BAB II
TINJAUAN PUSTAKA

1. Penelitian-Penelitian Sebelumnya

Penelitian mengenai *Cable Driven Parallel Robot* (CDPR) dimulai di Amerika pada tahun 1984. Inovasi ini dirancang khusus untuk operasi di bawah air. Lima tahun kemudian, pada tahun 1989, proyek Robot *Crane* dimulai di Amerika oleh National *Institute of Standards and Technology* (NIST). Proyek ini bertujuan untuk mengembangkan aplikasi yang luas terutama dalam pemrosesan mesin, penanganan kargo pelabuhan, konstruksi jembatan, pengelasan jembatan, dan bidang lainnya. Sejak itu, minat terhadap penelitian tentang CDPR telah meningkat pesat, didorong oleh kebutuhan akan kapasitas dan ruang kerja yang lebih besar dalam konteks teknologi modern. CDPR telah berhasil diterapkan dalam berbagai tugas yang relevan, seperti konstruksi dan sistem penyelamatan [4].

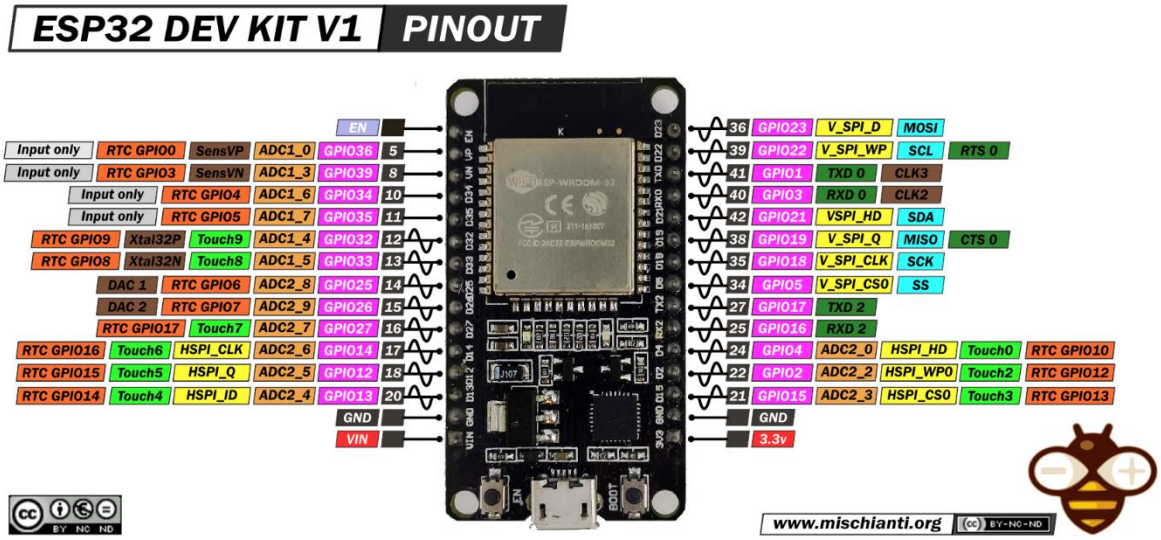
Ada juga penelitian yang membuat sebuah *prototype*, yaitu dengan tinggi tiang 1m, panjang dan lebar 0,5 meter, menggunakan empat motor *stepper* sebagai aktuator, menggunakan empat benang senar nilon sebagai kabel, dan bandul pemberat seberat 200 gram yang didefinisikan sebagai objek yang akan dipindahkan. Sistem yang dibuat akan bisa dikontrol secara manual dengan menekan enam buah tombol arah navigasi kemudian menekan tombol STOP untuk memberhentikannya dan otomatis dengan memasukkan angka koordinat pada kolom sumbu X, Y, dan Z kemudian menekan tombol SET untuk menjalankannya[5].

