagai "otot" dari sebuah sistem, memungkinkan perangkat atau sistem untuk melakukan aksi fisik berdasarkan perintah yang diberikan oleh pengontrol atau sensor.

Fungsi dan Jenis Aktuator

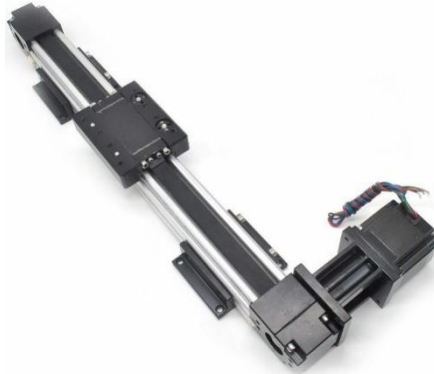
#### A. Fungsi Aktuator:

1. Konversi Energi: Aktuator mengubah energi listrik, hidrolik, atau pneumatik menjadi gerakan mekanis.
2. Eksekusi Tindakan: Aktuator mengeksekusi perintah dari sistem kontrol untuk melakukan aksi fisik seperti membuka katup, menggerakkan lengan robot, atau menghidupkan/mematikan perangkat.
3. Kontrol Proses: Aktuator berperan penting dalam mengontrol proses otomatisasi dengan melakukan tindakan berdasarkan data yang diterima dari sensor dan perintah dari pengontrol.

#### B. Jenis-jenis Aktuator:

##### 1. Aktuator Linear:

Menghasilkan gerakan linear (lurus). Contohnya adalah aktuator hidrolik dan pneumatik yang digunakan untuk mengangkat atau menggerakkan objek dalam garis lurus.



Gambar 2-25 Modul Aktuator linier

## 2. Aktuator Rotari:

Menghasilkan gerakan rotasi (putar). Contohnya adalah motor listrik dan motor servo yang digunakan dalam robotika dan mesin industri.



Gambar 2-26 Aktutor rotari

## 3. Aktuator Elektrik:

Mengubah energi listrik menjadi gerakan mekanis. Contohnya adalah motor listrik, solenoid, dan aktuator piezoelektrik.

## 4. Aktuator Hidrolik:

Menggunakan cairan bertekanan untuk menghasilkan gerakan. Contohnya adalah silinder hidrolik yang digunakan dalam alat berat seperti ekskavator.

#### 5. Aktuator Pneumatik:

Menggunakan udara bertekanan untuk menghasilkan gerakan. Contohnya adalah silinder pneumatik yang sering digunakan dalam sistem otomasi industri.

#### 6. Solenoid:

Mengubah energi listrik menjadi gerakan linear dengan menarik atau mendorong inti logam. Digunakan dalam katup solenoid dan kunci elektromagnetik.



Gambar 2-27 Modul solenoid

#### 7. Motor Servo:

Motor listrik dengan mekanisme umpan balik untuk kontrol posisi yang presisi. Digunakan dalam robotika dan kontrol gerakan.



Gambar 2-28 Motor servo

#### 8. Motor Stepper:

Motor listrik yang bergerak dalam langkah-langkah diskret, memungkinkan kontrol yang sangat presisi atas posisi. Digunakan dalam printer 3D dan alat CNC



Gambar 2-29 Modul stepper

Dengan kemampuannya untuk mengubah sinyal kontrol menjadi aksi fisik, aktuator memungkinkan sistem IoT untuk tidak hanya mengumpulkan dan menganalisis data, tetapi juga untuk melakukan tindakan yang dapat meningkatkan efisiensi, kenyamanan, dan keselamatan dalam berbagai aplikasi.

### 2.2.3 Perangkat Edge (*Edge Devices*):

Perangkat Edge (*Edge Devices*) merupakan komponen penting dalam arsitektur Internet of Things (IoT) yang bertanggung jawab untuk memproses data di dekat sumber data, daripada mengirimkannya langsung ke pusat data atau cloud. Perangkat Edge membantu mengurangi latensi, meningkatkan efisiensi, dan mengurangi beban jaringan dengan melakukan pemrosesan awal data dan hanya mengirimkan informasi penting ke cloud untuk analisis lebih lanjut.

Fungsi dan Keunggulan Perangkat Edge

Fungsi Edge

1. Fungsi Perangkat Edge: Pemrosesan Data Lokal: Perangkat Edge memproses data yang dikumpulkan oleh sensor secara lokal untuk mengambil keputusan cepat tanpa perlu mengirim semua data ke cloud.
2. Reduksi Latensi: Dengan memproses data di dekat sumbernya, perangkat Edge mengurangi waktu yang diperlukan untuk mengirim data ke cloud dan kembali, sehingga meningkatkan responsivitas sistem.
3. Penghematan Bandwidth: Hanya data yang relevan dan penting yang dikirim ke cloud, mengurangi penggunaan bandwidth dan biaya transmisi data.
4. Keamanan dan Privasi: Data sensitif dapat diproses dan disimpan secara lokal, mengurangi risiko pelanggaran data selama transmisi ke cloud.

5. Keandalan Sistem: Perangkat Edge dapat terus beroperasi dan membuat keputusan meskipun koneksi ke cloud terganggu.

#### Keunggulan Perangkat Edge:

1. Respons Cepat: Karena pemrosesan dilakukan secara lokal, sistem dapat merespons lebih cepat terhadap kejadian atau perubahan lingkungan.
2. Pengurangan Beban Cloud: Dengan memfilter dan menganalisis data secara lokal, beban pada infrastruktur cloud berkurang, sehingga lebih efisien.
3. Keberlanjutan Operasional: Sistem dapat terus berfungsi secara independen dari koneksi internet atau cloud, meningkatkan keandalan operasi.
4. Privasi Data: Mengurangi risiko privasi dengan memproses data sensitif secara lokal sebelum dikirim ke cloud.

#### Contoh dan Aplikasi Perangkat Edge

##### 1. Gateway IoT:

Fungsi: Menghubungkan berbagai perangkat IoT ke jaringan yang lebih luas dan melakukan pemrosesan awal data.

Contoh: Raspberry Pi yang digunakan sebagai gateway untuk mengumpulkan dan memproses data dari sensor sebelum mengirimnya ke cloud.

##### 2. Komputer Edge:

Fungsi: Memiliki kemampuan komputasi yang lebih kuat untuk analisis data dan pengambilan keputusan di lokasi.

Contoh: NVIDIA Jetson, yang digunakan untuk aplikasi AI dan machine learning di edge.

##### 3. Perangkat Industri Edge:

Fungsi: Digunakan dalam lingkungan industri untuk pemantauan dan pengendalian proses produksi secara real-time.

Contoh: PLC (*Programmable Logic Controller*) yang digunakan untuk mengontrol mesin manufaktur.

##### 4. Perangkat Kesehatan Edge:

Fungsi: Memproses data kesehatan pasien secara lokal untuk tindakan cepat dan pengawasan berkelanjutan.

Contoh: Alat pemantau jantung yang memproses data detak jantung secara lokal dan hanya mengirimkan alert jika terdeteksi kelainan.

## 5. Perangkat Pertanian Edge:

Fungsi: Memantau kondisi tanah, cuaca, dan tanaman untuk optimasi irigasi dan pengelolaan pertanian.

Contoh: Sensor kelembaban tanah yang mengirimkan data ke perangkat edge untuk analisis dan kontrol irigasi otomatis.

### Kesimpulan

Perangkat Edge merupakan komponen kritis dalam ekosistem IoT yang memungkinkan pemrosesan data yang cepat, efisien, dan aman di dekat sumber data. Dengan kemampuan untuk mengurangi latensi, menghemat bandwidth, dan meningkatkan privasi data, perangkat Edge memainkan peran penting dalam berbagai aplikasi mulai dari rumah pintar hingga industri, kesehatan, dan pertanian pintar.

## 2.2.4 Perangkat Komunikasi:

Perangkat komunikasi dalam *Internet of Things* (IoT) memainkan peran penting dalam menghubungkan berbagai sensor, aktuator, dan perangkat lain dalam jaringan, memungkinkan pertukaran data dan instruksi. Berikut adalah beberapa jenis perangkat komunikasi yang umum digunakan dalam ekosistem IoT:

### 1. Wi-Fi

Fungsi: Menghubungkan perangkat IoT ke internet melalui jaringan nirkabel lokal (WLAN).

Keunggulan: Kecepatan data tinggi, cakupan yang luas.

Contoh Penggunaan: Perangkat rumah pintar seperti kamera keamanan, termostat pintar, dan sistem hiburan rumah.

### 2. Bluetooth

Fungsi: Menghubungkan perangkat IoT untuk komunikasi jarak pendek.

Keunggulan: Konsumsi daya rendah, mudah dipasang.

Contoh Penggunaan: Perangkat wearable seperti jam tangan pintar, pelacak kebugaran, dan headset nirkabel.

### 3. Zigbee

Fungsi: Protokol komunikasi nirkabel untuk jaringan mesh yang memungkinkan banyak perangkat IoT untuk berkomunikasi satu sama lain.

Keunggulan: Konsumsi daya sangat rendah, ideal untuk perangkat dengan baterai kecil.

Contoh Penggunaan: Lampu pintar, sensor keamanan, dan sistem kontrol rumah pintar.

#### 4. Z-Wave

Fungsi: Protokol komunikasi nirkabel untuk jaringan mesh, mirip dengan Zigbee, dirancang untuk otomasi rumah.

Keunggulan: Konsumsi daya rendah, jaringan yang dapat diperluas.

Contoh Penggunaan: Kunci pintu pintar, termostat, dan sensor gerak.

#### 5. LoRaWAN (Long Range Wide Area Network)

Fungsi: Protokol komunikasi jarak jauh untuk perangkat IoT yang membutuhkan cakupan luas dengan konsumsi daya rendah.

Keunggulan: Cakupan jarak jauh, konsumsi daya sangat rendah.

Contoh Penggunaan: Sensor pertanian, pelacak aset, dan perangkat kota pintar.

#### 6. NB-IoT (Narrowband IoT)

Fungsi: Teknologi jaringan seluler untuk perangkat IoT dengan kebutuhan daya dan bandwidth yang rendah.

Keunggulan: Penetrasi sinyal yang baik, konsumsi daya rendah, cakupan luas.

Contoh Penggunaan: Meteran pintar, pemantauan lingkungan, dan pelacak kendaraan.

#### 7. Sigfox

Fungsi: Jaringan global untuk komunikasi IoT dengan konsumsi daya dan biaya yang sangat rendah.

Keunggulan: Konsumsi daya sangat rendah, biaya operasional rendah.

Contoh Penggunaan: Pemantauan infrastruktur, pelacakan logistik, dan sensor lingkungan.

#### 8. Cellular (GSM/3G/4G/5G)

Fungsi: Menghubungkan perangkat IoT ke internet melalui jaringan seluler.

Keunggulan: Cakupan global, kecepatan data tinggi (terutama dengan 4G dan 5G).



Contoh Penggunaan: Kendaraan terhubung, sistem pemantauan jarak jauh, dan perangkat IoT yang bergerak.

## 9. RFID (*Radio Frequency Identification*)

Fungsi: Menggunakan gelombang radio untuk mengidentifikasi dan melacak objek secara otomatis.

Keunggulan: Tidak memerlukan kontak langsung atau garis pandang.

Contoh Penggunaan: Manajemen inventaris, pelacakan logistik, dan akses kontrol.

## 10. NFC (*Near Field Communication*)

Fungsi: Protokol komunikasi jarak sangat pendek untuk pertukaran data antara perangkat yang berdekatan.

Keunggulan: Konsumsi daya sangat rendah, sangat aman untuk transaksi jarak pendek.

Contoh Penggunaan: Pembayaran nirkabel, kontrol akses, dan pertukaran data antara perangkat mobile.

## 11. EnOcean

Fungsi: Teknologi komunikasi nirkabel untuk perangkat IoT yang menggunakan energi dari lingkungan (seperti cahaya, gerakan) untuk operasional.

Keunggulan: Bebas baterai, konsumsi daya sangat rendah.

Contoh Penggunaan: Sensor lingkungan, sakelar nirkabel, dan kontrol bangunan pintar.

### Kesimpulan

Perangkat komunikasi dalam IoT sangat beragam dan dirancang untuk memenuhi berbagai kebutuhan spesifik dari aplikasi IoT. Pilihan teknologi komunikasi yang tepat tergantung pada faktor-faktor seperti jarak, konsumsi daya, kecepatan data, dan lingkungan operasi. Dengan memilih perangkat komunikasi yang sesuai, sistem IoT dapat dioptimalkan untuk kinerja yang lebih baik, efisiensi energi, dan keandalan.

### 2.2.5 Perangkat pemrosesan

Perangkat pemrosesan dalam Internet of Things (IoT) memainkan peran kunci dalam mengelola data yang dikumpulkan oleh sensor dan mengendalikan aktuator. Perangkat ini melakukan berbagai tingkat pemrosesan data, mulai dari pengambilan keputusan sederhana di perangkat tepi (edge) hingga analisis data kompleks di pusat data atau cloud. Berikut adalah beberapa jenis perangkat

pemrosesan yang umum dalam ekosistem IoT:

### 1. Mikrokontroler (Microcontroller Units - MCU)

**Fungsi:** Mikrokontroler adalah komputer kecil di dalam satu chip yang mengintegrasikan prosesor, memori, dan periferil input/output. Mereka digunakan untuk tugas pemrosesan sederhana dan kontrol langsung perangkat IoT.

**Keunggulan:** Konsumsi daya rendah, biaya rendah, ukuran kecil.

**Contoh:** Arduino, ESP8266, ESP32.

**Aplikasi:** Otomasi rumah, sensor lingkungan, perangkat wearable.

#### a) Arduino uno

Arduino Uno merupakan salah satu papan mikrokontroler berbasis dataseheet ATmega328 yang memiliki 14 pin input.



Gambar 2-30 Modul arduino uno

Arduino Uno merupakan salah satu papan kontroler mikro (*mikrokontroler*) berbasis dataseheet Atmega328. Papan kontroler ini bersifat sumber terbuka yang paling populer karena dirancang untuk memudahkan pengendalian elektronik di segala bidang. Arduino AG sebagai perusahaan Italia yang memegang merek Arduino melakukan kegiatan bisnisnya dengan memproduksi beberapa papan kontroler lainnya.

#### Kelebihan dan Kekurangan Arduino Uno

Penggunaan Arduino serie pertama sangat fleksibel dan mudah bagi programer pemula yang hendak mengaplikasikan papan kontroler tersebut. Karena rancangan papan kontroler ini sangat *simple* dan menggunakan bahasa pemograman yang mudah yakni bahasa program C. Prosesor yang ditanamkan

pada papan ini yakni Atmel AVR, serta menggunakan software IDE, Text-editor, Compiler, Serial Monitor, dan Serial ISP Programmer.

Papan pengendali ini memiliki 14 pin I/O digital dan terdapat juga 6 in analog. Untuk menghubungkan dapat menggunakan koneksi USB type A dan B seperti layaknya USB pada printer.

Disamping sejumlah kelebihan yang dimiliki, papan pengendali serie pertama ini memiliki sejumlah kekurangan. Diantaranya dari segi kapasitas memory penyimpanan yang sangat kecil sekitar 0,002 MB dan keterbatasan memory flash yakni 32 KB. Selain itu, kecepatan prosesor yang rendah dan tidak ada koneksi wifi, bluetooth, dan ethernet.

#### b) ESP8266

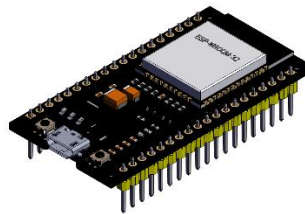
Modul WiFi ESP8266 merupakan sebuah komponen chip terintegrasi yang didesain untuk keperluan dunia masa kini yang serba tersambung. Chip ini menawarkan solusi networking Wi-Fi yang lengkap dan menyatu, yang dapat digunakan sebagai penyedia aplikasi atau untuk memisahkan semua fungsi networking Wi-Fi ke pemroses aplikasi lainnya. ESP8266 memiliki kemampuan on-board prosesi dan storage yang memungkinkan chip tersebut untuk diintegrasikan dengan sensor-sensor atau dengan aplikasi alat tertentu melalui pin input output hanya dengan pemrograman singkat. Modul komunikasi WiFi dengan IC SoC (System on Chip) ESP8266EX Serial-to-WiFi Communication Module ini merupakan modul WiFi dengan harga ekonomis. Kini Anda dapat menyambungkan rangkaian elektronika Anda ke internet secara nirkabel karena modul elektronika ini menyediakan akses ke jaringan WiFi secara transparan dengan mudah melalui interkoneksi serial (UART RX/TX).

Keunggulan utama modul ini adalah tersedianya mikrokontroler RISC(Tensilica 106 $\mu$  Diamond Standard Core LX3) dan Flash Memory SPI 4Mbit Winbond W2540BVNIG terpadu, dengan demikian dapat langsung menginjeksi kode program aplikasi langsung ke modul ini. 1. Fitur System on Chip (SoC) ESP8266EX: • Mendukung protokol 802.11 b/g/n • WiFi Direct (P2P / Point-to-Point), Soft-AP / Access Point • TCP/IP Protocol Stack terpadu • Mendukung WEP, TKIP, AES, dan WAPI • Pengalih T/R, balun, LNA (penguat derau

rendah) terpadu • Power Amplifier / penguat daya 24 dBm terpadu • Sirkuit PLL, pengatur tegangan, dan pengelola daya terpadu • Daya keluaran mencapai +19,5 dBm pada moda 802.11b • Sensor suhu internal terpadu • Mendukung berbagai macam antena • Kebocoran arus pada saat non-aktif kurang dari 10 $\mu$ A • CPU mikro 32-bit terpadu yang dapat digunakan sebagai pemroses aplikasi lewat antarmuka iBus, dBus, AHB (untuk akses register), dan JTAG (untuk debugging) • Antarmuka SDIO 2.0, SPI, UART • STBC, 1x1 MIMO, 2x1 MIMO • Agregasi A-MPDU dan A-MSDU dengan guard interval 0,4  $\mu$ s • Penulisan dan pengiriman data kurang dari 2 ms.

### c) ESP32

ESP32 adalah mikrokontroler yang dikenalkan oleh Espressif System merupakan penerus dari mikrokontroler ESP8266. Pada mikrokontroler ini sudah tersedia modul WiFi dalam chip sehingga sangat mendukung untuk membuat sistem aplikasi Internet of Things. ESP32 sendiri tidak jauh berbeda dengan ESP8266 yang familiar di pasaran, hanya saja ESP32 lebih kompleks dibandingkan ESP8266, cocok untuk sobat dengan proyek yang besar.



Gambar 2-31 Modul ESP32

Berikut ini merupakan spesifikasi yang dimiliki oleh mikrokontroler ESP32 (kece, 2021) :

Atribut	Detail
CPU	Tensilica Xtensa LX6 32bit Dual-Core di 160/240MHz
SRAM	520 KB
FLASH	2MB (max. 64MB)
Tegangan	2.2V sampai 3.6V
Arus Kerja	Rata-rata 80mA
Dapat diprogram	Ya (C, C++, Python, Lua, dll)
Open Source	Ya
<b>Konektivitas</b>	
Wi-Fi	802.11 b/g/n
Bluetooth	4.2BR/EDR + BLE
UART	3
<b>I/O</b>	
GPIO	32
SPI	4
I2C	2
PWM	8
ADC	18 (12-bit)
DAC	2 (8-bit)

Prosesor: Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz.

Memori: 520 KB SRAM.

Wireless connectivity: Wi-Fi 802.11 b/g/n, Bluetooth v4.2 BR/EDR and BLE (shares the radio with Wi-Fi).

Peripheral I/O: 12-bit SAR ADC (up to 18 channels), 2x 8-bit DACs, 10x touch sensors (capacitive sensing GPIOs), 4x SPI, 2x I2S interfaces, 2x I2C interfaces, 3x UART, SD/SDIO/CE-ATA/MMC/eMMC host controller, SDIO/SPI slave controller, Ethernet MAC interface, CAN bus 2.0, infrared remote controller (TX/RX, up to 8 channels), motor PWM, LED PWM (up to 16 channels), hall effect sensor, ultra low power analog pre-amplifier.

Security : IEEE 802.11 standard security, secure boot, flash, encryption, 1024-bit, OTP (up to 768-bit for customers), cryptographic hardware acceleration (AES, SHA-2, RSA, ECC), random number generator (RNG).

Software pemrograman yang bisa digunakan untuk memprogram Mikrokontroler ESP32

Mikrokontroler ESP32 ini dapat diprogram dengan menggunakan C++, C, Python, Lua, dll. untuk menjalankan program mikrokontroler ESP32 ini memerlukan suatu software pemrograman, berikut ini adalah contoh softwrenya untuk

menjalankan program mikrokontroler ESP32, diantaranya sebagai berikut :

- Arduino Promini.
- Arduino IDE.
- Ubuntu 14.04 LTS.
- ESP-IDF Visual Studio Code Extension.
- Espressif IoT Development Framework.

## 2. Mikroprosesor (Microprocessor Units - MPU)

Fungsi: Mikroprosesor lebih kuat dibandingkan mikrokontroler dan digunakan untuk tugas pemrosesan yang lebih kompleks yang membutuhkan kekuatan komputasi lebih besar.

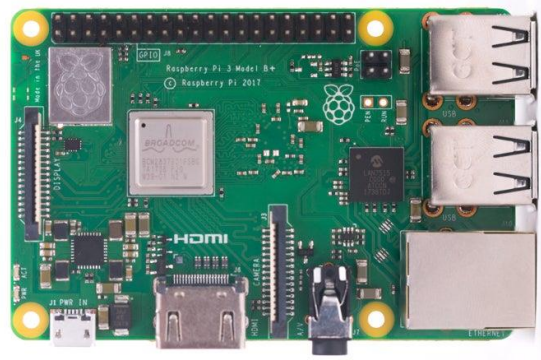
Keunggulan: Kinerja tinggi, mampu menjalankan sistem operasi yang kompleks.

Contoh: Raspberry Pi, BeagleBone.

Aplikasi: Gateway IoT, perangkat edge computing, aplikasi AI/ML.

### a) Raspberry Pi

Raspberry Pi adalah nama serangkaian komputer papan tunggal yang dibuat oleh Raspberry Pi Foundation , sebuah badan amal Inggris yang bertujuan untuk mengedukasi masyarakat dalam bidang komputasi dan menciptakan akses yang lebih mudah terhadap pendidikan komputasi.



Gambar 2-32 Modul Raspberry pi

Raspberry Pi diluncurkan pada tahun 2012, dan telah ada beberapa iterasi dan variasi yang dirilis sejak saat itu. Pi asli memiliki CPU single-core 700MHz dan hanya 256MB RAM,

dan model terbaru memiliki CPU quad-core yang memiliki clock lebih dari 1,5GHz, dan RAM 4GB.

Raspberry Pi adalah komputer sangat murah yang menjalankan Linux, tetapi juga menyediakan satu set pin GPIO (input/output tujuan umum), memungkinkan Anda mengontrol komponen elektronik untuk komputasi fisik dan menjelajahi Internet of Things (IoT).

#### b) BeagleBone

Beaglebone Black (BBB) sama seperti Komputer, yang hadir dalam paket ringkas dengan prosesor, akselerasi grafis, memori, dan semua IC yang diperlukan yang disolder untuk membentuk satu papan sirkuit. Oleh karena itu, ini juga disebut sebagai Komputer Papan Tunggal. Ia menggunakan prosesor kuat yang disebut prosesor ARM Cortex-A8 dengan 1GHz AM335x. Mikrokontroler Beaglebone Black diilustrasikan pada gambar di bawah ini.

Papan Mikrokontroler Beaglebone Black ini akan menyediakan semua koneksi yang diperlukan untuk layar, Jaringan Ethernet, Mouse, dan Keyboard. Booting prosesor ini dilakukan dengan menggunakan OS Linux.



Gambar 2-33 Modul Beaglebone Black

Alat ini terutama digunakan oleh para peneliti untuk merancang proyek yang kompleks dan mempelajari Sistem Operasi Linux secara efektif. Beaglebone Black ini mirip dengan Beaglebone dengan fitur tambahan. Kecepatan pengoperasiannya lebih cepat dan banyak digunakan jika

dibandingkan dengan Beaglebone. Ini digunakan dalam robotika, proyek IoT, dan otomatisasi di tingkat pengembangan.

Ikuti langkah-langkah untuk memulai Beaglebone Black dengan sangat mudah dan cepat

Pertama, AKTIFKAN Beaglebone Black ke PC/komputer Anda dengan bantuan kabel mini USB. Kemudian boot ke OS Linux. Itu distro Linux, Angstrom.

Sekarang sambungkan semua unit periferal seperti USB, dan layar.

Untuk menghubungkan Beaglebone Black ke browser web dan mengontrol dengan komputer, driver diinstal.

Kini Beaglebone Black siap dengan OS Linux, di mana pengguna dapat menulis dan menjalankan program perangkat lunak menggunakan fungsi perpustakaan dan python tanpa batasan apa pun. Ini membantu untuk mengelola dan mengontrol semua pin GPIO prosesor.

#### Konfigurasi Pin/Diagram Pin

Ada 8 mode berbeda- mode 0, mode 1, mode 2, mode 3, mode 4, mode 5, mode 6, mode 7 dalam beaglebone hitam untuk setiap pin I/O digital, termasuk pin GPIO. Ini berisi 2 header ekspansi P9 dan P8 dengan masing-masing 46 pin dan dapat memberikan sinyal I/O dengan 3,3 Volt.

Jika tegangan diberikan pada pin sebesar 5 Volt, maka seluruh papan akan rusak. Konfigurasi pin/diagram pin mikrokontroler hitam beaglebone diilustrasikan pada gambar di bawah ini. Konfigurasi pin mikrokontroler hitam beaglebone dari dua header ekstensi P8 dan P9 diberikan dalam bentuk tabel di bawah. PIN pada prosesor diwakili oleh nomor PROC.