Pada penelitian ini peneliti akan merujuk pada [6] sebagai bahan penelitian-penelitian. sebuah *Cable Driven Parallel Robot* (CDPR) berbasis Arduino UNO yang digunakan untuk memindahkan objek pada area *greenhouse*, dengan ukuran tiang tinggi 100cm, Panjang 50cm, dan lebar 50cm. Menggunakan empat buah motor *stepper* sebagai aktuator, benang sebagai kabel dan empat buah penyangga

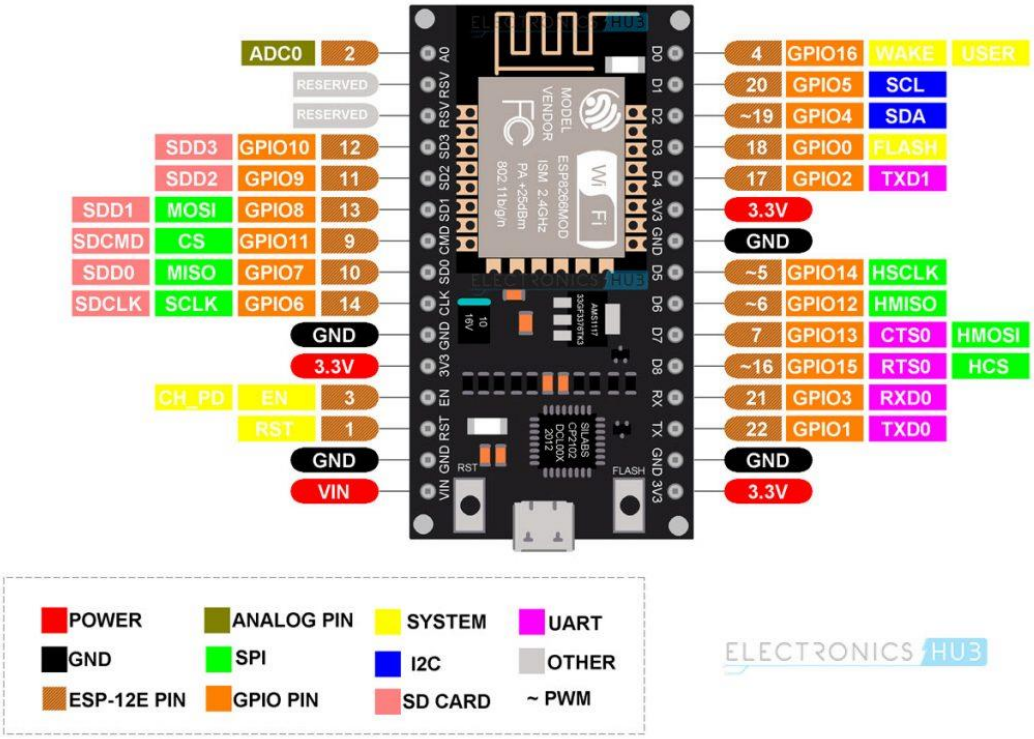
benang yang dapat mengubah posisi bentuk benang dari bentuk Asimetris dan Simetris. Serta bandul seberat 200gram yang digunakan sebagai objek yang akan digerakkan. Robot ini dapat dikendalikan secara manual dengan menentukan titik koordinat yang dimasukkan pada sumbu X, Y, dan Z. Yang digunakan untuk mengawasi area *Greenhouse*. Dengan robot ini diharapkan dapat membantu petani dalam melakukan pemantauan di *Greenhouse* [6].

2. Mikrokontroler

ESP32 dan ESP8266 adalah dua jenis mikrokontroler yang akan digunakan dalam pembuatan sistem modular terdistribusi pada penelitian ini. Keduanya dikembangkan oleh *Espressif Systems* dan memiliki kemiripan dalam kemampuan konektivitas *WiFi, Bluetooth*,, GPIO (*General Purpose Input/Output*), dan keduanya dapat menggunakan ESP-NOW ada dalam fitur dan spesifikasi keduanya.



Gambar 2.1 ESP32 dan Pin I/O



Gambar 2.2 ESP8266 dan Pin I/O

Dalam konteks aplikasi pada CDPR, ESP32 dapat digunakan untuk mengelola tugas-tugas kontrol yang lebih kompleks dan membutuhkan pemrosesan yang lebih tinggi, sementara ESP8266 dapat digunakan untuk tugas-tugas yang lebih sederhana atau untuk memperluas jaringan komunikasi dalam sistem terdistribusi.

Dengan memanfaatkan kedua mikrokontroler ini secara bersamaan dalam sistem CDPR, kita dapat memanfaatkan keunggulan masing-masing platform untuk mencapai tujuan desain yang lebih baik dan lebih efisien.

Tabel 2.1 Spesifikasi Mikrokontroler

NO	Fitur/Spesifikasi	ESP32	ESP8266
1.	Prosesor	*Dual-core Tensilica Xtensa* LX6, 32-bit	*Single-core Tensilica* L106, 32-bit
2.	Kecepatan Prosesor	Hingga 240 MHz	Hingga 160 MHz
3.	Memori Program	16 MB (pada beberapa model)	512 KB - 4 MB

	(*Flash*)		
4.	Memori RAM	520 KB	80 KB - 160 KB
5.	*WiFi*	802.11 b/g/n/ac (2.4 GHz)	802.11 b/g/n (2.4 GHz)
6.	*Bluetooth*	*Bluetooth* 4.2 dan *Bluetooth* Low Energy (BLE)	Tidak mendukung
7.	GPIO	34 - 39 (tergantung model)	17 - 19
8.	Antarmuka	SPI, I2C, UART, ADC, DAC	SPI, I2C, UART, ADC
9.	Konsumsi Daya	Rata-rata 80 mA (aktif), 5 µA (dalam mode *deep sleep*)	Rata-rata 80 mA (aktif), 20 µA (dalam mode *deep sleep*)
10.	Harga	Lebih mahal	Lebih terjangkau
11.	Ketersediaan	Lebih umum	Umum

Dalam *Cable Driven Parallel Robot* (CDPR), penggunaan ESP32 dan ESP8266 memiliki peran yang vital dalam mengendalikan berbagai aspek sistem:

- Pengendalian Motor: Baik ESP32 maupun ESP8266 dapat digunakan untuk menghasilkan sinyal PWM yang diperlukan untuk mengendalikan motor *stepper* atau motor DC yang menggerakkan kabel-kabel robot. Dengan kemampuan ini, keduanya dapat memungkinkan gerakan yang presisi dan koordinasi yang tepat pada CDPR.
- Komunikasi Jaringan: ESP32 dan ESP8266 memiliki kemampuan konektivitas ESP-NOW yang memungkinkan komunikasi nirkabel antara modul-modul dalam sistem. Hal ini memungkinkan koordinasi gerakan dan pertukaran data antara modul-modul yang terdistribusi dalam sistem CDPR.
- Pemrosesan Sensor: Sistem CDPR sering kali dilengkapi dengan sensor-sensor seperti sensor posisi atau sensor beban. ESP32 dan ESP8266 dapat digunakan untuk memproses data dari sensor-sensor ini dan menghasilkan sinyal kontrol yang sesuai untuk mengoptimalkan gerakan platform robot.

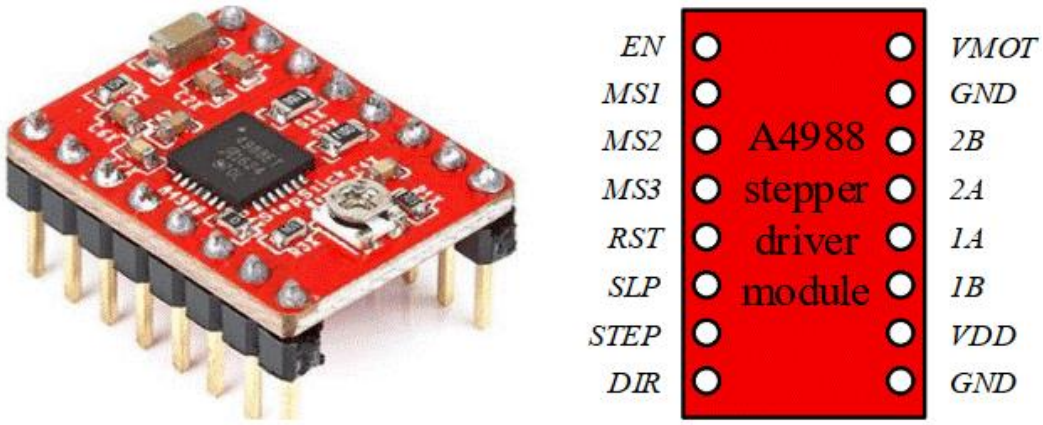
Dengan memanfaatkan kemampuan-kemampuan ini, ESP32 dan ESP8266 dapat secara efektif digunakan dalam sistem CDPR untuk mencapai kontrol gerakan yang presisi, komunikasi antar-modul yang efisien, dan pemrosesan sensor yang akurat. Berikut merupakan

konfigurasi pin I/O yang digunakan oleh esp8266 untuk mengendalikan motor *stepper* NEMA17 menggunakan *driver* A4988:

Tabel 2.2 Pin I/O ESP8266

MIKROKONTROLER	PIN	DRIVER PIN
SLAVE_1	D1	STEP MOTOR 1
SLAVE_2	D3	STEP MOTOR 2
SLAVE_3	D5	STEP MOTOR 3
SLAVE_4	D7	STEP MOTOR 4
SLAVE_1	D2	DIR MOTOR 1
SLAVE_2	D4	DIR MOTOR 2
SLAVE_3	D6	DIR MOTOR 3
SLAVE_4	D8	DIR MOTOR 4

3. Driver Motor Stepper



Gambar 2.3 *Driver* A4988 dan Pin I/O

Driver A4988 memiliki 16 pin dan memiliki fitur-fitur seperti pembatas arus yang bisa disesuaikan dengan kebutuhan, perlindungan terhadap arus dan suhu berlebih.