Untuk menghasilkan sinyal untuk mengendalikan motor tanpa menggunakan siklus CPU tambahan, hingga 8 pin I/O dikonfigurasi dengan modulator lebar pulsa (PWM)

Pada header ekstensi P9 ini, pin no 32 hingga pin no 40 berisi satu ADC 12-bit (konverter analog ke digital) dengan 8 saluran

Ini memiliki 2 port I2C. Salah satu port digunakan untuk membaca EEPROM dan menjalankan fungsi I/O digital tanpa gangguan apa pun pada pengoperasian tersebut. Port I2C lainnya digunakan untuk mengkonfigurasi kebutuhan pengguna.

Untuk perpindahan data dengan cepat, terdapat dua port



## SPI

Spesifikasi Mikrokontroler Beaglebone Black

Spesifikasi Mikrokontroler Beaglebone **Black** adalah sebagai berikut

- Jenis prosesor – Sitara AM3358BZCZ100 dengan 1 GHz dan 2000 MIPS
- Mesin Grafis- 20M Poligon/S, SGX530 3D
- Ukuran memori SDRAM – 512 MB DDR3L, 800 MHz
- Flash Onboard- MMC Tertanam 8-bit dengan 4 GB
- PMIC – 1 LDO tambahan, regulator PMIC TPS65217C
- Dukungan Debug – Serial Header, CTI 20-pin opsional terpasang
- Sumber Daya – mini USB, USB atau jack DC; DC eksternal 5 Volt melalui header ekspansi
- PCB – 3,4" X 2,1"; 6 lapisan
- Jenis indikator – 1 daya, 2 Ethernet, 4 LED, yang dapat dikontrol pengguna
- Port Host USB 2.0 HS – Dapat diakses ke USB1, Soket Tipe A, 500 mA LS/FS/HS
- Port Serial – Akses UART0 melalui header TTL 6-pin 3,3 Volt. Header terisi
- Ethernet – 10/100, RJ45
- Input Pengguna – Tombol daya, tombol Reset, tombol Boot
- Konektor SD/MMC – microSD, 3,3 Volt
- Keluaran video – HDMI 16b, 1280×1024 (maks), 1025×768, 1280×720, 1440×900, dengan dukungan EDID
- Port Klien HS USB 2.0 – Akses ke USB0, mode klien melalui miniUSB
- Audio – Stereo, melalui antarmuka HDMI
- Berat – 39,68 gram (1,4 ons)
- Konektor Ekspansi – 5 Volt, daya 3,3 Volt, VDD\_ADC 1,8 Volt.
- 3,3 Volt pada semua sinyal I/O – GPIO (maks 69), McASP0, I2C, SPI1, LCD, GPMC, MMC1, MMC2, 4 port serial, 4 timer, 7 AIN (maks 1,8 Volt), CAN0, interupsi XDMA, EHRPWM (0, 2), Tombol daya, ID papan ekspansi (menumpuk hingga 4)

### 3. Gateway IoT

Fungsi: Gateway IoT bertindak sebagai jembatan antara perangkat IoT dan cloud, melakukan pemrosesan data awal, protokol translasi, dan mengelola konektivitas jaringan.

Keunggulan: Mengurangi latensi, menghemat bandwidth, meningkatkan keamanan data.

Contoh: Intel NUC, Cisco IoT Gateway.

Aplikasi: Pemantauan industri, rumah pintar, kota pintar.

### 4. Komputer Edge (Edge Computers)

Fungsi: Komputer edge melakukan pemrosesan data di atau dekat sumber data untuk mengurangi latensi dan meningkatkan responsivitas.

Keunggulan: Pengurangan latensi, pemrosesan data lokal, peningkatan privasi.

Contoh: NVIDIA Jetson, Google Coral.

Aplikasi: Pemrosesan video di kamera keamanan, analisis data sensor di lokasi industri, sistem AI/ML di edge.

### 5. Komputer Single-Board (Single Board Computers - SBC)

Fungsi: SBC adalah komputer lengkap di satu papan sirkuit dengan prosesor, memori, penyimpanan, dan input/output.

Keunggulan: Fleksibilitas, biaya rendah, ukuran kecil.

Contoh: Raspberry Pi, Arduino.

Aplikasi: Prototyping, pendidikan, aplikasi IoT skala kecil.

### 6. Perangkat Keras AI/ML (AI/ML Hardware)

Fungsi: Perangkat keras AI/ML dirancang khusus untuk mempercepat tugas-tugas pembelajaran mesin dan kecerdasan buatan.

Keunggulan: Kinerja tinggi untuk tugas AI/ML, optimasi untuk inference dan training model.

Contoh: Google Edge TPU, NVIDIA Jetson Xavier.

Aplikasi: Pengenalan gambar, pemrosesan suara, analisis data real-time.

### 7. Field Programmable Gate Array (FPGA)

Fungsi: FPGA adalah perangkat yang dapat diprogram ulang untuk tugas-tugas tertentu, memungkinkan pemrosesan paralel yang sangat cepat dan efisien.

Keunggulan: Kustomisasi tinggi, kinerja sangat cepat untuk

tugas spesifik.

Contoh: Xilinx FPGA, Intel FPGA.

Aplikasi: Komunikasi jaringan, pemrosesan sinyal digital, aplikasi real-time.

## 8. System on Chip (SoC)

Fungsi: SoC mengintegrasikan semua komponen komputer (prosesor, memori, input/output) ke dalam satu chip untuk meningkatkan efisiensi dan mengurangi ukuran.

Keunggulan: Konsumsi daya rendah, ukuran kompak, integrasi tinggi.

Contoh: Qualcomm Snapdragon, Apple A-series.

Aplikasi: Smartphone, perangkat wearable, perangkat IoT dengan kebutuhan komputasi tinggi.

Kesimpulan

Perangkat pemrosesan dalam IoT sangat bervariasi dalam hal kemampuan, ukuran, konsumsi daya, dan biaya. Pemilihan perangkat pemrosesan yang tepat tergantung pada kebutuhan spesifik dari aplikasi IoT, termasuk kompleksitas pemrosesan data, kebutuhan latensi, dan lingkungan operasional. Dengan memanfaatkan perangkat pemrosesan yang tepat, sistem IoT dapat dioptimalkan untuk efisiensi, kecepatan, dan kinerja yang lebih baik.

### 2.2.6 Perangkat Penyimpanan Data

- Server Lokal: Penyimpanan data di lokasi, sering digunakan untuk aplikasi dengan kebutuhan latensi rendah.
- Penyimpanan Cloud: Menyimpan data yang dikumpulkan dari perangkat IoT secara aman dan memungkinkan akses dari mana saja.

### 2.2.7 Perangkat Antarmuka Pengguna (User Interface Devices):

- Smartphone/Tablet: Digunakan untuk memantau dan mengontrol perangkat IoT melalui aplikasi.
- Dashboard Web: Antarmuka berbasis web untuk mengakses dan menganalisis data IoT.

### 2.2.8 Perangkat Keamanan:

- Firewall IoT: Melindungi jaringan IoT dari akses yang tidak sah.
- Sistem Deteksi Intrusi: Memonitor dan mendeteksi aktivitas mencurigakan dalam jaringan IoT.
- Enkripsi Data: Perangkat dan software yang mengenkripsi data

untuk melindungi dari intersepsi selama transmisi.

#### 2.2.9 Perangkat Pendukung Khusus:

- Kamera Keamanan: Digunakan untuk pemantauan visual dalam sistem keamanan.
- Pelacak GPS: Melacak lokasi kendaraan atau aset.
- Alat Pemantau Kesehatan: Seperti monitor detak jantung atau perangkat pemantau diabetes.

Keseluruhan sistem IoT terdiri dari kombinasi berbagai jenis perangkat ini, yang bekerja bersama untuk mengumpulkan data, menganalisis informasi, dan mengambil tindakan berdasarkan data yang diperoleh.

### 2.3 Rangkuman

Sensor dan aktuator adalah komponen kunci dalam sistem otomatisasi dan kendali modern. Sensor mengumpulkan data dari lingkungan fisik dan mengirimkan informasi ini ke pengendali atau prosesor. Berdasarkan data ini, pengendali mengirim sinyal ke aktuator untuk melakukan aksi yang diperlukan, seperti membuka pintu, memutar motor, atau mengatur suhu.

Kombinasi sensor dan aktuator memungkinkan pembuatan sistem yang cerdas dan responsif, yang dapat meningkatkan efisiensi, keamanan, dan kenyamanan dalam berbagai aplikasi, mulai dari rumah pintar hingga pabrik otomatis dan kendaraan otonom. Memahami cara kerja dan aplikasi sensor serta aktuator adalah dasar penting dalam pengembangan dan implementasi teknologi *Internet of Things* (IoT) dan sistem kendali otomatis lainnya.

### 2.4 Latihan Soal Pengenalan Sensor

Berikut adalah beberapa latihan soal yang dirancang untuk memperdalam pemahaman dasar Anda tentang sensor. Soal-soal ini mencakup konsep dasar sensor, jenis-jenis sensor, prinsip kerja, dan aplikasi praktis dalam penggunaan sensor.

#### 1. Konsep Dasar Sensor

Soal 1: Apa yang dimaksud dengan sensor? Jelaskan peran sensor dalam sistem pengukuran dan kontrol.

Jawaban:

Sensor adalah perangkat yang mendeteksi perubahan lingkungan

fisik atau kimia dan mengubahnya menjadi sinyal yang dapat dibaca oleh instrumen atau pengguna. Peran sensor dalam sistem pengukuran dan kontrol adalah untuk menyediakan data yang diperlukan untuk memantau kondisi lingkungan atau proses, serta mengirimkan informasi tersebut ke sistem pengendali untuk tindakan lebih lanjut.

## 2. Jenis-Jenis Sensor

Soal 2: Sebutkan lima jenis sensor dan parameter fisik apa yang diukur oleh masing-masing sensor tersebut.

Jawaban:

1. Sensor Suhu: Mengukur suhu (misalnya, DHT11, DHT22, LM35)
2. Sensor Kelembaban: Mengukur kelembaban udara (misalnya, DHT11, DHT22)
3. Sensor Cahaya: Mengukur intensitas cahaya (misalnya, LDR - Light Dependent Resistor, fotodiode)
4. Sensor Tekanan: Mengukur tekanan udara atau cairan (misalnya, BMP180, BMP280)
5. Sensor Gas: Mengukur konsentrasi gas tertentu di udara (misalnya, MQ-2 untuk metana, LPG, dan asap)

## 3. Prinsip Kerja Sensor

Soal 3: Jelaskan prinsip kerja sensor suhu tipe termistor.

Jawaban:

Termistor adalah jenis sensor suhu yang resistansinya berubah dengan perubahan suhu. Terdapat dua jenis termistor: NTC (Negative Temperature Coefficient) dan PTC (Positive Temperature Coefficient). Pada NTC, resistansi berkurang saat suhu meningkat, sedangkan pada PTC, resistansi meningkat saat suhu meningkat. Perubahan resistansi ini dapat diukur dan dikonversi menjadi nilai suhu.

## 4. Aplikasi Sensor

Soal 4: Berikan dua contoh aplikasi di mana sensor cahaya digunakan dan jelaskan fungsinya dalam aplikasi tersebut.

Jawaban:

1. Lampu Jalan Otomatis: Sensor cahaya digunakan untuk mendeteksi intensitas cahaya lingkungan. Ketika cahaya di sekitar

menurun (misalnya, saat malam tiba), sensor mengaktifkan lampu jalan otomatis.

2. Ponsel Pintar: Sensor cahaya digunakan untuk menyesuaikan kecerahan layar secara otomatis berdasarkan intensitas cahaya di sekitar pengguna, sehingga meningkatkan kenyamanan penggunaan dan menghemat daya baterai.

## 5. Pemecahan Masalah Sensor

Soal 6: Jika sensor tekanan BMP180 tidak memberikan bacaan yang akurat pada mikrokontroler Anda, sebutkan tiga langkah pemecahan masalah yang bisa Anda lakukan.

Jawaban:

1. Periksa Koneksi Kabel: Pastikan semua kabel terhubung dengan benar dan kuat antara sensor dan mikrokontroler. Periksa pin VCC, GND, SCL, dan SDA.

2. Verifikasi Kode Program: Pastikan kode program Anda menggunakan alamat I2C yang benar dan menginisialisasi sensor dengan benar. Gunakan library yang sesuai dan pastikan tidak ada kesalahan dalam kode.

3. Cek Sensor dan Mikrokontroler: Coba sensor pada mikrokontroler lain atau uji mikrokontroler dengan sensor lain untuk memastikan tidak ada kerusakan perangkat keras.

Dengan latihan soal ini, Anda dapat memperdalam pemahaman tentang sensor, prinsip kerjanya, serta bagaimana menghubungkan dan memprogram sensor dalam aplikasi nyata.

## 2.5 Bahan Diskusi

Diskusi Lanjutan dan Pertanyaan

1. Bagaimana sensor dan aktuator dapat berkontribusi dalam pengembangan teknologi masa depan seperti smart farming dan kendaraan otonom?
2. Apa saja perkembangan terbaru dalam teknologi sensor dan aktuator?
3. Bagaimana cara mengintegrasikan beberapa sensor dan aktuator dalam satu sistem untuk aplikasi kompleks?

## 2.6 Rujukan

- Harmini, H., & Nurhayati, T. (2020). Desain Solar Power Inverter pada Sistem Photovoltaic. *Elektrika*, 1. kece, i. (2021, Juli 31). <https://iotkece.com>. Retrieved from apa-itu-esp32-spesifikasi-esp32: [https://iotkece.com/apa-itu-esp32-spesifikasi-esp32/google\\_vignette](https://iotkece.com/apa-itu-esp32-spesifikasi-esp32/google_vignette)
- Sulistiyanti, S., & Purwiyanti, ,. P. (2020). *Sensor dan Prinsip Kerjanya*. Lampung: Pustaka Media.
- Syafiqoh, U., Sunardi, S., & Yudhana, A. (2018). Pengembangan Wireless Sensor Network Berbasis Internet of Things untuk Sistem Pemantauan Kualitas Air dan Tanah Pertanian. *Jurnal Informasi dan Pengembangan IT*, vol. 3, no. 2, 285-289.
- Yusro, M., & Diamah, A. (2019). *Sensor dan Transduser*. Jakarta: Fakultas Teknik Universitas Negeri Jakarta.

## **BAB 3. PEMROGRAMAN SENSOR DAN AKTUATOR**

### **Capaian Pembelajaran Matakuliah (CPMK)**

Mampu memprogram arduino, esp32 untuk membaca data sensor dan mengendalikan aktuator

### **3.1 Pendahuluan**

#### **3.1.1 Latar belakang**

Dalam era teknologi yang berkembang pesat, sensor dan aktuator menjadi komponen kunci dalam sistem otomasi dan Internet of Things (IoT). Sensor adalah perangkat yang mendeteksi perubahan fisik atau kimia di lingkungan dan mengubahnya menjadi sinyal yang dapat diukur. Sebaliknya, aktuator adalah perangkat yang mengubah sinyal listrik menjadi gerakan fisik atau aksi mekanis. Pemrograman sensor dan aktuator memungkinkan sistem untuk berinteraksi dengan dunia nyata, menciptakan aplikasi yang dapat merespons kondisi lingkungan dan melakukan tindakan otomatis.

#### **3.1.2 Tujuan**

Tujuan dari pemrograman sensor dan aktuator adalah untuk mengembangkan sistem yang dapat:

1. Mengumpulkan data dari lingkungan fisik menggunakan sensor.
2. Memproses data yang dikumpulkan untuk mengambil keputusan.
3. Mengendalikan aktuator untuk melakukan aksi yang diperlukan berdasarkan keputusan tersebut.

#### **3.1.3 Langkah-langkah Pemrograman**

##### **3.1.3.1 Memahami Spesifikasi Sensor dan Aktuator:**

- i. Kenali parameter yang diukur oleh sensor dan cara kerja aktuator.
- ii. Pahami kebutuhan daya dan pin koneksi.



### 3.1.3.2 Koneksi Perangkat:

- i. Hubungkan sensor dan aktuator ke mikrokontroler (seperti Arduino atau ESP32).
- ii. Pastikan koneksi yang benar untuk pin daya, ground, dan sinyal.

### 3.1.3.3 Menggunakan Library yang Sesuai:

- i. Gunakan library yang tersedia untuk memudahkan pemrograman sensor dan aktuator.
- ii. Contoh library: `DHT` untuk sensor DHT11/DHT22, `Servo` untuk mengontrol servo motor.

### 3.1.3.4 Mengembangkan Kode Program:

- i. Inisialisasi sensor dan aktuator dalam fungsi `setup()`.
- ii. Baca data dari sensor dalam fungsi `loop()`.
- iii. Proses data yang dibaca dan kontrol aktuator berdasarkan data tersebut.

## 3.2 Pemrograman sensor suhu udara pada Arduino

Pemrograman sensor suhu pada Arduino adalah langkah dasar yang sangat berguna untuk banyak proyek berbasis mikrokontroler. Berikut adalah panduan langkah demi langkah untuk memprogram sensor suhu menggunakan Arduino dan sensor suhu populer seperti DHT11 atau DHT22.

### 3.2.1 Bahan yang Dibutuhkan

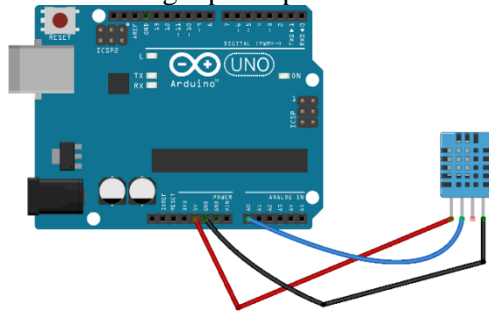
1. Arduino Board (misalnya, Arduino Uno)
2. Sensor Suhu (DHT11 atau DHT22)
3. Breadboard dan Kabel Jumper
4. Resistor (jika diperlukan, biasanya 10k ohm untuk pull-up)

### 3.2.2 Langkah-langkah Pemrograman

#### 3.2.2.1 Koneksi Hardware

- DHT11/DHT22 Pinout:
- VCC: Power supply (3.3V atau 5V)
- GND: Ground
- DATA: Data pin (dianggap terhubung ke pin digital Arduino)
- NC: Not Connected (DHT11 saja)
- Koneksi DHT11/DHT22 ke Arduino:

- VCC: Sambungkan ke 5V pada Arduino.
- GND: Sambungkan ke GND pada Arduino.
- DATA: Sambungkan ke pin digital 4 (atau pin lain yang diinginkan) pada Arduino.
- Jika diperlukan, sambungkan resistor 10k ohm antara pin VCC dan DATA sebagai pull-up resistor.



Gambar 3-1 Rangkaian sensor DHT11 pada Arduino

### 3.2.2.2 Instalasi Library DHT

Untuk memudahkan pemrograman, gunakan library DHT dari Adafruit.

- Buka Arduino IDE.
- Pergi ke Sketch > Include Library > Manage Libraries.
- Cari "DHT sensor library" oleh Adafruit dan instal.
- Juga instal "Adafruit Unified Sensor" library.

### 3.2.2.3 3. Kode Program

Berikut adalah contoh kode untuk membaca suhu dan kelembaban dari sensor DHT11 atau DHT22 dan menampilkan hasilnya di Serial Monitor.

```
#include <DHT.h>

// Pin yang terhubung ke sensor
#define DHTPIN 4

// Pilih jenis sensor (DHT11 atau DHT22)
#define DHTTYPE DHT11 // atau DHT22 jika menggunakan
sensor DHT22

// Inisialisasi sensor DHT
```

```

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  // Memulai komunikasi serial
  Serial.begin(9600);
  Serial.println("DHTxx test!");

  // Memulai sensor DHT
  dht.begin();
}

void loop() {
  // Menunggu antara pembacaan
  delay(2000);

  // Membaca suhu dan kelembaban
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float f = dht.readTemperature(true); // Menggunakan true untuk
Fahrenheit

  // Memeriksa jika pembacaan gagal dan keluar dari loop
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  // Menampilkan data ke Serial Monitor
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.print(" *C ");
  Serial.print(f);
  Serial.println(" *F");
}

```

#### 3.2.2.4 Menjalankan Program

- a) Sambungkan Arduino ke komputer menggunakan kabel

USB.

- b) Buka Arduino IDE dan paste kode di atas ke dalam editor.
- c) Pilih jenis board yang sesuai dan port yang terhubung ke Arduino melalui Tools > Board dan Tools > Port.
- d) Unggah kode ke board dengan menekan tombol Upload.
- e) Buka Serial Monitor dengan menekan tombol Serial Monitor di pojok kanan atas atau melalui Tools > Serial Monitor.
- f) Pastikan baud rate diatur ke 9600 baud.
- g) Anda akan melihat output suhu dan kelembaban dari sensor di Serial Monitor.

### 3.2.2.5 Kesimpulan

Dengan mengikuti langkah-langkah di atas, Anda dapat dengan mudah membaca data suhu dan kelembaban dari sensor DHT11 atau DHT22 menggunakan Arduino. Ini adalah dasar yang baik untuk banyak proyek berbasis IoT dan otomatisasi rumah. Dengan menambahkan aktuator atau komponen lain, Anda dapat mengembangkan sistem yang lebih kompleks dan fungsional.

## 3.3 Pemrograman sensor kelembaban tanah pada arduino

Pemrograman sensor kelembaban tanah pada Arduino adalah dasar yang sangat bermanfaat untuk proyek-proyek otomatisasi pertanian atau taman pintar. Berikut adalah panduan langkah demi langkah untuk memprogram sensor kelembaban tanah menggunakan Arduino.

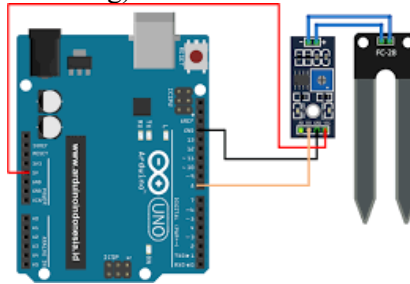
### 3.3.1 Bahan yang Dibutuhkan

- 1. Arduino Board (misalnya, Arduino Uno)
- 2. Sensor Kelembaban Tanah (Soil Moisture Sensor)
- 3. Breadboard dan Kabel Jumper
- 4. Resistor (jika diperlukan, biasanya 10k ohm untuk pull-up)
- 5. Modul ADC (Analog-to-Digital Converter) jika menggunakan board tanpa input analog

### 3.3.2 Langkah-langkah Pemrograman

#### 3.3.2.1 Koneksi Hardware

- Pinout Sensor Kelembaban Tanah:
- VCC: Power supply (3.3V atau 5V)
- GND: Ground
- A0: Analog output (mengirim nilai analog berdasarkan kelembaban tanah)
- D0: Digital output (mengirim sinyal digital berdasarkan ambang batas)
- Koneksi Sensor Kelembaban Tanah ke Arduino:
- VCC: Sambungkan ke 5V pada Arduino.
- GND: Sambungkan ke GND pada Arduino.
- A0: Sambungkan ke pin analog A0 pada Arduino (untuk pembacaan nilai analog).



Gambar 3-2 Rangkaian sensor kelembaban pada arduino

#### 3.3.2.2 Kode Program

Berikut adalah contoh kode untuk membaca nilai kelembaban tanah dari sensor dan menampilkan hasilnya di Serial Monitor.

```
// Pin yang terhubung ke sensor kelembaban tanah
#define SOIL_MOISTURE_PIN A0

void setup() {
  // Memulai komunikasi serial
  Serial.begin(9600);
}

void loop() {
  // Membaca nilai analog dari sensor
  int sensorValue = analogRead(SOIL_MOISTURE_PIN);
```

```

// Mengonversi nilai analog ke persentase kelembaban
// Asumsi: Nilai sensor 0 (sangat kering) hingga 1023 (sangat
basah)
float moisturePercentage = map(sensorValue, 0, 1023, 0, 100);

// Menampilkan data ke Serial Monitor
Serial.print("Soil Moisture Value: ");
Serial.print(sensorValue);
Serial.print(" - ");
Serial.print(moisturePercentage);
Serial.println(" %");

// Menunggu sebelum pembacaan berikutnya
delay(1000);
}

```

### 3.3.2.3 Menjalankan Program

- a) Sambungkan Arduino ke komputer menggunakan kabel USB.
- b) Buka Arduino IDE dan paste kode di atas ke dalam editor.
- c) Pilih jenis board yang sesuai dan port yang terhubung ke Arduino melalui Tools > Board dan Tools > Port.
- d) Unggah kode ke board dengan menekan tombol Upload.
- e) Buka Serial Monitor dengan menekan tombol Serial Monitor di pojok kanan atas atau melalui Tools > Serial Monitor.
- f) Pastikan baud rate diatur ke 9600 baud.
- g) Anda akan melihat output nilai kelembaban tanah dari sensor di Serial Monitor.

### 3.3.2.4 Penjelasan Kode

- a) Inisialisasi Pin:
  - `#define SOIL_MOISTURE_PIN A0`: Menetapkan pin A0 sebagai pin yang terhubung ke sensor kelembaban tanah.
- b) Setup:
  - `Serial.begin(9600)`: Memulai komunikasi serial dengan baud rate 9600.
- c) Loop:
  - `analogRead(SOIL_MOISTURE_PIN)`: Membaca nilai

analog dari sensor kelembaban tanah.

- `map(sensorValue, 0, 1023, 0, 100)`: Mengonversi nilai analog (0-1023) ke persentase kelembaban (0-100%).
- `Serial.print` dan `Serial.println`: Menampilkan nilai sensor dan persentase kelembaban di Serial Monitor.
- `delay(1000)`: Menunggu 1 detik sebelum melakukan pembacaan berikutnya.

### 3.3.2.5 Kesimpulan

Dengan mengikuti langkah-langkah di atas, Anda dapat dengan mudah membaca data kelembaban tanah menggunakan sensor kelembaban tanah dan Arduino. Ini adalah dasar yang baik untuk proyek-proyek yang memerlukan pemantauan kondisi tanah, seperti sistem irigasi otomatis atau aplikasi pertanian pintar. Dengan menambahkan komponen lain seperti pompa air atau sistem alarm, Anda dapat mengembangkan sistem yang lebih kompleks dan fungsional.

## 3.4 Pemrograman sensor suhu tanah

Pemrograman sensor suhu tanah pada Arduino adalah langkah penting untuk proyek-proyek yang berhubungan dengan monitoring kondisi tanah, seperti pertanian pintar atau sistem irigasi otomatis. Berikut adalah panduan langkah demi langkah untuk memprogram sensor suhu tanah menggunakan Arduino dan sensor suhu tanah yang umum seperti DS18B20.

### 3.4.1 Bahan yang Dibutuhkan

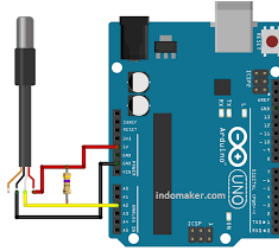
1. Arduino Board (misalnya, Arduino Uno)
2. Sensor Suhu Tanah DS18B20 (dalam bentuk tahan air untuk aplikasi luar ruangan)
3. Resistor 4.7k ohm (untuk pull-up)
4. Breadboard dan Kabel Jumper

### 3.4.2 Langkah-langkah Pemrograman

#### 3.4.2.1 Koneksi Hardware

- Pinout DS18B20:
- VCC: Power supply (3.3V atau 5V)
- GND: Ground
- DQ: Data pin (dianggap terhubung ke pin digital Arduino)

- Koneksi DS18B20 ke Arduino:
- VCC: Sambungkan ke 5V pada Arduino.
- GND: Sambungkan ke GND pada Arduino.
- DQ: Sambungkan ke pin digital 2 pada Arduino.
- Sambungkan resistor 4.7k ohm antara pin VCC dan DQ sebagai pull-up resistor.



Gambar 3-3 Rangkaian sensor DS18B20 pada arduino

### 3.4.2.2 Instalasi Library DS18B20

Untuk memudahkan pemrograman, gunakan library "OneWire" dan "DallasTemperature".

1. Buka Arduino IDE.
2. Pergi ke Sketch > Include Library > Manage Libraries.
3. Cari "OneWire" dan instal.
4. Cari "DallasTemperature" dan instal.

### 3.4.2.3 Kode Program

Berikut adalah contoh kode untuk membaca suhu tanah dari sensor DS18B20 dan menampilkan hasilnya di Serial Monitor.

```
#include <OneWire.h>
#include <DallasTemperature.h>

// Pin yang terhubung ke sensor DS18B20
#define ONE_WIRE_BUS 2

// Mengatur instance untuk komunikasi OneWire
OneWire oneWire(ONE_WIRE_BUS);

// Mengatur instance untuk sensor suhu Dallas
DallasTemperature sensors(&oneWire);

void setup() {
```



```

// Memulai komunikasi serial
Serial.begin(9600);
Serial.println("DS18B20 Soil Temperature Sensor Test");

// Memulai sensor
sensors.begin();
}

void loop() {
// Meminta sensor untuk membaca suhu
sensors.requestTemperatures();

// Membaca suhu dalam Celsius
float soilTempC = sensors.getTempCByIndex(0);

// Menampilkan data ke Serial Monitor
Serial.print("Soil Temperature: ");
Serial.print(soilTempC);
Serial.println(" *C");

// Menunggu sebelum pembacaan berikutnya
delay(2000);
}

```

#### 3.4.2.4 Menjalankan Program

1. Sambungkan Arduino ke komputer menggunakan kabel USB.
2. Buka Arduino IDE dan paste kode di atas ke dalam editor.
3. Pilih jenis board yang sesuai dan port yang terhubung ke Arduino melalui Tools > Board dan Tools > Port.
4. Unggah kode ke board dengan menekan tombol Upload.
5. Buka Serial Monitor dengan menekan tombol Serial Monitor di pojok kanan atas atau melalui Tools > Serial Monitor.
6. Pastikan baud rate diatur ke 9600 baud.
7. Anda akan melihat output suhu tanah dari sensor di Serial Monitor.

#### 3.4.2.5 Penjelasan Kode

- a) Inisialisasi Pin dan Library:
  - i. `#define ONE_WIRE_BUS 2`: Menetapkan pin 2 sebagai pin yang terhubung ke sensor DS18B20.

- ii. ``OneWire oneWire(ONE_WIRE_BUS)``: Mengatur instance untuk komunikasi OneWire.
- iii. ``DallasTemperature sensors(&oneWire)``: Mengatur instance untuk sensor suhu Dallas.
- b) **Setup**:
  - i. ``Serial.begin(9600)``: Memulai komunikasi serial dengan baud rate 9600.
  - ii. ``sensors.begin()``: Memulai sensor suhu DS18B20.
- c) **Loop**:
  - i. ``sensors.requestTemperatures()``: Meminta sensor untuk membaca suhu.
  - ii. ``float soilTempC = sensors.getTempCByIndex(0)``: Membaca suhu dalam Celsius.
  - iii. ``Serial.print`` dan ``Serial.println``: Menampilkan suhu tanah di Serial Monitor.
  - iv. ``delay(2000)``: Menunggu 2 detik sebelum melakukan pembacaan berikutnya.

#### 3.4.2.6 Kesimpulan

Dengan mengikuti langkah-langkah di atas, Anda dapat dengan mudah membaca data suhu tanah menggunakan sensor DS18B20 dan Arduino. Ini adalah dasar yang baik untuk proyek-proyek yang memerlukan pemantauan kondisi tanah, seperti sistem irigasi otomatis atau aplikasi pertanian pintar. Dengan menambahkan komponen lain seperti sensor kelembaban tanah atau pompa air, Anda dapat mengembangkan sistem yang lebih kompleks dan fungsional.

### 3.5 Pemrograman sensor tekanan

Pemrograman sensor tekanan pada Arduino dapat digunakan dalam berbagai aplikasi seperti pengukuran tekanan udara, tekanan air, atau tekanan mekanis. Salah satu sensor tekanan yang umum digunakan adalah sensor tekanan BMP180 atau BMP280 yang mengukur tekanan udara. Berikut adalah panduan langkah demi langkah untuk memprogram sensor tekanan BMP280 menggunakan Arduino.

#### 3.5.1 Bahan yang Dibutuhkan

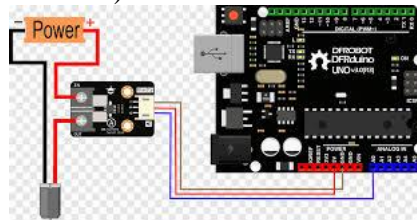
1. Arduino Board (misalnya, Arduino Uno)
2. Sensor Tekanan BMP280

### 3. Breadboard dan Kabel Jumper

#### 3.5.2 Langkah-langkah Pemrograman

##### 3.5.2.1 Koneksi Hardware

- Pinout BMP280:
- VIN: Power supply (3.3V atau 5V)
- GND: Ground
- SCL: Serial Clock Line
- SDA: Serial Data Line
- Koneksi BMP280 ke Arduino:
- VIN: Sambungkan ke 3.3V atau 5V pada Arduino (cek spesifikasi sensor Anda).
- GND: Sambungkan ke GND pada Arduino.
- SCL: Sambungkan ke pin A5 pada Arduino Uno (atau pin SCL pada board lain).
- SDA: Sambungkan ke pin A4 pada Arduino Uno (atau pin SDA pada board lain).



Gambar 3-4 Rangkaian sensor tekanan pada arduino

##### 3.5.2.2 Instalasi Library BMP280

Untuk memudahkan pemrograman, gunakan library Adafruit BMP280.

1. Buka Arduino IDE.
2. Pergi ke Sketch > Include Library > Manage Libraries.
3. Cari "Adafruit BMP280" dan instal.
4. Juga instal "Adafruit Unified Sensor" library.

##### 3.5.2.3 Kode Program

Berikut adalah contoh kode untuk membaca data tekanan dari sensor BMP280 dan menampilkan hasilnya di Serial Monitor.

```
#include <Wire.h>
```

```

#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>

// Inisialisasi objek sensor BMP280
Adafruit_BMP280 bmp;

void setup() {
  // Memulai komunikasi serial
  Serial.begin(9600);
  Serial.println(F("BMP280 test"));

  // Memulai sensor BMP280
  if (!bmp.begin()) {
    Serial.println(F("Could not find a valid BMP280 sensor, check
wiring!"));
    while (1);
  }

  // Mengatur pengaturan sensor
  bmp.setSampling(Adafruit_BMP280::MODE_NORMAL,
    Adafruit_BMP280::SAMPLING_X2, // temperature
    Adafruit_BMP280::SAMPLING_X16, // pressure
    Adafruit_BMP280::FILTER_X16,
    Adafruit_BMP280::STANDBY_MS_500);
}

void loop() {
  // Membaca tekanan dalam hPa (hectopascals)
  float pressure = bmp.readPressure() / 100.0F;

  // Menampilkan data ke Serial Monitor
  Serial.print(F("Pressure: "));
  Serial.print(pressure);
  Serial.println(F(" hPa"));

  // Menunggu sebelum pembacaan berikutnya
  delay(2000);
}

```

### 3.5.2.4 Menjalankan Program

- 1) Sambungkan Arduino ke komputer menggunakan kabel USB.
- 2) Buka Arduino IDE dan paste kode di atas ke dalam editor.
- 3) Pilih jenis board yang sesuai dan port yang terhubung ke Arduino melalui Tools > Board dan Tools > Port.
- 4) Unggah kode ke board dengan menekan tombol Upload.
- 5) Buka Serial Monitor dengan menekan tombol Serial Monitor di pojok kanan atas atau melalui Tools > Serial Monitor.
- 6) Pastikan baud rate diatur ke 9600 baud.
- 7) Anda akan melihat output nilai tekanan dari sensor di Serial Monitor.

### 3.5.2.5 Penjelasan Kode

- a) Inisialisasi Library dan Sensor:
  - `#include <Wire.h>`: Library untuk komunikasi I2C.
  - `#include <Adafruit_Sensor.h>`: Library untuk sensor Adafruit.
  - `#include <Adafruit_BMP280.h>`: Library khusus untuk sensor BMP280.
  - `Adafruit_BMP280 bmp`: Membuat objek sensor BMP280.
- b) Setup:
  - `Serial.begin(9600)`: Memulai komunikasi serial dengan baud rate 9600.
  - `bmp.begin()`: Memulai sensor BMP280 dan memeriksa koneksi.
  - `bmp.setSampling(...)`: Mengatur mode pengambilan sampel sensor.
- c) Loop:
  - a. `bmp.readPressure()`: Membaca nilai tekanan dari sensor.
  - b. `Serial.print` dan `Serial.println`: Menampilkan nilai tekanan di Serial Monitor.
  - c. `delay(2000)`: Menunggu 2 detik sebelum melakukan pembacaan berikutnya.

### 3.5.2.6 Kesimpulan

Dengan mengikuti langkah-langkah di atas, Anda dapat dengan mudah membaca data tekanan menggunakan sensor BMP280 dan Arduino. Ini adalah dasar yang baik untuk proyek-proyek yang memerlukan pemantauan tekanan udara, seperti stasiun cuaca atau aplikasi lingkungan lainnya. Dengan menambahkan komponen lain seperti sensor suhu atau kelembaban, Anda dapat mengembangkan sistem pemantauan yang lebih kompleks dan fungsional.

## 3.6 Pemrograman sensor gerak

Pemrograman sensor gerak pada Arduino adalah dasar yang sangat berguna untuk berbagai proyek, termasuk sistem keamanan, otomatisasi rumah, dan interaksi manusia-komputer. Sensor gerak yang umum digunakan adalah sensor PIR (Passive Infrared). Berikut adalah panduan langkah demi langkah untuk memprogram sensor gerak PIR menggunakan Arduino.

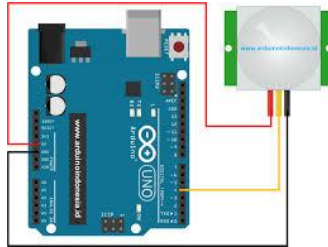
### 3.6.1 Bahan yang Dibutuhkan

1. Arduino Board (misalnya, Arduino Uno)
2. Sensor PIR (Passive Infrared)
3. Breadboard dan Kabel Jumper

### 3.6.2 Langkah-langkah Pemrograman

#### 3.6.2.1 Koneksi Hardware

- Pinout Sensor PIR:
- VCC: Power supply (3.3V atau 5V)
- GND: Ground
- OUT: Digital output (memberi sinyal HIGH ketika gerakan terdeteksi)
- Koneksi Sensor PIR ke Arduino:
- VCC: Sambungkan ke 5V pada Arduino.
- GND: Sambungkan ke GND pada Arduino.
- OUT: Sambungkan ke pin digital 2 pada Arduino.



Gambar 3-5 Rangkaian sensor gerak PIR pada Arduino

### 3.6.2.2 Kode Program

Berikut adalah contoh kode untuk membaca sinyal dari sensor PIR dan menampilkan hasilnya di Serial Monitor.

```
// Pin yang terhubung ke output sensor PIR
#define PIR_PIN 2

void setup() {
  // Memulai komunikasi serial
  Serial.begin(9600);

  // Mengatur pin PIR sebagai input
  pinMode(PIR_PIN, INPUT);

  Serial.println("PIR Sensor Test");
}

void loop() {
  // Membaca status dari sensor PIR
  int pirState = digitalRead(PIR_PIN);

  // Menampilkan status ke Serial Monitor
  if (pirState == HIGH) {
    Serial.println("Motion detected!");
  } else {
    Serial.println("No motion detected.");
  }

  // Menunggu sebelum pembacaan berikutnya
  delay(1000);
}
```

### 3.6.2.3 Menjalankan Program

- 1) Sambungkan Arduino ke komputer menggunakan kabel USB.
- 2) Buka Arduino IDE dan paste kode di atas ke dalam editor.
- 3) Pilih jenis board yang sesuai dan port yang terhubung ke Arduino melalui Tools > Board dan Tools > Port.
- 4) Unggah kode ke board dengan menekan tombol Upload.
- 5) Buka Serial Monitor dengan menekan tombol Serial Monitor di pojok kanan atas atau melalui Tools > Serial Monitor.
- 6) Pastikan baud rate diatur ke 9600 baud.
- 7) Anda akan melihat output deteksi gerakan dari sensor PIR di Serial Monitor.

### 3.6.2.4 Penjelasan Kode

- a. Inisialisasi Pin:
  - i. `#define PIR_PIN 2`: Menetapkan pin 2 sebagai pin yang terhubung ke output sensor PIR.
- b. Setup:
  - i. `Serial.begin(9600)`: Memulai komunikasi serial dengan baud rate 9600.
  - ii. `pinMode(PIR_PIN, INPUT)`: Mengatur pin PIR sebagai input.
  - iii. `Serial.println("PIR Sensor Test")`: Menampilkan pesan awal di Serial Monitor.
- c. Loop:
  - i. `digitalRead(PIR_PIN)`: Membaca status dari sensor PIR.
  - ii. `if (pirState == HIGH)`: Memeriksa apakah sensor mendeteksi gerakan (sinyal HIGH).
  - iii. `Serial.println("Motion detected!")`: Menampilkan pesan jika gerakan terdeteksi.
  - iv. `Serial.println("No motion detected.")`: Menampilkan pesan jika tidak ada gerakan.
  - v. `delay(1000)`: Menunggu 1 detik sebelum melakukan pembacaan berikutnya.

### 3.6.2.5 Kesimpulan

Dengan mengikuti langkah-langkah di atas, Anda dapat dengan mudah membaca data gerakan menggunakan sensor PIR dan Arduino. Ini adalah dasar yang baik untuk proyek-proyek yang memerlukan



deteksi gerakan, seperti sistem keamanan rumah atau otomatisasi pencahayaan. Dengan menambahkan komponen lain seperti buzzer atau lampu LED, Anda dapat mengembangkan sistem yang lebih kompleks dan fungsional.

### **3.7 Pemrograman Aktuator Relay**

Pemrograman aktuator relay pada ESP32 sangat berguna untuk mengontrol perangkat berdaya tinggi seperti lampu, kipas angin, atau motor melalui mikrokontroler. Relay memungkinkan mikrokontroler untuk mengendalikan rangkaian listrik terpisah. Berikut adalah panduan langkah demi langkah untuk memprogram aktuator relay menggunakan ESP32.

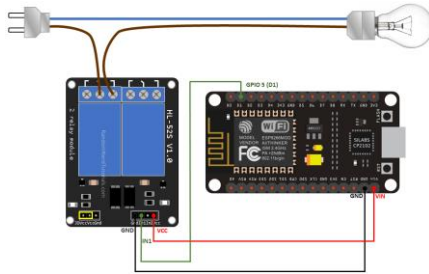
#### **3.7.1 Bahan yang Dibutuhkan**

1. ESP32 Board
2. Relay Module (biasanya modul 5V)
3. Breadboard dan Kabel Jumper

#### **3.7.2 Langkah-langkah Pemrograman**

##### **3.7.2.1 Koneksi Hardware**

- Pinout Relay Module:
  - VCC: Power supply (3.3V atau 5V)
  - GND: Ground
  - IN: Input control signal dari ESP32
  - COM, NO, NC: Terminal relay untuk menghubungkan perangkat yang akan dikendalikan
- Koneksi Relay Module ke ESP32:
  - VCC: Sambungkan ke 5V pada ESP32.
  - GND: Sambungkan ke GND pada ESP32.
  - IN: Sambungkan ke pin digital 23 pada ESP32 (atau pin lain yang tersedia).



Gambar 3-6 Rangkaian relay pada ESP32

### 3.7.2.2 Kode Program

Berikut adalah contoh kode untuk mengontrol relay menggunakan ESP32. Relay akan dinyalakan dan dimatikan secara bergantian dengan interval 1 detik.

```
// Pin yang terhubung ke input relay
#define RELAY_PIN 23

void setup() {
  // Mengatur pin relay sebagai output
  pinMode(RELAY_PIN, OUTPUT);

  // Memulai komunikasi serial
  Serial.begin(115200);
  Serial.println("Relay Control Test");
}

void loop() {
  // Menyalakan relay
  digitalWrite(RELAY_PIN, HIGH);
  Serial.println("Relay ON");
  delay(1000); // Menunggu 1 detik

  // Mematikan relay
  digitalWrite(RELAY_PIN, LOW);
  Serial.println("Relay OFF");
  delay(1000); // Menunggu 1 detik
}
```

### 3.7.2.3 Menjalankan Program

1. Sambungkan ESP32 ke komputer menggunakan kabel USB.
2. Buka Arduino IDE dan paste kode di atas ke dalam editor.
3. Pilih jenis board yang sesuai dan port yang terhubung ke ESP32 melalui Tools > Board dan Tools > Port.
4. Unggah kode ke board dengan menekan tombol Upload.
5. Buka Serial Monitor dengan menekan tombol Serial Monitor di pojok kanan atas atau melalui Tools > Serial Monitor.
6. Pastikan baud rate diatur ke 115200 baud.
7. Anda akan melihat output relay status dari ESP32 di Serial Monitor.

### 3.7.2.4 Penjelasan Kode

#### 1. Inisialisasi Pin:

- `#define RELAY_PIN 23`: Menetapkan pin 23 sebagai pin yang terhubung ke input relay.

#### 2. Setup:

- `pinMode(RELAY_PIN, OUTPUT)`: Mengatur pin relay sebagai output.
- `Serial.begin(115200)`: Memulai komunikasi serial dengan baud rate 115200.
- `Serial.println("Relay Control Test")`: Menampilkan pesan awal di Serial Monitor.

#### 3. Loop:

- `digitalWrite(RELAY_PIN, HIGH)`: Menyalakan relay.
- `Serial.println("Relay ON")`: Menampilkan pesan jika relay menyala.
- `delay(1000)`: Menunggu 1 detik.
- `digitalWrite(RELAY_PIN, LOW)`: Mematikan relay.
- `Serial.println("Relay OFF")`: Menampilkan pesan jika relay mati.
- `delay(1000)`: Menunggu 1 detik.

### 3.7.2.5 Kesimpulan

Dengan mengikuti langkah-langkah di atas, Anda dapat dengan mudah mengontrol relay menggunakan ESP32. Ini memungkinkan Anda untuk mengendalikan perangkat berdaya tinggi seperti lampu atau motor melalui mikrokontroler. Anda dapat mengembangkan sistem yang lebih kompleks dengan menambahkan sensor dan logika kontrol tambahan.

### **3.8 Pemrograman aktuator motor stepper**

Pemrograman aktuator motor stepper pada ESP32 memungkinkan Anda untuk mengontrol motor stepper dengan presisi, yang berguna dalam aplikasi seperti robotika, printer 3D, dan CNC. Berikut adalah panduan langkah demi langkah untuk memprogram motor stepper menggunakan ESP32 dan driver motor stepper seperti A4988 atau DRV8825.

#### **3.8.1 Bahan yang Dibutuhkan**

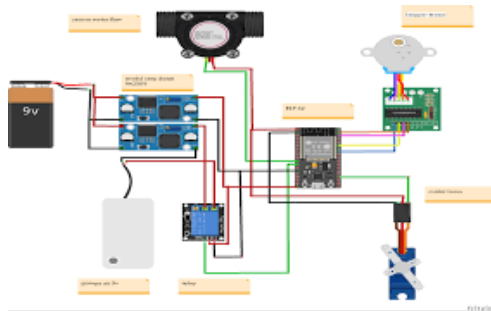
1. ESP32 Board
2. Motor Stepper (misalnya, NEMA 17)
3. Driver Motor Stepper (A4988 atau DRV8825)
4. Breadboard dan Kabel Jumper
5. Power Supply untuk Motor Stepper (biasanya 12V atau sesuai spesifikasi motor)

#### **3.8.2 Langkah-langkah Pemrograman**

##### **3.8.2.1 Koneksi Hardware**

- Pinout Driver A4988:
- VMOT: Power supply untuk motor (12V)
- GND: Ground
- 2B, 2A, 1A, 1B: Koneksi ke motor stepper
- VDD: Power supply untuk logic (3.3V atau 5V)
- GND: Ground
- STEP: Pin input step
- DIR: Pin input direction
- ENABLE: (Opsional) Pin enable/disable
- 
- Koneksi Driver A4988 ke ESP32 dan Motor Stepper:
- VMOT: Sambungkan ke 12V power supply.

- GND (motor side): Sambungkan ke ground power supply.
- 2B, 2A, 1A, 1B: Sambungkan ke koil motor stepper sesuai spesifikasi.
- VDD: Sambungkan ke 3.3V pada ESP32.
- GND (logic side): Sambungkan ke GND pada ESP32.
- STEP: Sambungkan ke pin digital 25 pada ESP32.
- DIR: Sambungkan ke pin digital 26 pada ESP32.
- ENABLE: (Opsional) Sambungkan ke pin digital 27 pada ESP32, atau ke GND untuk selalu aktif.



Gambar 3-7 Rangkaian motor stepper pada ESP32

### 3.8.2.2 Kode Program

Berikut adalah contoh kode untuk mengontrol motor stepper menggunakan driver A4988 dan ESP32. Motor akan berputar maju dan mundur dengan langkah tertentu.

```

...

// Definisikan pin untuk STEP, DIR, dan ENABLE (opsional)
#define STEP_PIN 25
#define DIR_PIN 26
#define ENABLE_PIN 27 // Opsional

// Jumlah langkah untuk satu putaran motor (misalnya, NEMA
17 dengan 200 langkah per putaran)
const int stepsPerRevolution = 200;

void setup() {
  // Mengatur pin sebagai output
  pinMode(STEP_PIN, OUTPUT);
  pinMode(DIR_PIN, OUTPUT);
  pinMode(ENABLE_PIN, OUTPUT); // Opsional

```

```

    // Aktifkan driver motor stepper (jika menggunakan pin
ENABLE)
    digitalWrite(ENABLE_PIN, LOW);

    // Memulai komunikasi serial
    Serial.begin(115200);
    Serial.println("Motor Stepper Control Test");
}

void loop() {
    // Mengatur arah motor searah jarum jam
    digitalWrite(DIR_PIN, HIGH);

    // Menjalankan motor satu putaran searah jarum jam
    for (int i = 0; i < stepsPerRevolution; i++) {
        digitalWrite(STEP_PIN, HIGH);
        delayMicroseconds(1000); // Sesuaikan sesuai kebutuhan
        digitalWrite(STEP_PIN, LOW);
        delayMicroseconds(1000); // Sesuaikan sesuai kebutuhan
    }

    // Menunggu sebelum beralih arah
    delay(1000);

    // Mengatur arah motor berlawanan arah jarum jam
    digitalWrite(DIR_PIN, LOW);

    // Menjalankan motor satu putaran berlawanan arah jarum jam
    for (int i = 0; i < stepsPerRevolution; i++) {
        digitalWrite(STEP_PIN, HIGH);
        delayMicroseconds(1000); // Sesuaikan sesuai kebutuhan
        digitalWrite(STEP_PIN, LOW);
        delayMicroseconds(1000); // Sesuaikan sesuai kebutuhan
    }

    // Menunggu sebelum beralih arah
    delay(1000);
}
...

```

### 3.8.2.3 Menjalankan Program

1. Sambungkan ESP32 ke komputer menggunakan kabel USB.
2. Buka Arduino IDE dan paste kode di atas ke dalam editor.
3. Pilih jenis board yang sesuai dan port yang terhubung ke ESP32 melalui Tools > Board dan Tools > Port.
4. Unggah kode ke board dengan menekan tombol Upload.
5. Buka Serial Monitor dengan menekan tombol Serial Monitor di pojok kanan atas atau melalui Tools > Serial Monitor.
6. Pastikan baud rate diatur ke 115200 baud.
7. Anda akan melihat output kontrol motor stepper dari ESP32 di Serial Monitor, dan motor akan bergerak maju dan mundur.

### 3.8.2.4 Penjelasan Kode

#### 1. Inisialisasi Pin:

- ``#define STEP_PIN 25``: Menetapkan pin 25 sebagai pin yang terhubung ke input step driver.
- ``#define DIR_PIN 26``: Menetapkan pin 26 sebagai pin yang terhubung ke input direction driver.
- ``#define ENABLE_PIN 27``: (Opsional) Menetapkan pin 27 sebagai pin yang terhubung ke input enable driver.

#### 2. Setup:

- ``pinMode(STEP_PIN, OUTPUT)``: Mengatur pin step sebagai output.
- ``pinMode(DIR_PIN, OUTPUT)``: Mengatur pin direction sebagai output.
- ``pinMode(ENABLE_PIN, OUTPUT)``: (Opsional) Mengatur pin enable sebagai output.
- ``digitalWrite(ENABLE_PIN, LOW)``: (Opsional) Mengaktifkan driver motor stepper.
- ``Serial.begin(115200)``: Memulai komunikasi serial dengan baud rate 115200.
- ``Serial.println("Motor Stepper Control Test")``: Menampilkan pesan awal di Serial Monitor.

#### 3. Loop:

- ``digitalWrite(DIR_PIN, HIGH)``: Mengatur arah motor searah jarum jam.
- ``for (int i = 0; i < stepsPerRevolution; i++)``:

- Melakukan satu putaran searah jarum jam.
- - ``digitalWrite(STEP_PIN, HIGH)``: Mengirim sinyal HIGH ke pin step.
- - ``delayMicroseconds(1000)``: Menunggu 1000 mikrodetik.
- - ``digitalWrite(STEP_PIN, LOW)``: Mengirim sinyal LOW ke pin step.
- - ``delayMicroseconds(1000)``: Menunggu 1000 mikrodetik.
- - ``delay(1000)``: Menunggu 1 detik sebelum beralih arah.
- - ``digitalWrite(DIR_PIN, LOW)``: Mengatur arah motor berlawanan arah jarum jam.
- - ``for (int i = 0; i < stepsPerRevolution; i++)``: Melakukan satu putaran berlawanan arah jarum jam.
- - Langkah yang sama seperti di atas untuk bergerak berlawanan arah jarum jam.
- - ``delay(1000)``: Menunggu 1 detik sebelum beralih arah lagi.

### 3.8.2.5 Kesimpulan

Dengan mengikuti langkah-langkah di atas, Anda dapat dengan mudah mengontrol motor stepper menggunakan ESP32. Ini memungkinkan Anda untuk mengendalikan motor dengan presisi tinggi dalam aplikasi seperti robotika, CNC, dan printer 3D. Anda dapat mengembangkan sistem yang lebih kompleks dengan menambahkan kontrol tambahan dan sensor untuk berbagai aplikasi.

## 3.9 Rangkuman

Pemrograman sensor dan aktuator pada ESP32 dan Arduino melibatkan penggunaan berbagai perangkat keras dan perangkat lunak untuk membaca data dari lingkungan dan mengontrol perangkat output. Kedua platform menggunakan bahasa pemrograman C/C++ di dalam Arduino IDE, tetapi ESP32 menawarkan kemampuan tambahan seperti konektivitas WiFi dan Bluetooth.

### 1. Pengenalan Platform

**Arduino**: Platform mikrokontroler open-source yang sangat



populer untuk proyek DIY dan edukasi. Mudah digunakan dengan berbagai jenis sensor dan aktuator.

**ESP32:** Mikrokontroler yang lebih canggih dari Espressif Systems dengan fitur tambahan seperti WiFi dan Bluetooth, memberikan fleksibilitas lebih besar untuk aplikasi IoT.

## 2. Dasar Pemrograman

**Setup:** Fungsi `setup()` dijalankan sekali saat board di-reset atau dinyalakan untuk inisialisasi.

**Loop:** Fungsi `loop()` dijalankan berulang kali selama board aktif, tempat di mana logika utama program ditulis.

Pemrograman sensor dan aktuator pada ESP32 dan Arduino memberikan kemampuan untuk membuat berbagai aplikasi elektronik. Arduino adalah platform yang sangat baik untuk pemula karena kemudahannya, sedangkan ESP32 menawarkan fitur tambahan seperti WiFi dan Bluetooth yang cocok untuk proyek IoT yang lebih kompleks. Memahami dasar pemrograman, inisialisasi pin, dan cara membaca data sensor serta mengendalikan aktuator adalah kunci untuk berhasil dalam proyek berbasis mikrokontroler ini.

## 3.10 Latihan soal

**Soal:** Buatlah program untuk membaca data suhu dan kelembaban dari sensor DHT11 yang terhubung ke pin digital 2 pada Arduino, dan tampilkan hasilnya di Serial Monitor setiap 2 detik.

**Petunjuk:**

Gunakan library DHT untuk membaca data dari sensor.

Inisialisasi sensor pada `setup()`.

Baca data suhu dan kelembaban pada `loop()` dan tampilkan di Serial Monitor.

## 3.11 Bahan Diskusi

### 1. Integrasi Sensor

Sensor Suhu dan Kelembaban: Seperti DHT11/DHT22.

Sensor Kelembaban Tanah, Sensor Cahaya, dll.

Poin Diskusi:

Cara kerja berbagai sensor dan cara mengintegrasikannya dengan Arduino dan ESP32.

Contoh kode dan pengalaman penggunaan sensor.

### 2. Proyek Nyata

Proyek IoT: Seperti monitoring suhu dan kelembaban secara real-time.

Automasi Rumah: Menggunakan relay dan sensor.

Robotik: Menggunakan motor stepper dan sensor.

Poin Diskusi:

Proyek yang telah dilakukan oleh peserta diskusi.

Tantangan dan solusi yang ditemukan selama proyek.

### 3.12 Rujukan

- Harmini, H., & Nurhayati, T. (2020). Desain Solar Power Inverter pada Sistem Photovoltaic. *Elektrika*, 1.
- kece, i. (2021, Juli 31). <https://iotkece.com>. Retrieved from apa-itu-esp32-spesifikasi-esp32: [https://iotkece.com/apa-itu-esp32-spesifikasi-esp32/#google\\_vignette](https://iotkece.com/apa-itu-esp32-spesifikasi-esp32/#google_vignette)
- Manager, T. C. (2018, Oktober 3). <https://www.topcropmanager.com>. Retrieved from New field monitoring system from Hoskins Scientific: <https://www.topcropmanager.com/new-field-monitoring-system-from-hoskins-scientific-21485/>
- Sulistiyanti, S., & Purwiyanti, ., P. (2020). *Sensor dan Prinsip Kerjanya*. Lampung: Pustaka Media.
- Yulianti, S. (2022, Februari 23). <https://www.indoniaga.co.id>. Retrieved from pertanian-green-house-berbasis-iot: <https://www.indoniaga.co.id/2022/02/pertanian-green-house-berbasis-iot.html>
- Yusro, M., & Diamah, A. (2019). *Sensor dan Transduser*. Jakarta: Fakultas Teknik Universitas Negeri Jakarta.

## BAB 4. PROTOKOL KOMUNIKASI DALAM IOT

### Capaian Pembelajaran Matakuliah (CPMK)

Mahasiswa mampu membuat interface / API yang menghubungkan embedded system ke server

Mahasiswa mampu menggunakan protokol komunikasi dalam iot

### 4.1 Pendahuluan

Dalam sistem IoT (*Internet of Things*), perangkat sering kali berkomunikasi dengan server atau perangkat lain untuk mengirim dan menerima data. Protokol dalam IoT memainkan peran penting dalam memungkinkan perangkat yang berbeda berkomunikasi dan bertukar data secara efektif. Fungsi utama protokol dalam IoT mencakup berbagai aspek mulai dari pengelolaan koneksi, pengiriman data, hingga keamanan dan manajemen perangkat. Berikut adalah fungsi-fungsi utama protokol dalam IoT:

1. Manajemen Koneksi

Protokol IoT mengelola pembentukan, pemeliharaan, dan penghentian koneksi antara perangkat. Menginisiasi koneksi, menjaga koneksi tetap aktif, dan menutup koneksi dengan benar.

Contoh Protokol: MQTT, CoAP

2. Pengiriman dan Pengaturan Data

Protokol mengatur cara data dikemas, dikirim, diterima, dan diuraikan oleh perangkat IoT. Mengirim data sensor, menerima perintah dari server, mengelola format data (JSON, XML).

Contoh Protokol: HTTP/HTTPS, MQTT, CoAP

3. Keandalan dan Jaminan Pengiriman

Beberapa protokol menyediakan mekanisme untuk memastikan bahwa data dikirim dan diterima dengan benar. Menggunakan pengakuan (acknowledgment), pengiriman ulang data jika terjadi kesalahan, memastikan integritas data.

Contoh Protokol: MQTT (Quality of Service), CoAP (Confirmable Messages)

4. Keamanan Data

Protokol IoT mengimplementasikan berbagai mekanisme untuk melindungi data dari akses yang tidak sah dan serangan. Fungsi enkripsi data selama pengiriman, otentikasi perangkat, integritas pesan

Contoh Protokol: HTTPS, MQTT dengan TLS, CoAP dengan DTLS

5. Efisiensi Penggunaan Sumber Daya

Protokol dirancang untuk mengoptimalkan penggunaan daya dan bandwidth, terutama untuk perangkat dengan sumber daya terbatas. Fungsi mengurangi overhead data, menghemat konsumsi daya, efisiensi penggunaan jaringan.

Contoh Protokol: MQTT, CoAP, LoRaWAN

6. Manajemen Jaringan

Protokol membantu dalam pengelolaan topologi jaringan, seperti pengaturan jaringan mesh atau jaringan ad-hoc. Fungsi menentukan rute terbaik untuk pengiriman data, memperbaiki jaringan secara otomatis, pengaturan ulang topologi jika ada perangkat yang gagal.

Contoh Protokol: Zigbee, Z-Wave, 6LoWPAN

7. Interoperabilitas

Protokol memastikan bahwa perangkat dari berbagai produsen dapat berkomunikasi satu sama lain. Fungsi menggunakan standar terbuka yang diakui secara luas, kompatibilitas dengan berbagai platform dan perangkat.

Contoh Protokol: HTTP, MQTT, CoAP

8. Penskalaan

Protokol mendukung penskalaan dari beberapa perangkat hingga ribuan perangkat dalam jaringan yang sama. Fungsi mendukung banyak perangkat dengan efisiensi yang sama, manajemen beban, distribusi data yang efektif.

Contoh Protokol: MQTT, CoAP

Pemilihan protokol komunikasi yang tepat sangat penting untuk memastikan efisiensi, keandalan, dan keamanan komunikasi. Berikut adalah beberapa protokol komunikasi yang umum digunakan dalam IoT:

1. HTTP (*Hypertext Transfer Protocol*)

Protokol standar untuk komunikasi di web. Digunakan untuk mengirim dan menerima data antara klien (seperti browser web) dan server.

- Keunggulan: Mudah digunakan dan didukung luas oleh

- perangkat dan layanan web.
- Kelemahan: Relatif berat karena overhead HTTP, tidak ideal untuk perangkat dengan sumber daya terbatas.
2. **HTTPS (*HTTP Secure*)**

Versi aman dari HTTP yang menggunakan SSL/TLS untuk mengenkripsi data yang ditransmisikan.

    - Keunggulan: Menyediakan keamanan yang lebih tinggi melalui enkripsi.
    - Kelemahan: Memerlukan lebih banyak sumber daya untuk enkripsi dan dekripsi data.
  3. **MQTT (*Message Queuing Telemetry Transport*)**

Protokol pesan ringan yang dirancang untuk konektivitas perangkat dengan bandwidth rendah dan keterbatasan daya.

    - Keunggulan: Sangat efisien, ideal untuk jaringan yang tidak stabil atau memiliki latensi tinggi.
    - Kelemahan: Diperlukan broker MQTT untuk mengelola komunikasi.
  4. **CoAP (*Constrained Application Protocol*)**

Protokol aplikasi untuk perangkat yang dibatasi sumber daya, mirip dengan HTTP tetapi lebih ringan.

    - Keunggulan: Efisien dalam penggunaan bandwidth dan daya.
    - Kelemahan: Kurang dikenal dan didukung dibandingkan dengan HTTP.
  5. **LoRaWAN (*Long Range Wide Area Network*)**

Protokol jaringan yang dirancang untuk perangkat IoT jarak jauh dengan konsumsi daya sangat rendah.

    - Keunggulan: Jangkauan yang sangat luas, ideal untuk aplikasi seperti pertanian pintar.
    - Kelemahan: Kecepatan data yang rendah.
  6. **Bluetooth Low Energy (BLE)**

Versi hemat daya dari Bluetooth yang dirancang untuk aplikasi IoT.

    - Keunggulan: Konsumsi daya rendah, ideal untuk perangkat yang perlu bekerja dengan baterai untuk waktu lama.
    - Kelemahan: Jarak komunikasi terbatas.
  7. **Zigbee**

Protokol komunikasi untuk jaringan area pribadi (PAN) dengan konsumsi daya rendah.

- Keunggulan: Efisien dalam penggunaan daya, mendukung jaringan mesh.
  - Kelemahan: Jarak komunikasi yang lebih pendek dibandingkan dengan beberapa protokol lain.
8. Z-Wave
- Protokol nirkabel yang dirancang khusus untuk automasi rumah dan aplikasi IoT lainnya.
- Keunggulan: Konsumsi daya rendah, jangkauan yang baik.
  - Kelemahan: Kurang didukung luas dibandingkan dengan protokol lain seperti Zigbee.
9. 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks)
- Protokol yang memungkinkan perangkat IoT menggunakan IPv6 pada jaringan PAN dengan daya rendah.
- Keunggulan: Integrasi dengan internet melalui IPv6.
  - Kelemahan: Kompleksitas yang lebih tinggi dibandingkan dengan beberapa protokol lain.
10. Cellular (2G/3G/4G/5G)
- Menggunakan jaringan seluler untuk komunikasi data.
- Keunggulan: Jangkauan global, kecepatan data tinggi (terutama dengan 4G dan 5G).
  - Kelemahan: Konsumsi daya yang lebih tinggi dan biaya operasional.

## **4.2 Contoh Implementasi Protokol**

Bagian ini menjelaskan tentang pemakaian beberapa protokol yang sering digunakan dalam sistem IOT.

### **4.2.1 Implementasi HTTP pada ESP32**

Berikut adalah contoh sederhana tentang cara menggunakan WiFi pada ESP32 untuk terhubung ke jaringan WiFi dan mengirimkan data ke server menggunakan HTTP. Contoh ini menunjukkan bagaimana mengatur koneksi WiFi, membuat permintaan HTTP GET, dan membaca respons dari server.

#### **4.2.1.1 Menghubungkan ESP32 ke Jaringan WiFi**

Kode berikut ini menunjukkan bagaimana menghubungkan ESP32 ke jaringan WiFi menggunakan library `WiFi.h`.

```
...  
#include <WiFi.h>  
  
// Ganti dengan SSID dan Password WiFi Anda  
const char* ssid = "your_SSID";  
const char* password = "your_PASSWORD";  
  
void setup() {  
  Serial.begin(115200);  
  delay(10);  
  
  // Mulai koneksi WiFi  
  WiFi.begin(ssid, password);  
  Serial.print("Connecting to WiFi");  
  
  // Tunggu hingga terhubung  
  while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
  }  
  
  Serial.println("");  
  Serial.println("WiFi connected");  
  Serial.print("IP address: ");  
  Serial.println(WiFi.localIP());  
}  
  
void loop() {  
  // Tidak ada yang dilakukan di loop  
}  
...
```

#### 4.2.1.2 Mengirim Permintaan HTTP GET

Setelah terhubung ke jaringan WiFi, Anda dapat mengirim permintaan HTTP GET ke server. Berikut adalah kode contoh untuk mengirim permintaan HTTP GET menggunakan library `WiFiClient`.

```

```cpp
#include <WiFi.h>
#include <HTTPClient.h>

// Ganti dengan SSID dan Password WiFi Anda
const char* ssid = "your_SSID";
const char* password = "your_PASSWORD";

// Ganti dengan URL server Anda
const char* serverName = "http://example.com/data";

void setup() {
  Serial.begin(115200);
  delay(10);

  // Mulai koneksi WiFi
  WiFi.begin(ssid, password);
  Serial.print("Connecting to WiFi");

  // Tunggu hingga terhubung
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
}

void loop() {
  if (WiFi.status() == WL_CONNECTED) { // Pastikan
    terhubung ke WiFi
      HTTPClient http;

      // Mulai koneksi dan kirim HTTP header
      http.begin(serverName);

      // Kirim permintaan HTTP GET

```



```

int httpCode = http.GET();

// Periksa hasil dari permintaan HTTP GET
if (httpCode > 0) {
    // HTTP header kirim berhasil
    Serial.printf("HTTP GET request sent, response code:
%d\n", httpCode);

    // Jika respon kode adalah HTTP_OK, baca respons
    if (httpCode == HTTP_CODE_OK) {
        String payload = http.getString();
        Serial.println("Response payload: " + payload);
    }
    } else {
        Serial.printf("HTTP GET request failed, error: %s\n",
http.errorToString(httpCode).c_str());
    }

    // Tutup koneksi
    http.end();
}

delay(10000); // Tunggu 10 detik sebelum mengirim
permintaan berikutnya
}
...

```

#### 4.2.1.3 Penjelasan

1. Menghubungkan ke WiFi:\*\*
  - Library `WiFi.h` digunakan untuk mengelola koneksi WiFi pada ESP32.
  - `WiFi.begin(ssid, password);` memulai koneksi ke jaringan WiFi.
  - `WiFi.status() != WL\_CONNECTED` mengecek status koneksi, dan program akan menunggu hingga terhubung.
  - `WiFi.localIP()` mendapatkan alamat IP yang diberikan oleh jaringan WiFi.
2. Mengirim Permintaan HTTP GET:\*\*

- Library `HTTPClient.h` digunakan untuk mengirim permintaan HTTP.
- `http.begin(serverName);` memulai koneksi ke server dengan URL yang ditentukan.
- `http.GET();` mengirim permintaan HTTP GET ke server.
- `http.getString();` membaca respons dari server jika permintaan berhasil.

Dengan menggunakan contoh di atas, Anda dapat menghubungkan ESP32 ke jaringan WiFi dan berkomunikasi dengan server menggunakan permintaan HTTP GET. Anda juga bisa mengubah kode untuk menggunakan permintaan HTTP POST atau protokol lain sesuai kebutuhan proyek Anda.

#### 4.2.2 Implementasi MQTT pada ESP32

Berikut adalah contoh implementasi MQTT pada ESP32 menggunakan library `PubSubClient`. Contoh ini akan menunjukkan bagaimana menghubungkan ESP32 ke broker MQTT, menerbitkan pesan, dan berlangganan topik untuk menerima pesan.

Sebelum memulai, pastikan Anda telah menginstal library `PubSubClient` pada Arduino IDE.

##### 4.2.2.1 Menyiapkan Konfigurasi

Kode program

```
...
```

```
#include <WiFi.h>
```

```
#include <PubSubClient.h>
```

```
// Konfigurasi WiFi
```

```
const char* ssid = "your_wifi_ssid";
```

```
const char* password = "your_wifi_password";
```

```
// Konfigurasi MQTT broker
```

```
const char* mqtt_server = "your_mqtt_broker_address";
```

```
const int mqtt_port = 1883;
```

```
// Konfigurasi topik MQTT
```

```
const char* topic = "test/topic";
```

```

// Fungsi callback untuk menerima pesan yang diterbitkan ke
topik yang di-subscribe
void callback(char* topic, byte* message, unsigned int length) {
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
    for (int i = 0; i < length; i++) {
        Serial.print((char)message[i]);
    }
    Serial.println();
}

```

```

WiFiClient espClient;
PubSubClient client(espClient);

```

```

void setup_wifi() {
    delay(10);
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

```

```

    WiFi.begin(ssid, password);

```

```

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

```

```

    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}

```

```

void reconnect() {
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        if (client.connect("ESP32Client")) {
            Serial.println("connected");
            client.subscribe(topic);

```

```

    } else {
        Serial.print("failed, rc=");
        Serial.print(client.state());
        Serial.println(" try again in 5 seconds");
        delay(5000);
    }
}

void setup() {
    Serial.begin(115200);
    setup_wifi();
    client.setServer(mqtt_server, mqtt_port);
    client.setCallback(callback);
}

void loop() {
    if (!client.connected()) {
        reconnect();
    }
    client.loop();

    // Kirim pesan ke topik MQTT
    client.publish(topic, "Hello from ESP32");

    delay(5000); // Tunggu 5 detik sebelum mengirim pesan
    berikutnya
}
...

```

#### 4.2.2.2 Menghubungkan ke WiFi

- Dalam fungsi `setup\_wifi()`, ESP32 akan terhubung ke jaringan WiFi yang ditentukan menggunakan SSID dan kata sandi yang diberikan.

#### 4.2.2.3 Menghubungkan ke Broker MQTT

- Dalam fungsi `reconnect()`, ESP32 akan mencoba untuk

terhubung ke broker MQTT yang ditentukan. Jika berhasil terhubung, ESP32 akan berlangganan topik `test/topic`.

#### 4.2.2.4 Menerima Pesan dari Broker MQTT

- Fungsi callback `callback()` akan dipanggil setiap kali ada pesan yang diterima dari topik yang di-subscribe. Pesan yang diterima akan dicetak ke Serial Monitor.

#### 4.2.2.5 Mengirim Pesan ke Broker MQTT

- Dalam loop `loop()`, ESP32 akan mengirimkan pesan "Hello from ESP32" ke topik `test/topic` setiap 5 detik.

Dengan mengikuti langkah-langkah ini, Anda akan dapat mengimplementasikan MQTT pada ESP32 dan berkomunikasi dengan broker MQTT untuk menerbitkan dan menerima pesan. Pastikan untuk menyesuaikan konfigurasi WiFi dan broker MQTT sesuai dengan lingkungan Anda.

#### 4.2.3 Implementasi Lora pada ESP32

Implementasi LoRaWAN pada ESP32 memerlukan penggunaan modul LoRa seperti SX127x dan library LoRaWAN yang kompatibel dengan ESP32. Berikut adalah langkah-langkah umum untuk mengimplementasikan LoRaWAN pada ESP32:

##### 4.2.3.1 Menyiapkan Perangkat dan Lingkungan Pengembangan

1. Persiapkan Modul LoRa: Pastikan Anda telah terhubung dengan modul LoRa SX127x ke ESP32. Anda dapat menggunakan modul SX1278 atau SX1276 yang kompatibel dengan ESP32.
2. Unduh Library LoRaWAN: Unduh dan instal library LoRaWAN yang kompatibel dengan ESP32. Beberapa library yang populer untuk LoRaWAN di ESP32 antara lain adalah Arduino-LoRaWAN dan TheThingsNetwork Arduino Library.

##### 4.2.3.2 Konfigurasi LoRaWAN

1. Konfigurasi Perangkat LoRa: Tentukan frekuensi, faktor spreading (spreading factor), bandwidth, dan kekuatan pengirim daya (transmit power) yang sesuai dengan regulasi

lokal dan kebutuhan jaringan.

2. Konfigurasi Koneksi: Atur konfigurasi LoRaWAN, seperti DevEUI, AppEUI, AppKey, dan alamat jaringan (DevAddr), yang diberikan oleh penyedia jaringan LoRaWAN (seperti The Things Network).

#### 4.2.3.3 Implementasi Kode

Berikut adalah contoh sederhana kode untuk mengimplementasikan LoRaWAN pada ESP32 menggunakan library TheThingsNetwork Arduino Library:

```

...
#include <TheThingsNetwork.h>

#define DEV_ADDR "your_dev_addr"
#define APP_EUI "your_app_eui"
#define APP_KEY "your_app_key"

#define freqPlan TTN_FP_EU868

TheThingsNetwork ttn(LORA_SS, LORA_IRQ, LORA_RST,
freqPlan);

void setup() {
  Serial.begin(115200);
  while (!Serial);

  // Inisialisasi koneksi LoRaWAN
  ttn.begin();

  // Konfigurasi kunci perangkat
  ttn.personalize(DEV_ADDR, APP_EUI, APP_KEY);
}

void loop() {
  // Kirim pesan ke server LoRaWAN
  ttn.sendBytes("Hello, LoRaWAN!");

  // Tunggu 10 detik sebelum mengirim pesan berikutnya
  delay(10000);
}
```

...

#### 4.2.3.4 Pengujian

1. Mengunggah Kode: Unggah kode ke ESP32 menggunakan Arduino IDE atau platform pengembangan lainnya.
2. Monitor Serial: Buka monitor serial pada Arduino IDE atau aplikasi sejenisnya untuk melihat keluaran debug dari ESP32. Hal ini membantu dalam memeriksa apakah perangkat berhasil terhubung ke jaringan LoRaWAN dan mengirimkan data.
3. Pemantauan Jaringan: Pemantauan jaringan LoRaWAN, seperti The Things Network Console, dapat digunakan untuk melacak aktivitas perangkat dan menerima data yang dikirimkan.

Dengan mengikuti langkah-langkah ini, Anda dapat mengimplementasikan LoRaWAN pada ESP32 dan menggunakan jaringan LoRaWAN untuk mengirimkan data secara nirkabel. Pastikan untuk mengikuti pedoman regulasi lokal dan petunjuk dari penyedia jaringan LoRaWAN yang Anda gunakan.

#### 4.2.4 Implementasi Cellular pada ESP32

Implementasi protokol seluler pada ESP32 memerlukan penggunaan modul seluler eksternal dan library yang mendukung komunikasi dengan modul tersebut. Berikut adalah langkah-langkah umum untuk mengimplementasikan protokol seluler pada ESP32:

##### 4.2.4.1 Persiapan Perangkat

1. Pilih Modul Seluler  
Pilih modul seluler eksternal yang kompatibel dengan ESP32, seperti SIM800L atau SIM900. Pastikan modul tersebut mendukung protokol seluler yang digunakan di wilayah Anda (misalnya, GSM, GPRS, atau 3G).
2. Hubungkan Modul Seluler ke ESP32  
Hubungkan modul seluler ke ESP32 menggunakan komunikasi serial (UART). Pastikan Anda telah menghubungkan kabel TX dan RX secara benar antara ESP32 dan modul seluler.

##### 4.2.4.2 Instalasi Library

Instal library yang mendukung komunikasi dengan modul seluler yang Anda gunakan. Misalnya, jika Anda menggunakan modul

SIM800L, Anda dapat menggunakan library seperti TinyGSM.

#### 4.2.4.3 Implementasi Kode

Berikut adalah contoh sederhana kode untuk mengirimkan pesan SMS menggunakan modul SIM800L dan library TinyGSM pada ESP32:

```
...
#include <TinyGsmClient.h>
#include <Wire.h>

#define RX_PIN 16
#define TX_PIN 17

#define SIM800L_TINY_GSM_IP5306_VERSION_20190529
#include <TinyGsmClient.h>
#include <Wire.h>

const char apn[] = "your_apn";
const char gprsUser[] = "";
const char gprsPass[] = "";

const char simPIN[] = "1234";

TinyGsm modem(SerialAT);
TinyGsmClient client(modem);

void setup() {
  Serial.begin(115200);
  SerialAT.begin(115200, SERIAL_8N1, RX_PIN, TX_PIN);

  Wire.begin(I2C_SDA, I2C_SCL, 400000);

  delay(10);

  Serial.println("Initializing modem...");
  modem.restart();

  String modemInfo = modem.getModemInfo();
  Serial.print("Modem Info: ");
```



```

Serial.println(modemInfo);

Serial.print("Waiting for network...");
if (!modem.waitForNetwork()) {
    Serial.println(" fail");
    delay(10000);
    return;
}
Serial.println(" OK");

if (modem.isNetworkConnected()) {
    Serial.println("Network connected");
}

Serial.print("Connecting to ");
Serial.print(apn);
if (!modem.gprsConnect(apn, gprsUser, gprsPass)) {
    Serial.println(" fail");
    delay(10000);
    return;
}
Serial.println(" OK");

if (modem.isGprsConnected()) {
    Serial.println("GPRS connected");
}
}

void loop() {
    Serial.println("Sending SMS...");
    if (modem.sendSMS("+1234567890", "Hello from ESP32!")) {
        Serial.println("SMS sent successfully");
    } else {
        Serial.println("SMS sending failed");
    }
    delay(10000); // Kirim pesan setiap 10 detik
}
...

```

#### 4.2.4.4 Pengujian

1. Mengunggah Kode

Unggah kode ke ESP32 menggunakan Arduino IDE atau platform pengembangan lainnya.

2. Monitor Serial

Buka monitor serial pada Arduino IDE atau aplikasi sejenisnya untuk melihat keluaran debug dari ESP32. Hal ini membantu dalam memeriksa apakah ESP32 berhasil terhubung ke modul seluler dan mengirimkan pesan SMS.

3. Penerimaan Pesan

Pastikan Anda telah mengatur nomor penerima dengan benar. Anda dapat menggunakan ponsel untuk memeriksa apakah pesan SMS telah diterima oleh penerima yang dituju.

Dengan mengikuti langkah-langkah ini, Anda dapat mengimplementasikan protokol seluler pada ESP32 untuk berkomunikasi melalui jaringan seluler dan melakukan tugas-tugas seperti mengirim pesan SMS, mengakses internet, atau berkomunikasi dengan server jarak jauh. Pastikan untuk mengonfigurasi modul seluler dan ESP32 dengan benar sesuai dengan kebutuhan Anda.

#### 4.2.5 Implementasi *Bluetooth Low Energy* pada ESP32

Implementasi Bluetooth Low Energy (BLE) pada ESP32 dapat dilakukan menggunakan perangkat keras dan perangkat lunak yang sesuai. Berikut langkah-langkah umumnya.

##### 4.2.5.1 Persiapan Perangkat

Pastikan Anda memiliki ESP32 yang didukung BLE. Mayoritas modul ESP32 mendukung BLE, tetapi pastikan Anda memiliki modul yang sesuai.

##### 4.2.5.2 Instalasi Library

Unduh dan instalasi library yang diperlukan untuk memprogram ESP32 dengan menggunakan BLE. Library yang populer untuk pengembangan BLE pada ESP32 adalah ArduinoBLE. Anda dapat menginstalnya melalui Arduino Library Manager.

##### 4.2.5.3 Implementasi Kode

Buat sketsa Arduino yang menggunakan library ArduinoBLE

untuk mengatur perangkat ESP32 sebagai perangkat BLE peripheral atau central. Berikut adalah contoh sederhana untuk membuat ESP32 sebagai perangkat BLE peripheral yang mengirimkan data:

```

...

#include <ArduinoBLE.h>

BLEService      customService("19B10000-E8F2-537E-4F6C-
D104768A1214"); // UUID servis kustom

BLECharacteristic      customCharacteristic("19B10001-E8F2-
537E-4F6C-D104768A1214", BLERead | BLEWrite, 20); // UUID
karakteristik kustom

void setup() {
  Serial.begin(9600);
  while (!Serial);

  if (!BLE.begin()) {
    Serial.println("Starting BLE failed!");
    while (1);
  }

  BLE.setLocalName("CustomPeripheral");
  BLE.setAdvertisedService(customService);
  customService.addCharacteristic(customCharacteristic);
  BLE.addService(customService);
  customCharacteristic.setValue("Hello BLE");

  BLE.advertise();
  Serial.println("BLE Peripheral - Advertise started");
}

void loop() {
  // Do nothing
}
...
```

#### 4.2.5.4 Uji Coba

Uji coba sketsa di atas pada ESP32 Anda. Pastikan perangkat

Anda terdeteksi oleh aplikasi ponsel atau perangkat BLE lainnya dan data yang dikirim dapat diterima dengan benar.

#### 4.2.5.5 Hal yang patut diingat

- Pastikan untuk memahami cara kerja BLE, termasuk konsep peripheral dan central, serta karakteristik dan servis BLE.
- Pastikan untuk memahami dokumentasi library yang digunakan dan gunakan contoh kode sebagai referensi.
- Pastikan perangkat Anda mendukung BLE dan memiliki firmware yang diperbarui jika diperlukan.

### 4.3 Rangkuman

Pemilihan protokol komunikasi dalam IoT sangat bergantung pada kebutuhan spesifik aplikasi, termasuk kebutuhan daya, jangkauan, kecepatan data, dan kompleksitas implementasi. Protokol seperti MQTT dan HTTP sering digunakan untuk aplikasi umum, sementara protokol seperti LoRaWAN dan BLE digunakan untuk aplikasi dengan kebutuhan daya rendah atau jangkauan khusus. Memahami karakteristik dan aplikasi dari berbagai protokol dapat membantu dalam merancang sistem IoT yang efektif dan efisien.

### 4.4 Latihan Soal

Tentu, berikut beberapa latihan soal terkait implementasi protokol dalam Internet of Things (IoT):

1. Protokol MQTT: Jelaskan peran MQTT dalam komunikasi antara perangkat dalam Internet of Things. Bagaimana cara kerja MQTT dan apa keuntungannya dibandingkan dengan protokol komunikasi lainnya?
2. Protokol CoAP: Apa itu CoAP (Constrained Application Protocol) dan bagaimana cara kerjanya dalam konteks Internet of Things? Berikan contoh implementasi CoAP dalam sebuah proyek IoT.
3. Protokol HTTP vs MQTT: Bandingkan antara protokol HTTP dan MQTT dalam konteks penggunaan dalam Internet of Things. Kapan Anda akan memilih HTTP dan kapan Anda akan memilih MQTT?
4. Keamanan Protokol IoT: Apa tantangan keamanan yang dihadapi dalam implementasi protokol IoT seperti MQTT atau CoAP?

Jelaskan langkah-langkah yang dapat diambil untuk meningkatkan keamanan komunikasi dalam IoT.

#### 4.5 Bahan Diskusi

1. Integrasi Protokol: Bagaimana Anda akan mengintegrasikan protokol MQTT dengan protokol CoAP dalam sebuah proyek IoT? Apa keuntungan dan kerugian dari pendekatan ini?
2. Protokol Nirkabel dalam IoT: Diskusikan berbagai protokol nirkabel yang digunakan dalam IoT, seperti Wi-Fi, Bluetooth, LoRa, dan Zigbee. Kapan Anda akan menggunakan masing-masing protokol ini dalam implementasi proyek IoT?

#### 4.6 Rujukan

- Annisa, S. (2021, November 15). *Apa itu IoT? Cara Kerja, Tujuan dan Manfaatnya di Beberapa Industri*. Retrieved from Niaga Host: <https://www.niagahoster.co.id/blog/iot-adalah/>
- Chusna, F. (2023, Mei 10). <https://www.linknet.id>. Retrieved from contoh-iot-dalam-bidang-pertanian: <https://www.linknet.id/article/contoh-iot-dalam-bidang-pertanian>
- Manager, T. C. (2018, Oktober 3). <https://www.topcropmanager.com>. Retrieved from New field monitoring system from Hoskins Scientific: <https://www.topcropmanager.com/new-field-monitoring-system-from-hoskins-scientific-21485/>
- Syafiqoh, U., Sunardi, S., & Yudhana, A. (2018). Pengembangan Wireless Sensor Network Berbasis Internet of Things untuk Sistem Pemantauan Kualitas Air dan Tanah Pertanian. *Jurnal Informasi dan Pengembangan IT*, vol. 3, no. 2, 285-289.
- Yulianti, S. (2022, Februari 23). <https://www.indoniaga.co.id>. Retrieved from pertanian-green-house-berbasis-iot: <https://www.indoniaga.co.id/2022/02/pertanian-green-house-berbasis-iot.html>

## **BAB 5. PERANCANGAN DAN PEMROGRAMAN DATABASE**

### **Capaian Pembelajaran Matakuliah (CPMK)**

Mahasiswa mampu membuat server web dan database untuk menyimpan data dari perangkat embedded dan konfigurasi user client

Mahasiswa mampu membuat server web dan database untuk menyimpan data dari perangkat embedded dan konfigurasi user client

### **5.1 Pendahuluan**

Pemrograman database pada sistem embedded sering kali memerlukan pendekatan yang berbeda dibandingkan dengan pengembangan perangkat lunak pada komputer konvensional. Berikut adalah beberapa dasar-dasar pemrograman database pada sistem embedded:

#### **5.1.1 Pemilihan Database yang Sesuai**

Pertama, Anda perlu memilih database yang sesuai dengan kebutuhan dan keterbatasan sistem embedded. Beberapa database yang umum digunakan dalam sistem embedded termasuk SQLite, Berkeley DB, dan LevelDB. Pilihlah database yang ringan, memiliki konsumsi sumber daya yang rendah, dan kompatibel dengan platform hardware Anda.

#### **5.1.2 Integrasi Database dengan Sistem Embedded**

Setelah memilih database yang sesuai, Anda perlu mengintegrasikannya dengan sistem embedded Anda. Ini melibatkan instalasi database di perangkat target dan mengonfigurasi pengaturan yang diperlukan. Pastikan bahwa database dapat berjalan secara efisien pada perangkat embedded dengan sumber daya terbatas.

#### **5.1.3 Interaksi dengan Database**

Anda dapat berinteraksi dengan database pada sistem embedded melalui API atau antarmuka yang disediakan oleh database tersebut.

Ini melibatkan operasi dasar seperti penambahan, penghapusan, pembaruan, dan pencarian data. Pastikan untuk memahami dokumentasi database dan API yang digunakan untuk mengakses dan memanipulasi data.

#### 5.1.4 Perlakuan Terhadap Keterbatasan Hardware

Sistem embedded sering memiliki keterbatasan dalam hal daya komputasi, memori, dan penyimpanan. Oleh karena itu, saat memprogram database, pertimbangkan keterbatasan-keterbatasan ini dan upayakan untuk mengoptimalkan penggunaan sumber daya. Contohnya, hindari melakukan operasi database yang terlalu kompleks atau memakan waktu.

#### 5.1.5 Manajemen Kesalahan

Karena keterbatasan sumber daya, sistem embedded rentan terhadap kegagalan atau kesalahan. Oleh karena itu, pastikan untuk menangani kesalahan dengan baik dalam kode Anda. Pertimbangkan untuk mengimplementasikan mekanisme pemulihan kesalahan, logging, dan pemantauan untuk mengelola kesalahan yang mungkin terjadi saat berinteraksi dengan database.

#### 5.1.6 Keamanan

Terakhir, pastikan untuk mempertimbangkan keamanan data saat mengembangkan aplikasi dengan database di sistem embedded. Lakukan praktik-praktik keamanan yang umum seperti enkripsi data, penggunaan parameterized queries untuk mencegah SQL injection, dan pengaturan izin akses yang tepat.

Dengan memahami dan menerapkan dasar-dasar pemrograman database pada sistem embedded, Anda dapat mengembangkan aplikasi yang efisien dan andal dengan kemampuan penyimpanan dan pengambilan data yang diperlukan.

## 5.2 Pemrograman Database dalam IOT

Pada dasarnya, ESP32 sendiri tidak memiliki kemampuan untuk menjalankan database seperti MySQL atau SQLite seperti yang dapat dilakukan oleh server komputer yang lebih kuat. Namun, Anda dapat menggunakan ESP32 untuk berkomunikasi dengan server database eksternal yang dapat diakses melalui jaringan (Liska, 2021).

Berikut adalah langkah-langkah umum untuk memprogram ESP32 untuk berkomunikasi dengan server database eksternal:

1. Menghubungkan dengan Server Database Eksternal

Pastikan ESP32 terhubung ke jaringan yang sama dengan server database eksternal. Ini dapat dilakukan melalui koneksi Wi-Fi atau Ethernet, tergantung pada kemampuan ESP32 Anda.

2. Memilih Protokol Komunikasi

Pilih protokol komunikasi yang sesuai untuk berkomunikasi dengan server database. Umumnya, Anda akan menggunakan protokol seperti HTTP, MQTT, atau TCP/IP, tergantung pada jenis database yang Anda gunakan dan preferensi Anda.

3. Implementasikan Kode Komunikasi

Implementasikan kode di ESP32 untuk berkomunikasi dengan server database eksternal menggunakan protokol yang telah Anda pilih. Ini melibatkan membuat permintaan (request) ke server untuk menyimpan atau mengambil data, serta memproses respons (response) dari server.

Contoh (menggunakan HTTP dengan ESP32):

Kode program

...

```
#include <WiFi.h>
```

```
#include <HTTPClient.h>
```

```
const char* ssid = "YourSSID";
```

```
const char* password = "YourPassword";
```

```
const char* serverUrl = "http://your-database-server.com/api/data";
```

```
void setup() {
```

```
  Serial.begin(115200);
```

```
  WiFi.begin(ssid, password);
```

```
  while (WiFi.status() != WL_CONNECTED) {
```

```
    delay(1000);
```

```
    Serial.println("Connecting to WiFi...");
```

```
  }
```

```
  Serial.println("Connected to WiFi");
```

```
  // Example HTTP request
```

```
  HTTPClient http;
```

```
  http.begin(serverUrl);
```

```
  int httpCode = http.GET();
```



```

if (httpCode > 0) {
    String payload = http.getString();
    Serial.println(payload);
}
http.end();
}

void loop() {
    // Your code here
}
...

```

#### 4. Kelola Koneksi dan Kesalahan

Pastikan untuk mengelola koneksi Wi-Fi dan koneksi ke server dengan benar. Tangani kesalahan yang mungkin terjadi selama proses komunikasi, seperti kehilangan koneksi atau respons yang tidak valid.

#### 5. Keamanan

Penting untuk memastikan bahwa komunikasi antara ESP32 dan server database aman. Gunakan protokol yang aman seperti HTTPS jika memungkinkan, serta otentikasi dan enkripsi data untuk melindungi informasi sensitif.

#### 6. Uji dan Validasi

Uji kode Anda secara menyeluruh untuk memastikan bahwa ESP32 dapat berkomunikasi dengan server database eksternal dengan benar. Periksa apakah data dapat disimpan dan diambil dengan sukses.

Dengan mengikuti langkah-langkah di atas, Anda dapat memprogram ESP32 untuk berkomunikasi dengan server database eksternal dan menggunakan layanan database yang dikelola dari perangkat Anda.

### 5.3 Pemrograman Database SQL menggunakan protokol HTTP

Untuk membuat database SQL dengan HTTP pada IoT, Anda akan memerlukan server yang dapat menerima permintaan HTTP dari perangkat IoT dan menyimpan data ke dalam database SQL. Berikut langkah-langkah umumnya:

#### 1. Tentukan Kebutuhan Database

Pertama, tentukan kebutuhan Anda untuk database SQL, termasuk struktur tabel, jenis data, dan skema yang akan digunakan.

## 2. Persiapkan Server HTTP

Siapkan server HTTP yang dapat menerima permintaan dari perangkat IoT. Anda dapat menggunakan bahasa pemrograman seperti Python dengan bantuan framework seperti Flask atau Django, atau bahasa lainnya seperti Node.js dengan Express.js.

## 3. Buat Endpoints untuk Menerima Data

Buatlah endpoints pada server HTTP Anda yang akan digunakan oleh perangkat IoT untuk mengirimkan data. Misalnya, Anda dapat membuat endpoint `/sensor` yang akan menerima data dari sensor.

## 4. Terima Data dan Simpan ke Database

Implementasikan logika pada endpoint untuk menerima data yang dikirimkan oleh perangkat IoT dan menyimpannya ke dalam database SQL. Gunakan koneksi database dan bahasa SQL untuk menyimpan data dengan benar.

Contoh (menggunakan Python dan Flask):

```
```python
from flask import Flask, request
import sqlite3

app = Flask(__name__)

@app.route('/sensor', methods=['POST'])
def add_sensor_data():
    data = request.get_json()
    # Menghubungkan ke database
    conn = sqlite3.connect('sensor_data.db')
    c = conn.cursor()
    # Menyimpan data ke dalam database
    c.execute("INSERT INTO sensor_data (sensor_name, value)
VALUES (?, ?)", (data['sensor_name'], data['value']))
    conn.commit()
    conn.close()
    return 'Data berhasil disimpan'

if __name__ == '__main__':
    app.run(debug=True)
```
```

## 5. Uji Endpoint

Uji endpoint yang telah Anda buat untuk memastikan bahwa mereka dapat menerima data dari perangkat IoT dan menyimpannya ke dalam database SQL dengan benar.

#### 6. Keamanan

Pastikan untuk menerapkan langkah-langkah keamanan yang sesuai, termasuk validasi data yang diterima dan penggunaan parameterized queries untuk mencegah serangan SQL injection.

Dengan mengikuti langkah-langkah di atas, Anda dapat membuat database SQL dengan HTTP pada IoT, memungkinkan perangkat IoT untuk mengirimkan data ke server yang akan menyimpan data tersebut dalam database SQL.

### 5.4 Pemrograman Database menggunakan Broker Blynk

Untuk membuat database menggunakan Blynk dalam konteks IoT, Anda perlu mengintegrasikan perangkat IoT yang menjalankan Blynk dengan server yang menyimpan data ke database. Blynk adalah platform yang memungkinkan Anda membuat aplikasi untuk mengontrol perangkat IoT melalui internet (Syukhro & Rahmadewi, 2021). Langkah-Langkah Umum

1. Siapkan Perangkat IoT dengan Blynk: Program perangkat IoT (seperti ESP32 atau Arduino) untuk mengirim data ke aplikasi Blynk.
2. Siapkan Server HTTP dengan Database: Buat server yang dapat menerima permintaan HTTP dari perangkat IoT dan menyimpan data ke database SQL.
3. Integrasi dengan Blynk: Kirim data dari Blynk ke server melalui webhook atau endpoint HTTP.
4. Simpan Data ke Database: Implementasikan logika server untuk menyimpan data yang diterima ke database SQL.

Penjelasan lengkap pembuatan program Perangkat IoT dengan Blynk untuk ESP32 yang mengirimkan data suhu dan kelembaban ke Blynk:

1. Persiapan penulisan kode program

```
Kode program
...
#define BLYNK_PRINT Serial
```

```

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>

char auth[] = "YourAuthToken";
char ssid[] = "YourSSID";
char pass[] = "YourPassword";

#define DHTPIN 4
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;

void sendSensorData() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  Blynk.virtualWrite(V5, t);
  Blynk.virtualWrite(V6, h);
}

void setup() {
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  dht.begin();

  timer.setInterval(2000L, sendSensorData);
}

void loop() {
  Blynk.run();
  timer.run();
}

```

...

## 2. Siapkan Server HTTP dengan Database

Buat server menggunakan Python dan Flask, serta SQLite sebagai database.

Buat Database dan Tabel:

```
```python
import sqlite3

conn = sqlite3.connect('sensor_data.db')
c = conn.cursor()
c.execute("""
    CREATE TABLE sensor_data
    (id INTEGER PRIMARY KEY AUTOINCREMENT,
    temperature REAL,
    humidity REAL,
    timestamp          DATETIME          DEFAULT
CURRENT_TIMESTAMP)
""")
conn.commit()
conn.close()
```
```

Server Flask:

```
```python
from flask import Flask, request, jsonify
import sqlite3

app = Flask(__name__)

@app.route('/add_data', methods=['POST'])
def add_data():
    data = request.json
    temperature = data.get('temperature')
    humidity = data.get('humidity')

    conn = sqlite3.connect('sensor_data.db')
    c = conn.cursor()
    c.execute("INSERT INTO sensor_data (temperature,
humidity) VALUES (?, ?)", (temperature, humidity))
```
```

```

conn.commit()
conn.close()

return jsonify({'status': 'success'}), 201

if __name__ == '__main__':
    app.run(debug=True)

```

### 3. Integrasi dengan Blynk

Gunakan Blynk Webhook widget untuk mengirim data ke server Flask setiap kali ada perubahan data.

Konfigurasi Webhook Widget:

- URL: `http://<your\_server\_ip>:5000/add\_data`
- Method: POST
- Content Type: application/json
- Body: `{ "temperature": "/pin[5]/", "humidity": "/pin[6]/" }`

### 4. Uji Sistem

1. Jalankan server Flask.
2. Pastikan perangkat IoT mengirim data ke Blynk.
3. Periksa apakah Webhook mengirimkan data ke server Flask dan data tersebut disimpan dalam database SQLite.

### 5. Keamanan

- Gunakan HTTPS untuk mengamankan komunikasi antara perangkat IoT, Blynk, dan server.
- Validasi input di server untuk mencegah serangan seperti SQL Injection.
- Otentikasi dan otorisasi endpoint server untuk memastikan hanya perangkat yang sah yang dapat mengirim data.

Dengan langkah-langkah ini, Anda dapat membuat sistem yang mengirim data dari perangkat IoT ke database SQL menggunakan Blynk.

## 5.5 Pemrograman Database menggunakan Broker MQTT

Pemrograman database menggunakan broker MQTT dalam konteks IoT melibatkan beberapa komponen utama: perangkat IoT

yang mengirimkan data ke broker MQTT, broker MQTT itu sendiri, dan server aplikasi yang berlangganan ke topik MQTT dan menyimpan data ke dalam database. Berikut adalah langkah-langkah terperinci untuk mencapai ini:

### 1. Siapkan Broker MQTT

Anda perlu memiliki broker MQTT untuk menangani komunikasi pesan. Beberapa opsi populer termasuk Mosquitto, HiveMQ, dan EMQX. Anda dapat menginstalnya secara lokal atau menggunakan layanan broker MQTT yang tersedia secara online.

### 2. Program Perangkat IoT untuk Mengirim Data ke Broker MQTT

Program perangkat IoT Anda untuk mengirim data ke broker MQTT. Di sini, kita akan menggunakan ESP32 sebagai contoh perangkat IoT.

Kode Contoh untuk ESP32

```
```cpp
#include <WiFi.h>
#include <PubSubClient.h>

// Konfigurasi WiFi
const char ssid = "YourSSID";
const char password = "YourPassword";

// Konfigurasi MQTT
const char mqttServer = "your-mqtt-broker.com";
const int mqttPort = 1883;
const char mqttUser = "your-mqtt-username";
const char mqttPassword = "your-mqtt-password";

WiFiClient espClient;
PubSubClient client(espClient);

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
```

```

}
Serial.println("Connected to WiFi");

client.setServer(mqttServer, mqttPort);

while (!client.connected()) {
    Serial.println("Connecting to MQTT...");
    if (client.connect("ESP32Client", mqttUser, mqttPassword)) {
        Serial.println("Connected to MQTT");
    } else {
        Serial.print("Failed with state ");
        Serial.print(client.state());
        delay(2000);
    }
}

// Contoh pengiriman data suhu dan kelembaban
float temperature = 24.5;
float humidity = 60.0;

String payload = "{\"temperature\":" + String(temperature) +
",\"humidity\":" + String(humidity) + "}";
client.publish("sensor/data", payload.c_str());
}

void loop() {
    // Anda dapat mengirim data berkala atau berdasarkan kondisi
    // tertentu di sini
    client.loop();
}
...

```

### 3. Siapkan Server Aplikasi untuk Berlangganan ke Broker MQTT dan Menyimpan Data ke Database

Buat server aplikasi yang berlangganan ke topik MQTT dan menyimpan data yang diterima ke dalam database SQL. Kita akan menggunakan Python dengan Paho MQTT client dan SQLite untuk contoh ini.

Instalasi Pustaka yang Dibutuhkan

```
```sh
```

```
pip install paho-mqtt sqlite3
```



```

...

#Kode Server Aplikasi
```python
import paho.mqtt.client as mqtt
import sqlite3
import json

# Konfigurasi MQTT
mqtt_broker = "your-mqtt-broker.com"
mqtt_port = 1883
mqtt_user = "your-mqtt-username"
mqtt_password = "your-mqtt-password"
mqtt_topic = "sensor/data"

# Konfigurasi Database
db_file = 'sensor_data.db'

# Callback ketika terhubung ke broker MQTT
def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))
    client.subscribe(mqtt_topic)

# Callback ketika pesan diterima dari broker MQTT
def on_message(client, userdata, msg):
    print("Message received -> " + msg.topic + " " + " " +
str(msg.payload))
    data = json.loads(msg.payload)
    save_to_db(data['temperature'], data['humidity'])

# Fungsi untuk menyimpan data ke dalam database
def save_to_db(temperature, humidity):
    conn = sqlite3.connect(db_file)
    c = conn.cursor()
    c.execute("INSERT INTO sensor_data (temperature,
humidity) VALUES (?, ?)", (temperature, humidity))
    conn.commit()
    conn.close()

# Membuat tabel jika belum ada
def create_db():

```

```

conn = sqlite3.connect(db_file)
c = conn.cursor()
c.execute("""
    CREATE TABLE IF NOT EXISTS sensor_data
    (id INTEGER PRIMARY KEY AUTOINCREMENT,
    temperature REAL,
    humidity REAL,
    timestamp          DATETIME          DEFAULT
CURRENT_TIMESTAMP)
    """)
conn.commit()
conn.close()

# Inisialisasi MQTT Client
client = mqtt.Client()
client.username_pw_set(mqtt_user, mqtt_password)
client.on_connect = on_connect
client.on_message = on_message

create_db()
client.connect(mqtt_broker, mqtt_port, 60)

# Looping MQTT client
client.loop_forever()
```

```

#### 4. Uji Sistem Anda

1. Jalankan broker MQTT: Pastikan broker MQTT Anda berjalan dan dapat menerima pesan.
2. Jalankan perangkat IoT: Pastikan perangkat IoT Anda mengirimkan data ke broker MQTT.
3. Jalankan server aplikasi: Pastikan server aplikasi Anda terhubung ke broker MQTT dan menerima serta menyimpan data ke dalam database.

#### 5. Keamanan

Pastikan komunikasi antara perangkat IoT, broker MQTT, dan server aplikasi aman. Gunakan protokol TLS/SSL untuk enkripsi data dan implementasikan autentikasi yang tepat untuk broker MQTT.

Dengan mengikuti langkah-langkah di atas, Anda dapat

mengatur perangkat IoT yang mengirim data ke broker MQTT, kemudian menerima data tersebut di server aplikasi dan menyimpannya dalam database SQL.

## 5.6 Rangkuman

Pemrograman database pada Internet of Things (IoT) adalah topik yang penting karena melibatkan pengelolaan dan penyimpanan data yang dihasilkan oleh perangkat IoT. Berikut adalah rangkuman dari konsep utama terkait pemrograman database dalam konteks IoT:

1. Pentingnya Database dalam IoT
2. Jenis Database yang Digunakan
  - SQL (Relational Databases): Misalnya MySQL, PostgreSQL. Cocok untuk data yang terstruktur dan memerlukan integritas tinggi.
  - NoSQL (Non-relational Databases): Misalnya MongoDB, Cassandra. Cocok untuk data yang tidak terstruktur atau semi-terstruktur dan skala besar.
  - Time-series Databases: Misalnya InfluxDB, TimescaleDB. Dirancang khusus untuk data yang dihasilkan berdasarkan waktu seperti data sensor IoT.
3. Arsitektur Database pada IoT
4. Integrasi dan Sinkronisasi Data
5. Keamanan dan Privasi Data
6. Optimasi dan Skalabilitas
7. Analitik dan Visualisasi Data
8. Teknologi dan Alat Terkait
  - Database Management Systems (DBMS): Seperti MySQL, MongoDB, InfluxDB.
  - IoT Platforms: Seperti AWS IoT, Google Cloud IoT, Azure IoT Hub.
  - Middleware: Perangkat lunak yang menghubungkan perangkat IoT dengan database dan aplikasi.

Dengan memahami konsep-konsep di atas, pengembang dapat merancang dan mengimplementasikan solusi database yang efisien dan efektif untuk aplikasi IoT.

## 5.7 Latihan Soal

Berikut adalah beberapa latihan soal terkait pemrograman

database dalam konteks IoT. Latihan-latihan ini mencakup berbagai aspek seperti desain database, integrasi data, dan pengolahan data real-time.

## 1. Desain Database

Soal 1: Buatlah skema database relasional untuk sebuah sistem smart home yang mencakup tabel untuk perangkat (devices), pembacaan sensor (sensor readings), dan pengguna (users). Pastikan untuk menyertakan kunci primer dan hubungan antar tabel.

Jawaban:

```
sql
Salin kode
CREATE TABLE users (
    user_id INT PRIMARY KEY,
    username VARCHAR(50),
    email VARCHAR(100)
);

CREATE TABLE devices (
    device_id INT PRIMARY KEY,
    user_id INT,
    device_name VARCHAR(50),
    device_type VARCHAR(50),
    FOREIGN KEY (user_id) REFERENCES
users(user_id)
);

CREATE TABLE sensor_readings (
    reading_id INT PRIMARY KEY,
    device_id INT,
    reading_value FLOAT,
    timestamp DATETIME,
    FOREIGN KEY (device_id) REFERENCES
devices(device_id)
);
```

## 2. Pengambilan Data

Soal 2: Tuliskan query SQL untuk mengambil semua pembacaan sensor dari perangkat tertentu dalam satu hari terakhir.

Jawaban:

```
sql
Salin kode
SELECT * FROM sensor_readings
WHERE device_id = [ID_PERANGKAT]
AND timestamp >= NOW() - INTERVAL 1 DAY;
```

## 3. Integrasi dan Sinkronisasi Data

Soal 3: Jelaskan bagaimana Anda akan menggunakan MQTT untuk mengirim data sensor dari perangkat IoT ke server database.

Jawaban:

- Konfigurasi Broker MQTT: Siapkan broker MQTT (misalnya, Mosquitto) untuk menerima pesan dari perangkat IoT.
- Publish Data dari Perangkat: Perangkat IoT mengirim data sensor dengan menggunakan topik tertentu.

```
python
Salin kode
import paho.mqtt.client as mqtt

client = mqtt.Client()
client.connect("mqtt.broker.address", 1883,
60)

topic = "home/sensors/temperature"
payload = '{"device_id": 1,
"reading_value": 23.5, "timestamp": "2024-
06-07T12:00:00Z"}'
client.publish(topic, payload)
```

- **Subscribe dan Simpan Data di Server:** Server database berlangganan ke topik tersebut dan menyimpan data yang diterima ke dalam database.

```
python
Salin kode
import paho.mqtt.client as mqtt
import json
import mysql.connector

def on_message(client, userdata, msg):
    data = json.loads(msg.payload)
    connection = mysql.connector.connect(
        host="localhost",
        user="root",
        password="password",
        database="iot_db"
    )
    cursor = connection.cursor()
    sql = "INSERT INTO sensor_readings
(device_id, reading_value, timestamp)
VALUES (%s, %s, %s)"
    values = (data['device_id'],
data['reading_value'], data['timestamp'])
    cursor.execute(sql, values)
    connection.commit()
    cursor.close()
    connection.close()

client = mqtt.Client()
client.connect("mqtt.broker.address", 1883,
60)
client.subscribe("home/sensors/temperature"
)
client.on_message = on_message
client.loop_forever()
```

## 5.8 Bahan Diskusi

Diskusi mengenai pemrograman basis data dalam konteks

Internet of Things (IoT) dapat melibatkan berbagai topik yang mencakup aspek teknis, keamanan, serta tantangan dan peluang dalam implementasinya. Berikut adalah beberapa topik yang bisa digunakan sebagai bahan diskusi:

#### Topik Diskusi

##### 1. Pemilihan Basis Data untuk IoT

Kapan menggunakan SQL vs. NoSQL: Diskusikan skenario di mana penggunaan database relasional lebih cocok dibandingkan dengan database non-relasional, dan sebaliknya.

Time-series Database: Keuntungan menggunakan database time-series seperti InfluxDB atau TimescaleDB untuk data IoT.

##### 2. Arsitektur Sistem Basis Data untuk IoT

Edge vs. Cloud Computing: Manfaat dan tantangan dalam memproses data di edge dibandingkan dengan cloud.

Fog Computing: Bagaimana arsitektur fog computing dapat membantu dalam pengelolaan data IoT.

## 5.9 Daftar Rujukan

- Liska, B. H. (2021). Implementasi IoT Data Storage Dengan Menggunakan Sistem Basis Data Terdistribusi Berbasis MySQL Cluster. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, . 2986-2993 .
- Syafiqoh, U., Sunardi, S., & Yudhana, A. (2018). Pengembangan Wireless Sensor Network Berbasis Internet of Things untuk Sistem Pemantauan Kualitas Air dan Tanah Pertanian. *Jurnal Informasi dan Pengembangan IT*, vol. 3, no. 2, 285-289.
- Syukhro, I., & Rahmadewi, R. (2021). Penggunaan Aplikasi Blynk Untuk Monitoring dan Kontrol Jarak Jauh pada Sistem Kompos Pintar. *ELECTRICIAN – Jurnal Rekayasa dan Teknologi Elektro*, 1-11.

## **BAB 6. INTEGRASI SISTEM IOT DAN SISTEM INFORMASI**

### **Capaian Pembelajaran Matakuliah (CPMK)**

Mahasiswa mampu membuat server web dan database untuk menyimpan data dari perangkat embedded dan konfigurasi user client

### **6.1 Pendahuluan**

Pengintegrasian sistem IoT (Internet of Things) dan sistem informasi melibatkan berbagai langkah dan teknologi untuk memastikan data yang dihasilkan oleh perangkat IoT dapat digunakan secara efektif oleh sistem informasi yang ada. Berikut adalah penjelasan detail mengenai proses, teknologi, tantangan, dan contoh kasus dalam pengintegrasian tersebut:

Langkah-langkah Integrasi Sistem IoT dan Sistem Informasi

#### **1. Identifikasi Kebutuhan Bisnis**

Tujuan Integrasi: Tentukan apa yang ingin dicapai melalui integrasi ini, seperti peningkatan efisiensi operasional, penghematan biaya, atau peningkatan layanan pelanggan.

Data yang Dibutuhkan: Identifikasi jenis data yang diperlukan dan dari perangkat IoT mana data tersebut akan dikumpulkan.

#### **2. Pemilihan Teknologi dan Protokol**

Protokol Komunikasi: Pilih protokol yang sesuai untuk mengirim data dari perangkat IoT ke sistem informasi. Beberapa protokol umum adalah:

MQTT (Message Queuing Telemetry Transport): Protokol ringan yang cocok untuk jaringan dengan bandwidth rendah.

CoAP (Constrained Application Protocol): Protokol yang dirancang untuk perangkat dengan sumber daya terbatas.

HTTP/HTTPS: Protokol standar untuk komunikasi web yang juga digunakan dalam beberapa aplikasi IoT.

Middleware: Gunakan middleware untuk menghubungkan perangkat IoT dengan sistem informasi, memungkinkan pengumpulan, pemrosesan, dan distribusi data secara efisien.

#### **3. Desain Arsitektur Sistem**



Edge Computing: Mengolah data di dekat sumber (perangkat IoT) untuk mengurangi latensi dan beban jaringan.

Fog Computing: Memproses data di node intermediary antara perangkat IoT dan cloud untuk optimasi.

Cloud Computing: Menggunakan layanan cloud untuk penyimpanan dan analisis data skala besar.

#### 4. Pengembangan dan Pengujian

Data Ingestion: Implementasikan proses pengumpulan data dari perangkat IoT ke sistem informasi.

Normalisasi Data: Mengubah data mentah menjadi format yang lebih terstruktur dan mudah dianalisis.

Pengujian Sistem: Lakukan pengujian untuk memastikan data dari perangkat IoT dapat ditangani dengan benar oleh sistem informasi.

#### 5. Implementasi dan Monitoring

Deploy Sistem: Terapkan solusi integrasi pada lingkungan produksi.

Monitoring dan Pemeliharaan: Pantau performa sistem secara kontinu untuk memastikan operasi yang optimal dan lakukan pemeliharaan berkala.

#### 6. Teknologi dan **Protokol** untuk Integrasi

Protokol MQTT: Sering digunakan karena efisiensinya dalam mengirim data dalam jumlah kecil dengan latensi rendah.

Protokol CoAP: Ideal untuk perangkat IoT yang memiliki sumber daya terbatas dan membutuhkan komunikasi efisien.

Middleware: Solusi middleware seperti Apache Kafka, RabbitMQ, dan Node-RED sering digunakan untuk mengelola data yang masuk dari perangkat IoT ke sistem informasi.

#### 7. Tantangan dan Solusi

##### - Skalabilitas

Tantangan: Menangani jumlah perangkat dan volume data yang terus meningkat.

Solusi: Menggunakan arsitektur berbasis cloud dan teknologi seperti microservices untuk memastikan sistem dapat dengan mudah ditingkatkan.

##### - Interoperabilitas

Tantangan: Berbagai perangkat IoT menggunakan protokol dan format data yang berbeda.

Solusi: Menggunakan standar terbuka dan middleware yang dapat mengintegrasikan berbagai sumber data.

##### - Keamanan

Tantangan: Melindungi data dari ancaman siber.

Solusi: Implementasi enkripsi, otentikasi yang kuat, dan pemantauan keamanan yang berkelanjutan.

- Data Integrity

Tantangan: Memastikan data yang dikumpulkan akurat dan dapat diandalkan.

Solusi: Menggunakan mekanisme validasi data dan redundansi untuk menjaga integritas data.

## 8. Contoh Implementasi

- Smart Home

Deskripsi: Mengintegrasikan data dari perangkat rumah pintar seperti lampu, termostat, dan kamera keamanan dengan aplikasi manajemen rumah.

Manfaat: Otomatisasi kontrol rumah, peningkatan keamanan, dan penghematan energi.

- Industrial IoT

Deskripsi: Menghubungkan sensor mesin pabrik dengan sistem ERP (Enterprise Resource Planning) untuk pemantauan dan pemeliharaan prediktif.

Manfaat: Mengurangi downtime, meningkatkan efisiensi produksi, dan optimasi rantai pasokan.

- Healthcare IoT

Deskripsi: Integrasi perangkat pemantauan kesehatan seperti monitor jantung dan alat pengukur tekanan darah dengan sistem informasi kesehatan.

Manfaat: Pemantauan pasien secara real-time, respons medis yang lebih cepat, dan perawatan yang lebih personal.

Integrasi sistem IoT dan sistem informasi adalah langkah penting untuk mengoptimalkan penggunaan data dari perangkat IoT. Dengan memahami kebutuhan bisnis, memilih teknologi yang tepat, dan mengatasi tantangan seperti skalabilitas, interoperabilitas, dan keamanan, organisasi dapat meningkatkan efisiensi operasional, membuat keputusan yang lebih baik, dan menawarkan layanan yang lebih baik kepada pelanggan.

## 6.2 Sistem Monitoring Hidroponik Berbasis IoT Dengan Sensor Suhu, pH, dan Ketinggian Air Menggunakan ESP8266

Hidroponik adalah budidaya pertanian tanpa menggunakan media tanah, sehingga hidroponik merupakan aktivitas pertanian yang

dijalankan dengan menggunakan air sebagai medium untuk menggantikan tanah (Adiputra, Kristanto, Sayid Albana, & Wednestwo, 2022). Sistem hidroponik semakin populer di kalangan petani dan penyedia jasa pertanian karena mampu menghasilkan tanaman yang lebih sehat dan produktif tanpa menggunakan tanah sebagai media tanam. Dalam sistem hidroponik, tanaman ditanam dengan menggunakan larutan nutrisi yang mengandung unsur hara yang dibutuhkan tanaman untuk tumbuh dan berkembang. Namun, keberhasilan dalam budidaya hidroponik sangat bergantung pada pengaturan parameter lingkungan, seperti suhu, kelembaban dan pH, yang harus dipantau secara terus menerus dan diatur sesuai kebutuhan tanaman. Contohnya perubahan pH sangat mempengaruhi pertumbuhan, khususnya tanaman sayur. (Karim, 2021) Untuk mempermudah pemantauan dan pengaturan parameter lingkungan pada budidaya hidroponik, pengembangan teknologi IoT dapat dimanfaatkan. Dengan adanya sebuah sistem yang dapat melakukan monitoring tanaman akan menjadikan proses pengambilan data menjadi lebih akurat. IoT merupakan seperangkat alat elektronika disertai sensor atau gabungan beberapa sensor, program komputer dan perangkat digital yang saling terhubung satu sama lain dan berkomunikasi. Dengan menerapkan IoT, kita dapat membangun teknologi yang bekerja secara jarak jauh yang lebih praktis untuk dikontrol kapanpun. Dalam budidaya hidroponik, IoT dapat digunakan untuk memonitor kondisi lingkungan secara real-time dan memberikan peringatan jika terdapat perubahan kondisi lingkungan yang perlu diatasi. Salah satu teknologi IoT yang dapat digunakan dalam sistem monitoring hidroponik adalah modul ESP8266. Modul ini merupakan modul Wi-Fi yang memungkinkan perangkat elektronik untuk terhubung ke internet secara nirkabel. Guna mendapatkan hasil yang sesuai dilakukan beberapa tahapan.

#### 1. Tahap analisis kebutuhan

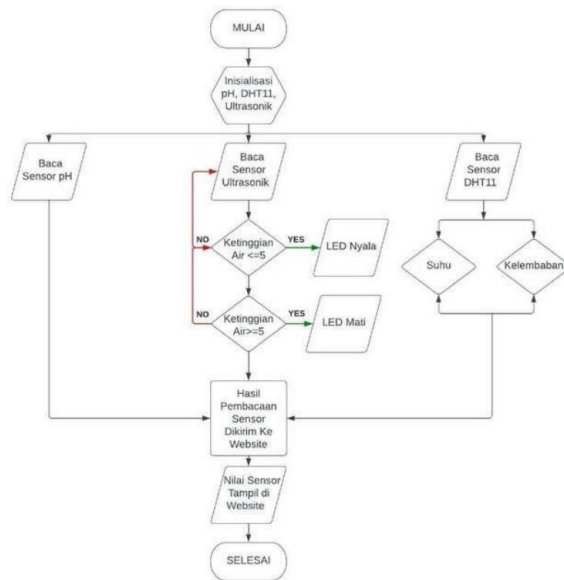
Pada tahap ini analisis dilakukan tahapan perencanaan, dalam melakukan perencanaan seperti penentuan konsep sistem yang akan dibuat dan tujuan dalam mengembangkan sistem monitoring hidroponik berbasis IoT. Tahap perencanaan adalah langkah awal dalam siklus pengembangan sistem yang akan menjadi acuan dalam pelaksanaan penelitian hingga akhir yang merupakan bagian dari analisis kebutuhan.

Berikutnya tahap observasi, dalam tahap ini dilakukan identifikasi masalah yang perlu diatasi dalam sistem monitoring hidroponik. Selain identifikasi juga studi tentang teknologi IoT dan

Modul ESP8266 sebagai perangkat IoT yang akan digunakan dalam sistem monitoring hidroponik. Modul ESP8266 sendiri merupakan salah satu mikrokontroler yang memungkinkan perangkat untuk terhubung dengan jaringan internet melalui WiFi. Sehingga data yang terbaca dari sensor akan diolah oleh modul ESP8266 dan dikirimkan menuju server/website melalui jaringan internet.

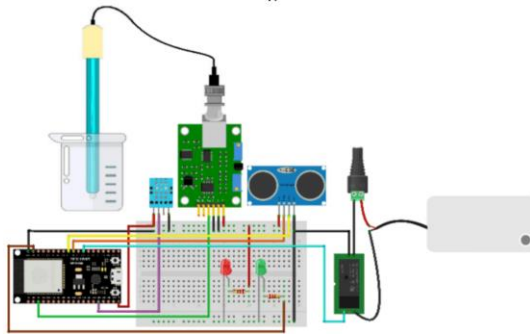
## 2. Tahap desain

Pada tahap desain, dirancang arsitektur sistem dan spesifikasi teknis yang diperlukan, seperti merancang sistem monitoring menggunakan sensor suhu, kelembaban, dan pH serta integrasi dengan ESP8266 sebagai perangkat IoT. Diagram alir dalam penelitian ini menggunakan diagram aliran sistem seperti yang terlihat pada gambar 6-1.



Gambar 6-1 Diagram alir sistem

Sedangkan desain sistem rangkaian elektronik dapat dilihat pada gambar 6-2.



Gambar 6-2 Rangkaian sistem

### 3. Implementasi

Pada tahap implementasi, mengimplementasikan sistem monitoring hidroponik berbasis IoT yang telah dirancang pada tahap sebelumnya dengan menentukan perangkat yang dibutuhkan menuliskan kode program kedalam sistem.

### Perangkat Keras yang Dibutuhkan

1. ESP8266 (misalnya, NodeMCU)
2. Sensor Suhu (misalnya, DHT11 atau DHT22)
3. Sensor pH
4. Sensor Ketinggian Air (misalnya, sensor ultrasonik HC-SR04)
5. Breadboard dan kabel jumper

### Perpustakaan yang Dibutuhkan

- DHT sensor library (`DHT.h`)
- Adafruit Unified Sensor (`Adafruit\_Sensor.h`)
- ThingSpeak (untuk mengirim data ke ThingSpeak) atau PubSubClient (untuk MQTT)

### Kode Program

Berikut ini adalah contoh kode program yang mengirimkan data ke ThingSpeak. Anda bisa menggantinya untuk menggunakan MQTT jika diperlukan.

```

...

#include <ESP8266WiFi.h>
#include <ThingSpeak.h>
#include <DHT.h>

// Pin Definitions
#define DHTPIN D4           // Pin for DHT sensor
#define DHTTYPE DHT11       // DHT 11
#define TRIGPIN D5          // Pin for HC-SR04 Trig

```

```

#define ECHOPIN D6          // Pin for HC-SR04 Echo
#define PHPIN A0           // Analog pin for pH
sensor

// WiFi credentials
const char* ssid = "your_SSID";
const char* password = "your_PASSWORD";

// ThingSpeak credentials
unsigned long myChannelNumber =
YOUR_CHANNEL_NUMBER;
const char * myWriteAPIKey =
"YOUR_WRITE_API_KEY";

WiFiClient client;
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(115200);
  dht.begin();

  // Initialize the HC-SR04 sensor
  pinMode(TRIGPIN, OUTPUT);
  pinMode(ECHOPIN, INPUT);

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("Connected to Wi-Fi");

  // Initialize ThingSpeak
  ThingSpeak.begin(client);
}

void loop() {
  // Read data from DHT sensor
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  // Read data from pH sensor
  int pHValue = analogRead(PHPIN);
  float voltage = pHValue * (3.3 / 1024.0);
  float pH = 3.5 * voltage; // Convert voltage

```

```

to pH value (calibration needed)

    // Read data from HC-SR04 sensor
    digitalWrite(TRIGPIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIGPIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIGPIN, LOW);
    long duration = pulseIn(ECHOPIN, HIGH);
    float distance = (duration * 0.034) / 2;    //
Convert to centimeters

    // Print data to serial monitor
    Serial.print("Temperature: ");
    Serial.print(temperature);
    Serial.print(" C, Humidity: ");
    Serial.print(humidity);
    Serial.print(" %, pH: ");
    Serial.print(pH);
    Serial.print(", Water Level: ");
    Serial.print(distance);
    Serial.println(" cm");

    // Send data to ThingSpeak
    ThingSpeak.setField(1, temperature);
    ThingSpeak.setField(2, humidity);
    ThingSpeak.setField(3, pH);
    ThingSpeak.setField(4, distance);
    ThingSpeak.writeFields(myChannelNumber,
myWriteAPIKey);

    // Delay before next reading
    delay(20000);    // ThingSpeak allows updates
every 15 seconds
}
...

```

### ### Penjelasan Kode

1. Mengimpor Perpustakaan: Perpustakaan untuk WiFi, ThingSpeak, dan DHT sensor diimpor.
2. Inisialisasi Pin: Pin untuk DHT sensor, HC-SR04 sensor, dan pH sensor diatur.
3. Kredensial WiFi dan ThingSpeak: Kredensial WiFi dan API key ThingSpeak diset.
4. Fungsi `setup`:
  - Inisialisasi komunikasi serial.

- Mulai sensor DHT dan atur pin HC-SR04.
  - Hubungkan ke WiFi.
  - Inisialisasi ThingSpeak.
5. Fungsi `loop`:
- Membaca data dari sensor DHT, pH, dan HC-SR04.
  - Mengirim data ke ThingSpeak.
  - Menunda pengulangan selama 20 detik.

#### ### Catatan

- Kalibrasi Sensor pH: Rumus untuk mengonversi nilai analog dari sensor pH mungkin perlu dikalibrasi sesuai dengan sensor yang digunakan.
- Keamanan WiFi: Jangan hardcode kredensial WiFi dan API key dalam kode saat digunakan dalam aplikasi nyata. Gunakan metode yang lebih aman untuk menyimpan kredensial.

Dengan kode ini, sistem hidroponik Anda akan memantau suhu, pH, dan ketinggian air secara terus-menerus dan mengirim data ke ThingSpeak untuk dianalisis lebih lanjut atau dilihat secara real-time.

#### 4. Pengujian

Tahap pengujian dilakukan untuk memastikan sistem yang dibangun berfungsi dengan baik dan memenuhi spesifikasi teknis yang telah ditentukan.

### 2.1 Uraian Materi

#### 2.2 1

.....  
 .....  
 .....  
 .....

#### 2.3 Uraian Materi 2 (dst disesuaikan dengan banyaknya uraian materi)

.....  
 .....  
 .....  
 .....

#### 2.4 Rangkuman



.....  
.....  
.....  
.....

## **2.5 Latihan Soal**

.....  
.....  
.....  
.....

## **2.6 Bahan Diskusi**

.....  
.....  
.....  
.....

## **2.7 Daftar Rujukan (Gunakan Reference Manager)**

.....  
.....  
.....  
.....

## **BAB 7. PERANCANGAN UI DAN PEMROGRAMAN WEB, DESKTOP, ATAU MOBILE**

### **Capaian Pembelajaran Matakuliah (CPMK)**

Mahasiswa mampu membuat antar muka untuk aplikasi desktop, mobile dan web untuk integrasi dengan sistem embedded

### **7.1 Pendahuluan**

Perancangan UI (*User Interface*) dan pemrograman untuk aplikasi IoT dapat dilakukan pada berbagai platform, seperti web, desktop, atau mobile. Setiap platform memiliki kelebihan dan kekurangan masing-masing, serta teknik dan alat yang berbeda untuk implementasinya. Berikut adalah panduan umum untuk perancangan UI dan pemrograman aplikasi IoT pada ketiga platform tersebut:

#### **### 1. Perancangan UI**

##### **#### Prinsip Dasar Perancangan UI untuk IoT**

- Kesederhanaan: Buat antarmuka yang sederhana dan intuitif, mudah dipahami oleh pengguna.
- Responsif: UI harus responsif dan menyesuaikan dengan berbagai ukuran layar.
- Real-time Data: Menampilkan data secara real-time untuk memberikan informasi terkini kepada pengguna.
- Interaktivitas: Memungkinkan pengguna untuk berinteraksi dengan perangkat IoT, seperti mengontrol perangkat atau menerima notifikasi.
- Visualisasi Data: Menggunakan grafik dan diagram untuk memvisualisasikan data dari perangkat IoT.

#### **### 2. Pemrograman Web**

##### **#### Teknologi yang Digunakan**

- Front-end: HTML, CSS, JavaScript, Frameworks (React, Angular, Vue.js)

- Back-end: Node.js, Express.js, Python (Django, Flask), PHP
- Database: MySQL, PostgreSQL, MongoDB
- Real-time Communication: WebSockets, MQTT

## 7.2 Pemrograman Front end dan Back end

Berikut merupakan pemrograman antarmuka front end dengan menggunakan javascript dan pemrograman backend.

### ##### Front-end (React)

```

```javascript
import React, { useState, useEffect } from 'react';
import io from 'socket.io-client';

const socket = io('http://your-server-address');

function App() {
  const [temperature, setTemperature] = useState(0);
  const [humidity, setHumidity] = useState(0);

  useEffect(() => {
    socket.on('sensorData', data => {
      setTemperature(data.temperature);
      setHumidity(data.humidity);
    });

    return () => socket.off('sensorData');
  }, []);

  return (
    <div className="App">
      <h1>Smart Farming Dashboard</h1>
      <p>Temperature: {temperature} °C</p>
      <p>Humidity: {humidity} %</p>
    </div>
  );
}

export default App;
```

```

### ##### Back-end (Node.js with Express)

```

```javascript
const express = require('express');
const http = require('http');
const socketIo = require('socket.io');
const mqtt = require('mqtt');

```

```

const app = express();
const server = http.createServer(app);
const io = socketIo(server);

const mqttClient = mqtt.connect('mqtt://broker.hivemq.com');

mqttClient.on('connect', () => {
  mqttClient.subscribe('sensorData');
});

mqttClient.on('message', (topic, message) => {
  const data = JSON.parse(message.toString());
  io.emit('sensorData', data);
});

server.listen(3000, () => {
  console.log('Server listening on port 3000');
});
``

```

## 7.3 Pemrograman Desktop

### #### Teknologi yang Digunakan

- Electron: Framework untuk membuat aplikasi desktop menggunakan HTML, CSS, dan JavaScript.
- Python with Tkinter: Untuk aplikasi desktop sederhana.

### #### Contoh Implementasi

#### ##### Menggunakan Electron

##### main.js

```

````javascript
const { app, BrowserWindow } = require('electron');

function createWindow () {
  const win = new BrowserWindow({
    width: 800,
    height: 600,
    webPreferences: {
      nodeIntegration: true
    }
  });

  win.loadFile('index.html');
}

```

```

app.whenReady().then(createWindow);

app.on('window-all-closed', () => {
  if (process.platform !== 'darwin') {
    app.quit();
  }
});

app.on('activate', () => {
  if (BrowserWindow.getAllWindows().length === 0) {
    createWindow();
  }
});
```

index.html
```html
<!DOCTYPE html>
<html>
<head>
  <title>Smart Farming Dashboard</title>
</head>
<body>
  <h1>Smart Farming Dashboard</h1>
  <div id="content"></div>
  <script src="renderer.js"></script>
</body>
</html>
```

renderer.js
```javascript
const io = require('socket.io-client');
const socket = io('http://your-server-address');

socket.on('sensorData', data => {
  document.getElementById('content').innerHTML = `
    <p>Temperature: ${data.temperature} °C</p>
    <p>Humidity: ${data.humidity} %</p>
  `;
});
```

```

## 7.4 Pemrograman Mobile

#### Teknologi yang Digunakan

- Native Mobile Development: Android (Java/Kotlin), iOS (Swift)
- Cross-platform Frameworks: Flutter, React Native

#### #### Contoh Implementasi

##### ##### Menggunakan React Native

```

```javascript
import React, { useEffect, useState } from 'react';
import { StyleSheet, Text, View } from 'react-native';
import io from 'socket.io-client';

const socket = io('http://your-server-address');

export default function App() {
  const [temperature, setTemperature] = useState(0);
  const [humidity, setHumidity] = useState(0);

  useEffect(() => {
    socket.on('sensorData', data => {
      setTemperature(data.temperature);
      setHumidity(data.humidity);
    });

    return () => socket.off('sensorData');
  }, []);

  return (
    <View style={styles.container}>
      <Text>Temperature: {temperature} °C</Text>
      <Text>Humidity: {humidity} %</Text>
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```

```

## 7.5 Rangkuman

Perancangan UI dan pemrograman untuk aplikasi IoT dapat dilakukan pada berbagai platform, tergantung kebutuhan pengguna. Dengan menggunakan teknologi yang sesuai dan merancang UI yang intuitif, responsif, dan interaktif, aplikasi IoT dapat memberikan manfaat maksimal kepada pengguna. Implementasi kode di atas memberikan gambaran dasar tentang bagaimana membangun aplikasi IoT pada platform web, desktop, dan mobile.

## 2.8 Latihan Soal

Berikut adalah beberapa latihan soal pemrograman front-end untuk aplikasi IoT di platform mobile, desktop, dan web. Latihan ini dirancang untuk membantu Anda menguasai keterampilan pemrograman front-end dan integrasi dengan perangkat IoT.

### Latihan Soal Pemrograman Front-end Mobile untuk IoT

#### Soal 1: Monitoring Suhu dengan React Native

Buat aplikasi mobile sederhana menggunakan React Native yang menampilkan data suhu dari sensor IoT yang terhubung melalui WebSocket.

##### Langkah-langkah:

1. Install dependencies: Install React Native dan library WebSocket.
2. Setup WebSocket connection: Buat koneksi WebSocket ke server yang menyediakan data suhu.
3. Display data: Tampilkan data suhu secara real-time di layar utama aplikasi.

##### Contoh Kode:

```
``javascript
import React, { useEffect, useState } from 'react';
import { StyleSheet, Text, View } from 'react-native';
import { WebSocket } from 'react-native-websocket';

const ws = new WebSocket('ws://your-server-address');

export default function App() {
  const [temperature, setTemperature] = useState(0);

  useEffect(() => {
```

```

ws.onmessage = (message) => {
  const data = JSON.parse(message.data);
  setTemperature(data.temperature);
};

return () => {
  ws.close();
};
}, []);

return (
  <View style={styles.container}>
    <Text>Temperature: {temperature} °C</Text>
  </View>
);
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
`);

```

### ### Latihan Soal Pemrograman Front-end Desktop untuk IoT

#### #### Soal 2: Kontrol Lampu dengan Electron

Buat aplikasi desktop menggunakan Electron yang memungkinkan pengguna untuk menyalakan dan mematikan lampu IoT melalui MQTT.

#### ##### Langkah-langkah:

1. Setup Electron: Buat proyek baru menggunakan Electron.
2. Integrasi MQTT: Tambahkan library MQTT dan buat koneksi ke broker MQTT.
3. Kontrol UI: Buat antarmuka pengguna dengan tombol untuk menyalakan dan mematikan lampu.

#### ##### Contoh Kode:

```

main.js
```javascript
const { app, BrowserWindow } = require('electron');
const path = require('path');

```



```

function createWindow () {
  const win = new BrowserWindow({
    width: 800,
    height: 600,
    webPreferences: {
      preload: path.join(__dirname, 'preload.js')
    }
  });

  win.loadFile('index.html');
}

app.whenReady().then(() => {
  createWindow();

  app.on('activate', function () {
    if (BrowserWindow.getAllWindows().length === 0)
      createWindow();
  });

  app.on('window-all-closed', function () {
    if (process.platform !== 'darwin') app.quit();
  });
});

preload.js
```javascript
const mqtt = require('mqtt');
const client =
mqtt.connect('mqtt://broker.hivemq.com');

window.addEventListener('DOMContentLoaded', () => {
  const onButton = document.getElementById('on');
  const offButton = document.getElementById('off');

  onButton.addEventListener('click', () => {
    client.publish('home/lamp', 'ON');
  });

  offButton.addEventListener('click', () => {
    client.publish('home/lamp', 'OFF');
  });
});

index.html
```html
<!DOCTYPE html>
<html>
  <head>

```

```

    <title>Lamp Control</title>
  </head>
  <body>
    <h1>Control Lamp</h1>
    <button id="on">Turn ON</button>
    <button id="off">Turn OFF</button>
    <script src="preload.js"></script>
  </body>
</html>
'''

```

### ### Latihan Soal Pemrograman Front-end Web untuk IoT

#### #### Soal 3: Dashboard Monitoring IoT dengan React

Buat aplikasi web menggunakan React yang menampilkan data dari berbagai sensor IoT (misalnya suhu, kelembaban, dan tekanan) secara real-time menggunakan WebSocket.

#### ##### Langkah-langkah:

1. Setup React: Buat proyek baru menggunakan Create React App.
2. Integrasi WebSocket: Buat koneksi WebSocket ke server yang menyediakan data sensor.
3. Display data: Tampilkan data dari sensor dalam bentuk grafik menggunakan library chart (misalnya, Chart.js atau Recharts).

#### ##### Contoh Kode:

```

```javascript
import React, { useEffect, useState } from 'react';
import io from 'socket.io-client';
import { Line } from 'react-chartjs-2';

const socket = io('http://your-server-address');

function App() {
  const [temperature, setTemperature] = useState([]);
  const [humidity, setHumidity] = useState([]);
  const [pressure, setPressure] = useState([]);

  useEffect(() => {
    socket.on('sensorData', data => {
      setTemperature(prevTemp => [...prevTemp,
data.temperature]);
      setHumidity(prevHum => [...prevHum,
data.humidity]);
      setPressure(prevPres => [...prevPres,

```

```

data.pressure]));
    });

    return () => socket.off('sensorData');
  }, []);

const data = {
  labels: Array(temperature.length).fill(''),
  datasets: [
    {
      label: 'Temperature',
      data: temperature,
      fill: false,
      borderColor: 'red',
    },
    {
      label: 'Humidity',
      data: humidity,
      fill: false,
      borderColor: 'blue',
    },
    {
      label: 'Pressure',
      data: pressure,
      fill: false,
      borderColor: 'green',
    },
  ],
};

return (
  <div className="App">
    <h1>IoT Sensor Dashboard</h1>
    <Line data={data} />
  </div>
);
}

export default App;

```

Latihan soal di atas mencakup berbagai aspek pemrograman front-end untuk aplikasi IoT di platform mobile, desktop, dan web. Dengan mengerjakan latihan ini, Anda akan mendapatkan pemahaman yang lebih baik tentang bagaimana mengintegrasikan data IoT ke dalam aplikasi front-end dan bagaimana membuat antarmuka pengguna yang responsif dan interaktif.

## 7.6 Bahan Diskusi

Pemrograman front-end untuk IoT adalah bidang yang sangat dinamis dan memerlukan pemahaman mendalam tentang berbagai teknologi dan praktik terbaik. Berikut adalah beberapa topik yang bisa dijadikan bahan diskusi terkait pemrograman front-end untuk IoT:

### 1. Peran dan Tantangan Pemrograman Front-end dalam IoT

**Desain UI/UX:** Diskusikan bagaimana merancang antarmuka pengguna yang intuitif dan user-friendly untuk aplikasi IoT, yang sering kali harus menampilkan data real-time dan memungkinkan interaksi langsung dengan perangkat IoT.

**Responsivitas dan Skalabilitas:** Bagaimana memastikan aplikasi front-end tetap responsif dan skalabel saat menangani data dalam jumlah besar dari banyak perangkat IoT?

### 2. Teknologi dan Framework yang Digunakan

**Web Development:** Teknologi seperti HTML, CSS, dan JavaScript, serta framework seperti React, Angular, dan Vue.js. Bagaimana masing-masing teknologi ini mendukung pengembangan aplikasi IoT?

**Mobile Development:** Penggunaan React Native, Flutter, atau pengembangan native (Swift untuk iOS, Kotlin/Java untuk Android). Bagaimana memilih teknologi yang tepat untuk aplikasi IoT?

**Desktop Development:** Penggunaan Electron untuk membuat aplikasi desktop cross-platform. Apa kelebihan dan kekurangannya?

## 7.7 Daftar Rujukan

- Adiputra, D., Kristanto, T., Sayid Albana, A., & W. S. (2022). Penerapan Teknologi Hidroponik Berbasis IoT Untuk Mendukung Pengembangan Desa Wisata Edukasi. *ABDINE: Jurnal Pengabdian Masyarakat*, 200-209.
- Karim, S. K. (2021). Sistem Monitoring Pada Tanaman Hidroponik Menggunakan Arduino UNO dan NodeMCU. *Buletin Poltanesa*.
- Liska, B. H. (2021). Implementasi IoT Data Storage Dengan Menggunakan Sistem Basis Data Terdistribusi Berbasis MySQL Cluster. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, . 2986-2993 .

a.

- Syafiqoh, U., Sunardi, S., & Yudhana, A. (2018). Pengembangan Wireless Sensor Network Berbasis Internet of Things untuk Sistem Pemantauan Kualitas Air dan Tanah Pertanian. *Jurnal Informasi dan Pengembangan IT*, vol. 3, no. 2, 285-289.
- Syukhro, I., & Rahmadewi, R. (2021). Penggunaan Aplikasi Blynk Untuk Monitoring dan Kontrol Jarak Jauh pada Sistem Kompos Pintar. *ELECTRICIAN – Jurnal Rekayasa dan Teknologi Elektro*, 1-11.

## **BAB 8. IMPLEMENTASI IOT DALAM AGROINDUSTRI**

### **Capaian Pembelajaran Matakuliah (CPMK)**

Mahasiswa mampu merancang dan membuat sistem IoT secara utuh

### **8.1 Pengembangan Smart Fogponic System Pada Budidaya Tanaman Pakcoy di Pertanian Indoor**

Proses penyerapan nutrisi untuk proses fotosintesis yang dilakukan oleh tanaman bisa dilakukan melalui akar dan daunnya. Proses penyerapan melalui akar bisa dilakukan dengan beberapa cara seperti intersepsi akar, aliran massa, dan difusi ion. Selain penyerapan dengan akar, tumbuhan juga dapat melakukan penyerapan melalui daun dengan menggunakan stomata, kutikula atau penetrasi langsung melalui sel. Proses penyerapan melalui stomata yang hanya berukuran 0,05 hingga 0,1 milimeter akan membantu tumbuhan menyerap air, gas dan cahaya dilingkungannya, sedangkan proses penyerapan melalui kutikula dan penetrasi langsung melalui sel hanya terjadi dengan kondisi tertentu dan apabila unsur hara hanya berbentuk larutan (Suriani, 2018). Berbagai macam proses penyerapan yang dilakukan oleh tanaman tersebut, jumlah nutrisi yang akan terserap oleh tanaman juga dipengaruhi oleh berbagai faktor sehingga jumlah dan kecepatan proses penyerapan setiap tanaman tidak sama.

Salah satu faktor yang mempengaruhi kecepatan penyerapan nutrisi oleh tanaman adalah media tanam atau tempat akar tanaman yang menyerap unsur hara. Salah satu media tanam yang dikembangkan untuk mempercepat proses penyerapan nutrisi adalah air atau biasa disebut dengan metode tanam hidroponik. Hidroponik dibagi menjadi beberapa tipe diantaranya wick, Deep Water Culture (DWC), EBB and Flow, hidroponik tetes, Nutrisi Teknik Film (NFT), Aeroponik dan Fogponic. Beberapa hidroponik tersebut, tipe fogponic merupakan tipe yang dapat membantu mempercepat penyerapan nutrisi tanaman dengan tingkat oksigen yang tinggi dan minimnya jumlah larutan nutrisi yang digunakan.

Sistem fogponic merupakan pengembangan dari sistem

aeroponik. Sistem ini menggunakan air yang diubah dalam bentuk kabut dengan ukuran 5 – 30 mikron (m) guna memawa nutrisi ke akar tanaman (Modu et al., 2020). Keunggulan fogponic dibandingkan dengan aeroponik yakni terletak pada efisiensi penyerapan nutrisi oleh akar tanaman yang lebih cepat, karena ukuran tetesan (kabut) yang lebih kecil mendorong penyerapan nutrisi yang lebih baik. Kabut juga dapat menjangkau lebih banyak bagian akar tanaman dibandingkan dengan metode semprot yang ada pada aeroponik (Dr Peter Wootton-Beard, 2019). Parameter penting yang harus diperhatikan dalam penanaman pola fogponic yaitu kuantitas nutrisi yang diberikan, tingkat keasaman (pH), suhu, kelembapan, dan juga penjadwalan pencahayaan (Shyam et al., 2022a). Metode fogponic ini sangat cocok diterapkan pada beberapa jenis tanaman sayur yang memiliki ukuran akar lebih kecil seperti pakcoy.

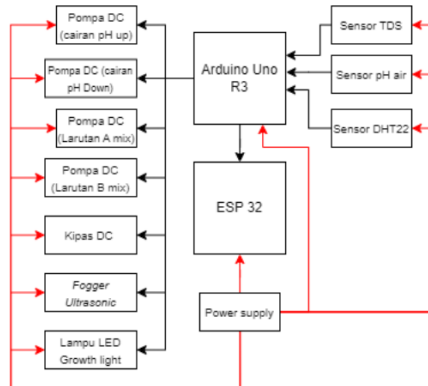
Berdasarkan beberapa poin tersebut, maka diperlukan sebuah sistem guna mengendalikan dan memantau penanaman menggunakan metode fogponic. Sistem ini akan berfokus pada perubahan kadar nutrisi, pH, suhu, kelembapan dan cahaya pada penerapan secara real time. Data yang diperoleh dari system smart fogponic ini akan ditampilkan melalui website yang dapat diakses oleh user dimana saja selama terhubung dengan internet.

#### 8.1.1 Rancangan Sistem Smart Fogponic

Rancangan Sistem Smart Fogponic, dalam tahap desain dan perancangan sistem bertujuan memastikan pengembangan dilakukan secara sistematis yang mencakup perancangan perangkat keras, perangkat lunak, dan kemasan perangkat keras sistem.

##### a. Perangkat Keras (Hardware)

Perangkat keras dari fogponic terdiri dari beberapa komponen penting. Beberapa komponen tersebut diantaranya dua buah, tiga buah sensor, Power supplay, Relay, dan tujuh aktuator. Komponen ini akan mendapatkan data lingkungan dari fogponic dan memberikan respon melalui aktuator. Semua data yang diperoleh akan di kirimkan pada server untuk di tampilkan di website secara real time. Rancangan perangkat keras dalam bentuk blok diagram ditunjukan dalam gambar 8-1.



Gambar 8-1 Rancangan sistem fogponic

#### b. Perancangan Perangkat Lunak (Software)

Proses pengelolaan data dipengaruhi dari rancangan hardware. Tahap awal melibatkan pemrograman perangkat keras, yang disebut source node, untuk mengumpulkan data suhu, kelembaban, pH air, total dissolved solid (TDS), dan intensitas cahaya melalui sensor di sekitar tanaman. Data sensor dikemas dalam satu frame paket dan dikirim ke gateway melalui jaringan wireless setiap menit. Gateway memproses data, mengonversinya menjadi variabel string, dan menyimpannya dalam tabel data\_sensor di database. Selanjutnya, server berfungsi sebagai penghubung ke jaringan lokal dan publik. Server melakukan subscribe data sensor ke gateway dan publish data sensor ke public MQTT broker, memungkinkan akses melalui jaringan publik. Data yang diterima oleh server ditampilkan secara real-time melalui aplikasi web Node-RED Dashboard, yang hanya dapat diakses di jaringan lokal. Di jaringan publik, kondisi lingkungan pertanian dapat dipantau melalui aplikasi MQTT Dashboard di smartphone.

#### 8.1.2 Rangkaian arsitektur perangkat keras sistem smart fogponic

#### 8.1.3 Kode Program sistem smart fogponics

Kode program sistem ini dibuat untuk dua perangkat, arduino uno dan ESP8266

Kode program arduino uno sebagai berikut

```

#include "GravityTDS.h"
#include "DHT.h"
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>

```



```

#include <Wire.h>
#include "RTClib.h"

RTC_DS3231 rtc;
char daysOfTheWeek[7][12] = {"Minggu", "Senin",
"Selasa", "Rabu", "Kamis", "Jumat", "Sabtu"};
#define detik now.second()
#define menit now.minute()
#define jam now.hour()
#define tanggal now.day()
#define hari daysOfTheWeek[now.dayOfTheWeek()]
#define bulan now.month()
#define tahun now.year()

LiquidCrystal_I2C lcd(0x27, 16, 2);
SoftwareSerial serialtoESP(3, 4); // RX, TX

#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

const int TdsSensorPin = A1;
#define TdsSensorPin A1
GravityTDS gravityTds;
float temperature = 25, tdsValue = 0;

int pinPH = A0;
float calibration_value = 21.72; //
int phval = 0;
unsigned long int avgval;
int buffer_arr[10], temp;

int pompa = 6;
int pumpb = 7;
int pumpphdown = 8;
int pumpphup = 9;
int fogger = 10;
int kipas = 11;
int lamp = 12;
int aerator = 13;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  serialtoESP.begin(9600);
  Wire.begin();
  rtc.begin();
  dht.begin();

  if (! rtc.begin()) {
    Serial.println("Couldn't find RTC");

```

```

        while (1);
    }
    if (! rtc.lostPower()) {
        Serial.println("RTC lost power, lets set the
time!");
        // setting otomatis sesuai waktu program di compile
        rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
        // setting manual
        // January 21, 2014 at 3am you would call:
        rtc.adjust(DateTime(2021, 12, 30, 22, 12, 0));
    }
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));

    gravityTds.setPin(TdsSensorPin);
    gravityTds.setAref(5.0); //reference voltage on
ADC, default 5.0V on Arduino UNO
    gravityTds.setAdcRange(1024); //1024 for 10bit
ADC; 4096 for 12bit ADC
    gravityTds.begin(); //initialization

    pinMode(pumpa, OUTPUT);
    pinMode(pumpb, OUTPUT);
    pinMode(pumpphdown, OUTPUT);
    pinMode(pumpphup, OUTPUT);
    pinMode(fogger, OUTPUT);
    pinMode(kipas, OUTPUT);
    pinMode(lamp, OUTPUT);
    pinMode(aerator, OUTPUT);

    lcd.init();
    lcd.backlight();

    lcd.setCursor(0, 0); //tampilan lcd awal
    lcd.print(" MONITORING ");
    lcd.setCursor(0, 1);
    lcd.print(" FOGPONIC ");
    delay(2000);
    lcd.clear();
}

void loop() {
    // put your main code here, to run repeatedly:
    DateTime now = rtc.now();
    Serial.print(hari);
    Serial.print(", ");
    Serial.print(jam, DEC);
    Serial.print(":");
    Serial.print(menit, DEC);
    Serial.print(":");
    Serial.print(detik, DEC);
    Serial.println(" ");
}

```

```

for(int i=0; i<10; i++) {
    buffer_arr[i] = analogRead(pinPH);
    delay(30);
}
for(int i=0; i<9; i++) {
    for(int j=i+1; j<10; j++) {
        if (buffer_arr[i] > buffer_arr[j]) {
            temp = buffer_arr[i];
            buffer_arr[i] = buffer_arr[j];
            buffer_arr[j] = temp;
        }
    }
}
avgval = 0;

for(int i=2; i<8; i++)
    avgval += buffer_arr[i];

float volt=(float)avgval*5.0/1024/6;
float ph_act = -5.70 * volt + calibration_value;
Serial.println(ph_act);

gravityTds.setTemperature(temperature); // set the
temperature and execute temperature compensation
gravityTds.update(); //sample and calculate
tdsValue = gravityTds.getTdsValue(); // then get
the value
Serial.print(tdsValue,0);
Serial.println("ppm");

float t = dht.readTemperature();
float h = dht.readHumidity();
// if (isnan(h) || isnan(t)) {
//     Serial.println("Failed to read from DHT
sensor!");
//     return;
// }
Serial.print("Temperature: ");
Serial.print(t);
Serial.print(" *C ");
Serial.print("Humidity: ");
Serial.println(h);

lcd.setCursor(0, 0);
lcd.print("PH: ");
lcd.print(ph_act);
lcd.print(" ");

lcd.setCursor(9, 0);
lcd.print("TDS: ");

```

```

lcd.print(tdsValue,0);
lcd.print(" ");

lcd.setCursor(0, 1);
lcd.print("T: ");
lcd.print(t,1);
lcd.print("C ");

lcd.setCursor(9, 1);
lcd.print("H: ");
lcd.print(h,1);
lcd.print("% ");

digitalWrite (aerator,HIGH);

if (tdsValue < 800) {
    digitalWrite(pumpa, HIGH);
    digitalWrite(pumpb, HIGH);
}
else {
    digitalWrite(pumpa, LOW);
    digitalWrite(pumpb, LOW);
}

if (ph_act > 6.5) {
    digitalWrite(pumpphdown, HIGH);
}
else {
    digitalWrite(pumpphdown, LOW);
}
if (ph_act < 5) {
    digitalWrite(pumpphup, HIGH);
}
else {
    digitalWrite(pumpphup, LOW);
}

if (t > 28 && h < 85) {
    digitalWrite(fogger, HIGH);
    digitalWrite(kipas, LOW);
}
if (t < 28 && h > 98) {
    digitalWrite(fogger, LOW);
    digitalWrite(kipas, HIGH);
}
if (t > 28 && h > 98) {
    digitalWrite(fogger, LOW);
    digitalWrite(kipas, HIGH);
}
if (t < 28 && h < 85) {
    digitalWrite(fogger, HIGH);

```

```

    digitalWrite(kipas, LOW);
}

if (jam >= 5 && jam <= 18) {
    digitalWrite(lamp, HIGH);
}
if (jam < 5) {
    digitalWrite(lamp, LOW);
}
if (jam > 18) {
    digitalWrite(lamp, LOW);
}

int datapumpa = digitalRead(pumpa);
int datapumpb = digitalRead(pumpb);
int datapumpphdown = digitalRead(pumpphdown);
int datapumpphup = digitalRead(pumpphup);
int datafogger = digitalRead(fogger);
int datakipas = digitalRead(kipas);
int datalamp = digitalRead(lamp);
int dataaerator = digitalRead(aerator);

Serial.print(datapumpa);

// -- Kirim data ke ESP --
serialtoESP.print(ph_act);
serialtoESP.print("A");
serialtoESP.print(tdsValue);
serialtoESP.print("B");
serialtoESP.print(t);
serialtoESP.print("C");
serialtoESP.print(h);
serialtoESP.print("D");
serialtoESP.print(datapumpa);
serialtoESP.print("E");
serialtoESP.print(datapumpb);
serialtoESP.print("F");
serialtoESP.print(datapumpphdown);
serialtoESP.print("G");
serialtoESP.print(datapumpphup);
serialtoESP.print("H");
serialtoESP.print(datafogger);
serialtoESP.print("I");
serialtoESP.print(datakipas);
serialtoESP.print("J");
serialtoESP.print(datalamp);
serialtoESP.print("K");
serialtoESP.print(dataaerator);
serialtoESP.print("L");
serialtoESP.print("\n");

```

```

    delay(1000);
}

```

**Kode program untuk ESP8266 sebagai berikut:**

```

#include <WiFi.h>
#include <HTTPClient.h>
#include <SoftwareSerial.h>

SoftwareSerial serialtoESP(19, 18);  // RX, TX

char ph_act[10];
char tdsValue[10];
char t[10];
char h[10];
char datapumpa[10];
char datapumpb[10];
char datapumpphdown[10];
char datapumpphup[10];
char datafogger[10];
char datakipas[10];
char datalamp[10];
char dataaerator[10];

char c;
String dataIn;
int8_t indexA, indexB, indexC, indexD, indexE, indexF,
indexG;
int8_t indexH, indexI, indexJ, indexK, indexL;

int ledwifi = 2;

char ssid[] = "Azkiya";
char pass[] = "Senyumduluyaa";

void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
    serialtoESP.begin(9600);

    pinMode(ledwifi, OUTPUT);

    // connect to wifi.
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, pass);
    Serial.print("connecting");

    while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(500);
        digitalWrite(ledwifi, LOW);
    }
}

```

```

    }
    Serial.println();
    Serial.print("connected: ");
    Serial.println(WiFi.localIP());
    digitalWrite(ledwifi, HIGH);
}

void loop() {
    // put your main code here, to run repeatedly:
    while (serialtoESP.available() > 0) {
        c = serialtoESP.read();
        if (c == '\n') {
            break;
        }
        else {
            dataIn += c;
        }
    }

    if (c == '\n') {
        ParseData();

        Serial.print(ph_act);
        Serial.print(" ");
        Serial.print(tdsValue);
        Serial.print(" ");
        Serial.print(t);
        Serial.print(" ");
        Serial.print(h);
        Serial.print(" ");
        Serial.print(datapumpa);
        Serial.print(" ");
        Serial.print(datapumpb);
        Serial.print(" ");
        Serial.print(datapumpphdown);
        Serial.print(" ");
        Serial.print(datapumpphup);
        Serial.print(" ");
        Serial.print(datafogger);
        Serial.print(" ");
        Serial.print(datalamp);
        Serial.print(" ");
        Serial.print(dataaerator);

        c = 0;
        dataIn = "";
    }

    sendDataToServer();
    delay(5000); // Kirim data setiap 5 detik
}

```

```

void ParseData() {
    indexA = dataIn.indexOf("A");
    indexB = dataIn.indexOf("B");
    indexC = dataIn.indexOf("C");
    indexD = dataIn.indexOf("D");
    indexE = dataIn.indexOf("E");
    indexF = dataIn.indexOf("F");
    indexG = dataIn.indexOf("G");
    indexH = dataIn.indexOf("H");
    indexI = dataIn.indexOf("I");
    indexJ = dataIn.indexOf("J");
    indexK = dataIn.indexOf("K");
    indexL = dataIn.indexOf("L");

    strcpy(ph_act, dataIn.substring(0,
indexA).c_str());
    strcpy(tdsValue, dataIn.substring(indexA+1,
indexB).c_str());
    strcpy(t, dataIn.substring(indexB+1,
indexC).c_str());
    strcpy(h, dataIn.substring(indexC+1,
indexD).c_str());
    strcpy(datapumpa, dataIn.substring(indexD+1,
indexE).c_str());
    strcpy(datapumpb, dataIn.substring(indexE+1,
indexF).c_str());
    strcpy(datapumpphdown, dataIn.substring(indexF+1,
indexG).c_str());
    strcpy(datapumpphup, dataIn.substring(indexG+1,
indexH).c_str());
    strcpy(datafogger, dataIn.substring(indexH+1,
indexI).c_str());
    strcpy(datakipas, dataIn.substring(indexI+1,
indexJ).c_str());
    strcpy(datalamp, dataIn.substring(indexJ+1,
indexK).c_str());
    strcpy(dataaerator, dataIn.substring(indexK+1,
indexL).c_str());
}

void sendDataToServer() {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;

        http.begin("https://sheisfonics.000webhostapp.com/data.p
hp"); // URL untuk mengirimkan permintaan
        http.addHeader("Content-Type", "application/x-
www-form-urlencoded");

        // Membangun data POST

```



```

        String postData = "ph_act=" + String(ph_act);
        postData += "&tdsValue=" + String(tdsValue);
        postData += "&t=" + String(t);
        postData += "&h=" + String(h);
        postData += "&datapumpa=" + String(datapumpa);
        postData += "&datapumpb=" + String(datapumpb);
        postData += "&datapumpphdown=" +
String(datapumpphdown);
        postData += "&datapumpphup=" +
String(datapumpphup);
        postData += "&datafogger=" + String(datafogger);
        postData += "&datakipas=" + String(datakipas);
        postData += "&datalamp=" + String(datalamp);
        postData += "&dataaerator=" +
String(dataaerator);

        // Mengirim data POST ke server
        int httpCode = http.POST(postData);

        // Memeriksa respon HTTP
        if (httpCode > 0) {
            // Respon berhasil diterima
            String payload = http.getString();
            Serial.println(httpCode);
            Serial.println(payload);
        } else {
            // Gagal mengirim permintaan
            Serial.println("Error on HTTP request");
        }

        // Menutup koneksi
        http.end();
    }
}

// Send the POST

```

## 8.2 Implementasi Wireles Serial Master-Slave pada Sistem Penyemprotan Otomatis serta Monitoring Suhu dan Kelembapan Udara Budidaya Jamur Janggel

Budidaya jamur janggel telah dikenal sebagai salah satu alternatif pengolahan limbah organik termasuk bonggol jagung yang menjadi produk bernilai ekonomi tinggi. Jamur janggel tidak hanya memiliki nilai ekonomi yang tinggi, tetapi juga kaya akan kandungan gizi yang sangat baik bagi kesehatan manusia seperti protein, vitamin, mineral, dan serat (Rich Milton R Dulay 2014). Budidaya jamur janggel cukup

sederhana, proses pertumbuhannya menggunakan media tanam bonggol jagung yang dicampur dengan urea, dedak/bekatul dan ragi tape/tempe lalu disimpan didalam kumbung atau rumah jamur. Hal yang perlu di perhatikan dalam pengolahan budidaya jamur janggél adalah lokasi pembuatan dan kondisi lingkungan, lokasi yang di gunakan harus tetap terkena sinar matahari tetapi terlindungi dari air hujan karena jamur janggél membutuhkan kondisi udara yang lembab dengan suhu udara yang tinggi untuk pertumbuhannya. Media tanam jamur janggél diusahakan di atas permukaan tanah (Atikah 2022). Suhu dan kelembaban yang sesuai sangat penting untuk memastikan pertumbuhan jamur yang optimal. Jamur janggél dapat tumbuh dengan baik dan optimal pada kelembapan kurang lebih 80%. Suhu dan kelembapan udara berperan penting dalam proses inkubasi miselium kisaran suhu yang optimal untuk pertumbuhan miselium *C. comatus* pada rentang 23-26° (Hidayat 2023). Suhu dan kelembaban sangat mempengaruhi pertumbuhan tanaman jamur. Jamur akan terhambat pertumbuhannya ketika suhu dan kelembaban tidak sesuai sehingga mengering dan layu. Oleh karena itu, perlunya sistem pengendalian otomatis yang dapat mengatur suhu dan kelembapan kumbung budidaya jamur janggél agar tetap terjaga setiap saat seperti yang diharapkan dengan tujuan untuk meningkatkan produksi dan meringankan kerja pembudidaya.

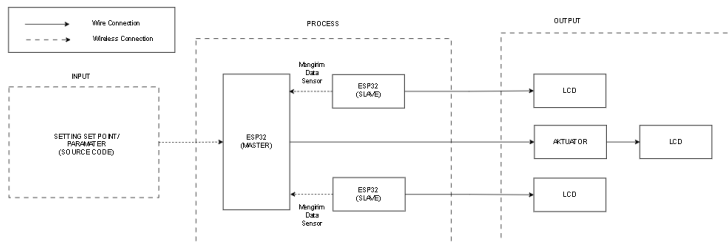
### 8.2.1 Desain Sistem

Tahap desain dan perancangan sistem mencakup perancangan perangkat keras, perangkat lunak, dan letak komponen sistem. Arsitektur sistem mencakup sensor suhu, sensor kelembapan, dan penyiraman otomatis yang terdiri dari perangkat keras (NodeMCU, sensor suhu dan kelembapan DHT22, pompa air mini, kipas angin, dan laptop).

#### 1. Perancangan Perangkat Keras (Hardware)

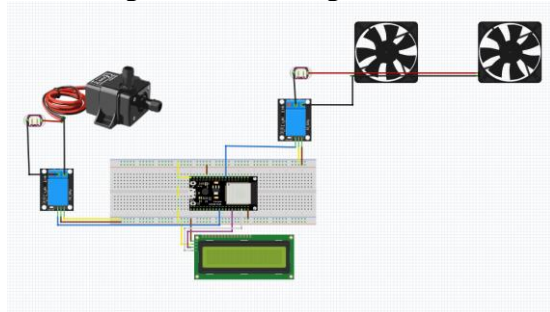
Pengembangan penyemprotan otomatis dan monitoring suhu serta kelembapan pada budidaya jamur janggél dengan menerapkan *wireless serial master-slave* menggunakan beberapa sensor, seperti DHT22 dan *Ultrasonic*. Sensor ini membantu mengukur suhu, kelembapan dan ketersediaan air pada wadah penampungan air. Proses sistem dimulai dari setiap slave yang ditempatkan dua sensor DHT22. Sensor suhu dan kelembapan pada setiap perangkat slave mengukur kondisi lingkungan setiap satu jam. Proses selanjutnya yakni data yang diukur dikirim melalui protokol ESP-NOW ke perangkat master. Perangkat master mengumpulkan dan menganalisis

data dari semua perangkat slave. Data dari sensor akan diproses oleh mikrokontroler sebagai master untuk mengaktifkan exhaust fan dan pompa air dalam menjaga kestabilan suhu dan kelembapan di dalam kumbung. Data juga akan ditampilkan pada LCD di kumbung. Gambaran sistem ditunjukkan dalam gambar blok diagram 8-3

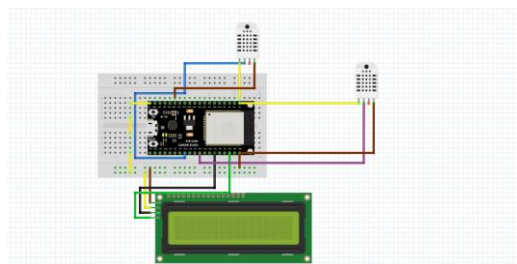


Gambar 8-2 Block Diagram System

Gambar rangkaian perangkat sistem untuk bagian master dan slave ditunjukkan dalam gambar 8-4 dan gambar 8-5 berikut.



Gambar 8-3 Rangkaian perangkat sistem bagian master

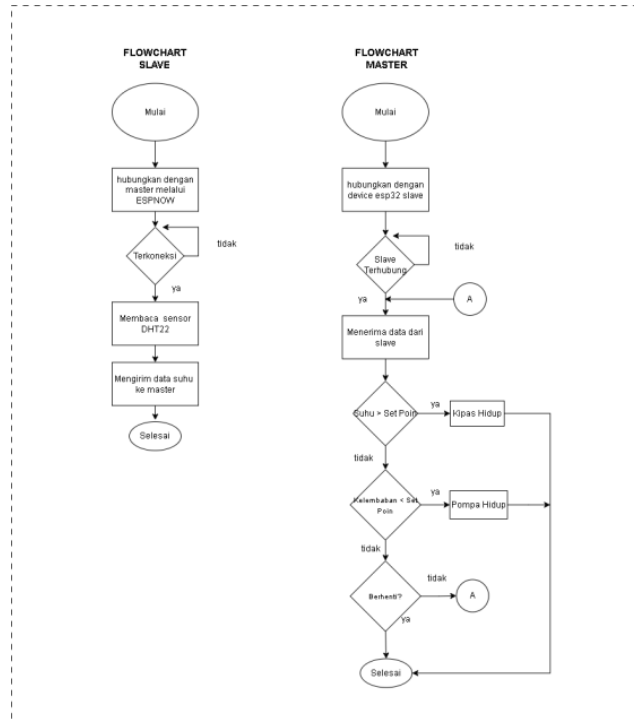


Gambar 8-4 Rangkaian perangkat bagian slave

## 2. Perancangan Perangkat Lunak (Software)

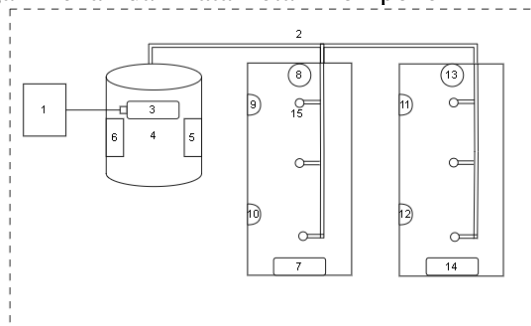
Dalam tahap perancangan sistem penyemprotan otomatis serta monitoring suhu dan kelembapan proses budidaya jamur janggel

berbasis *wireless serial master-slave*, peneliti menggunakan perangkat lunak Arduino IDE untuk mengkode program pada mikrokontroler.



Gambar 8-5 Flowchart Sistem

### Perancangan Denah dan Tata Letak Komponen



Gambar 8-6 Tata Letak Komponen pada Kumbung Jamur sebagai Slave

#### Keterangan:

1 = Sumber listrik

7,14 = Mainboard Slave + LCD

2	= Pipa	8,13	= Kipas Exhaust
3	= Mikrokontroller Master + LCD	9,10	= Sensor DHT Slave 1
4	= Penampung Air	11,12	= Sensor DHT Slave 2
5	= Pompa Air	15	= Nozzle Spray
6	= Sensor Ultrasonic		

### 8.2.2 Pemrograman system

#### Kode program master

```
#include <WiFi.h>
#include <esp_now.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C LCD = LiquidCrystal_I2C(0x27, 16,
2);

// struktur data yang diterima
typedef struct struct_message {
    char a[32];
    float b;
    float c;
} struct_message;
struct_message data_ku;

int totalSuhu = 0;
int totalKelembapan = 0;
int totalDataDiterima = 0;

int relayPin1 = 26;
int relayPin2 = 32;

// callback jika menerima data
void cb_terima(const uint8_t *mac_addr, const uint8_t
*dataDiterima, int panjang) {
    memcpy(&data_ku, dataDiterima, sizeof(data_ku));
    Serial.print("Bytes diterima: ");
    Serial.println(panjang);
    Serial.print("Char: ");
    Serial.println(data_ku.a);
    Serial.print("Kelembapan: ");
    Serial.println(data_ku.b);
    Serial.p...

#include <WiFi.h>
```

```

int relayPin1 = 26;
int relayPin2 = 32;

void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    WiFi.mode(WIFI_MODE_STA);
    Serial.println(WiFi.macAddress());

    pinMode(relayPin1, OUTPUT);
    pinMode(relayPin2, OUTPUT);

    digitalWrite(relayPin1, LOW);
    digitalWrite(relayPin2, LOW);
}

void loop() {
    // put your main code here, to run repeatedly:
    delay(5000);
    Serial.println(WiFi.macAddress());
}

```

### Kode program slave

```

#include <WiFi.h>
#include <esp_now.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C LCD = LiquidCrystal_I2C(0x27, 16, 2);

// struktur data yang diterima
typedef struct struct_message {
    char a[32];
    float b;
    float c;
} struct_message;
struct_message data_ku;

int totalSuhu = 0;
int totalKelembapan = 0;
int totalDataDiterima = 0;

int relayPin1 = 26;
int relayPin2 = 32;

// callback jika menerima data
void cb_terima(const uint8_t *mac_addr, const uint8_t
*dataDiterima, int panjang) {
    memcpy(&data_ku, dataDiterima, sizeof(data_ku));
    Serial.print("Bytes diterima: ");

```

```

Serial.println(panjang);
Serial.print("Char: ");
Serial.println(data_ku.a);
Serial.print("Kelembapan: ");
Serial.println(data_ku.b);
Serial.print("Suhu: ");
Serial.println(data_ku.c);

totalSuhu += data_ku.c;
totalKelembapan += data_ku.b;
totalDataDiterima++;
    Serial.print("total Data Diterima: ");
    Serial.println(totalDataDiterima);
if (totalDataDiterima == 2) {
    float rataRataSuhu = totalSuhu/4;
    float rataRataKelembapan = totalKelembapan/4;
    LCD.setCursor(0, 0);
    LCD.print("Temp = " + String(rataRataSuhu));
    LCD.setCursor(0, 1);
    LCD.print("Klmbp = " + String(rataRataKelembapan));
    delay(5000);
    LCD.clear();
    if (rataRataSuhu > 28) {
        Serial.print("Sensor Suhu: ");
        Serial.println(rataRataSuhu);
        digitalWrite(relayPin1, LOW);
        LCD.setCursor(0, 0);
        LCD.print("Kipas Nyala");
    } if (rataRataSuhu < 28) {
        Serial.print("Sensor Suhu: ");
        Serial.println(rataRataSuhu);
        digitalWrite(relayPin1, HIGH);
        LCD.setCursor(0, 0);
        LCD.print("Kipas Mati");
    } if (rataRataKelembapan < 80) {
        Serial.print("Sensor Suhu: ");
        Serial.println(rataRataKelembapan);
        digitalWrite(relayPin2, LOW);
        LCD.setCursor(0, 1);
        LCD.print("Pompa Nyala");
        delay(5000);
    } if (rataRataKelembapan > 80) {
        Serial.print("Sensor Suhu: ");
        Serial.println(rataRataKelembapan);
        digitalWrite(relayPin2, HIGH);
        LCD.setCursor(0, 1);
        LCD.print("Pompa Mati");
        delay(5000);
    }
    totalSuhu = 0;
    totalKelembapan = 0;

```

```

        totalDataDiterima = 0;
        LCD.clear();
    }
}

void setup() {
    Serial.begin(115200);

    LCD.init();
    LCD.backlight();
    LCD.setCursor(0, 0);

    // mengatur esp ke mode station
    WiFi.mode(WIFI_STA);

    // inisialisasi espnow
    if (esp_now_init() != ESP_OK) {
        Serial.println("Gagal Inisialisasi espnow");
        return;
    }

    pinMode(relayPin1, OUTPUT);
    pinMode(relayPin2, OUTPUT);

    digitalWrite(relayPin1, LOW);
    digitalWrite(relayPin2, LOW);

    // mendaftarkan fungsi callback
    esp_now_register_recv_cb(cb_terima);
}

void loop() {
}

```

### 8.3 Bahan Diskusi

Irigasi Cerdas:

- Sistem irigasi otomatis yang menggunakan data dari sensor kelembaban tanah untuk mengoptimalkan penggunaan air.
- Studi kasus: Penggunaan sistem irigasi tetes berbasis IoT di lahan pertanian.

### 8.4 Daftar Rujukan

Adiputra, D., Kristanto, T., Sayid Albana, A., & W. S. (2022). Penerapan Teknologi Hidroponik Berbasis IoT Untuk Mendukung



- Pengembangan Desa Wisata Edukasi. *ABDINE: Jurnal Pengabdian Masyarakat*, 200-209.
- Atikah, M. D. (2022). PENGOLAHAN BONGGOL JAGUNG SEBAGAI MEDIA TANAM JAMUR JANGGELDI DESA KURIPAN UTARA KECAMATAN KURIPAN KABUPATEN LOMBOK BARAT. *Jurnal Pengabdian Masyarakat*, 276.
- Chusna, F. (2023, Mei 10). <https://www.linknet.id>. Retrieved from contoh-iot-dalam-bidang-pertanian: <https://www.linknet.id/article/contoh-iot-dalam-bidang-pertanian>
- Hidayat, M. (2023). *SKRIPSI: UJI BERBAGAI KOMPOSISI MEDIA TANAM JAMUR JANGGEL (Coprinus comatus) JERAMI PADI DAN TONGKOL JAGUNG DENGAN BEBERAPA NUTRISI ORGANIK*. Medan: Universitas Medan Area.
- Karim, S. K. (2021). Sistem Monitoring Pada Tanaman Hidroponik Menggunakan Arduino UNO dan NodeMCU. *Buletin Poltanesa*.
- Liska, B. H. (2021). Implementasi IoT Data Storage Dengan Menggunakan Sistem Basis Data Terdistribusi Berbasis MySQL Cluster. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, . 2986-2993 .
- Rich Milton R Dulay, W. S. (2014). Aseptic Cultivation and Nutrient Compositions of *Coprinus comatus* (O.F. Müll.) Pers. On *Pleurotus* Mushroom Spent. *Journal of Microbiology and Biotechnology Research*, 1-7.
- Suriani, D. N. (2018). CORRELATION OF STOMATA DENSITY TO RUST SEVERITY ON SOME ACCESSIONS OF MAIZE GERMPLASM. . *Journal of Tropical Plant Pests and Diseases*, 95-104.
- Syafiqoh, U., Sunardi, S., & Yudhana, A. (2018). Pengembangan Wireless Sensor Network Berbasis Internet of Things untuk Sistem Pemantauan Kualitas Air dan Tanah Pertanian. *Jurnal Informasi dan Pengembangan IT*, vol. 3, no. 2, 285-289.
- Syukhro, I., & Rahmadewi, R. (2021). Penggunaan Aplikasi Blynk Untuk Monitoring dan Kontrol Jarak Jauh pada Sistem Kompos Pintar. *ELECTRICIAN – Jurnal Rekayasa dan Teknologi Elektro*, 1-11.
- Yulianti, S. (2022, Februari 23). <https://www.indoniaga.co.id>. Retrieved from pertanian-green-house-berbasis-iot: <https://www.indoniaga.co.id/2022/02/pertanian-green-house-berbasis-iot.html>

## **DAFTAR ISTILAH (GLOSARIUM)**

## **BIOGRAFI PENULIS**

## **RINGKASAN BUKU AJAR**