Tabel 2.3 Fungsi Pin *Driver* A4988

PIN	**FUNGSI PIN**
STEP	Mengatur langkah motor *stepper*
DIR	Mengatur arah putaran motor *stepper*
MS1	Pengatur *microstepp* 1
MS2	Pengatur *microstepp* 2
MS3	Pengatur *microstepp* 3
ENABLE	Mengaktifkan/menonaktifkan *driver*
RESET	Mengatur ulang *driver*
SLEEP	Memasukkan *driver* ke mode *sleep*
GND	*Ground*
VMOT	Sumber tegangan motor
VDD	Sumber tegangan logika

2B	Koneksi kumparan B motor *stepper* kedua
2A	Koneksi kumparan A motor *stepper* kedua
1A	Koneksi kumparan A motor *stepper* pertama
1B	Koneksi kumparan B motor *stepper* pertama

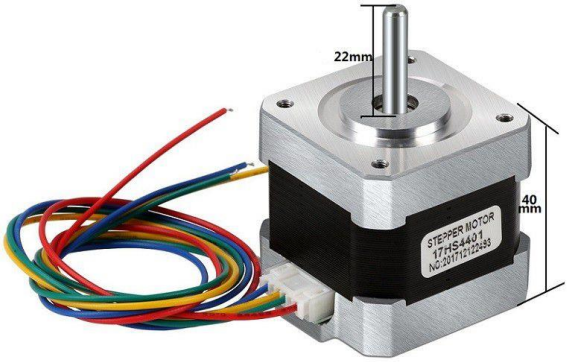
Dalam mengendalikan motor *stepper* menggunakan *driver* A4988, jumlah langkah yang diperlukan untuk satu putaran (atau 1 *revolution*) dapat diatur melalui pengaturan langkah pada *driver*. Berikut adalah beberapa jenis langkah yang umumnya digunakan dalam pengendalian motor *stepper*:

- *Full Step* (Langkah Penuh): Ini adalah langkah utama penuh dari motor *stepper*, yang dalam konteks NEMA17, biasanya setara dengan 1.8 derajat per langkah. Dengan demikian, motor akan bergerak 1.8 derajat untuk setiap langkah penuhnya atau sama dengan 200 step.
- *Half Step* (Langkah Setengah): Dalam langkah setengah, setiap langkah penuh dibagi menjadi dua langkah lebih kecil. Ini berarti ada dua kali lipat langkah per putaran motor. Dalam konteks NEMA17, langkah setengah akan menjadi sekitar 0.9 derajat per langkah atau sama dengan 400 step.
- *Quarter Step* (Langkah Perempat): Setiap langkah penuh dibagi menjadi empat langkah perempat. Ini menghasilkan empat kali lipat langkah per putaran motor. Dalam konteks NEMA17, langkah perempat akan menjadi sekitar 0.45 derajat per langkah atau sama dengan 800 step.
- *Eighth Step* (Langkah Delapan): Setiap langkah penuh dibagi menjadi delapan langkah. Ini menghasilkan delapan kali lipat langkah per putaran motor. Dalam konteks NEMA17, langkah delapan akan menjadi sekitar 0.225 derajat per langkah atau sama dengan 1600 step.
- *Microstep* (Langkah Mikro): Teknik ini membagi setiap langkah penuh menjadi puluhan, ratusan, atau ribuan langkah mikro yang sangat kecil. *Microstepping* memungkinkan gerakan yang sangat halus dan presisi tinggi, tetapi memerlukan *driver motor stepper* yang mendukung fitur *microstepping* .

Selain pengaturan langkah, batas tegangan motor *stepper* dapat diatur menggunakan potensio yang ada di *driver* A4988. *Driver* ini mampu memberikan *output drive* maksimum

hingga 35V dengan arus sekitar +-2A. dan untuk penelitian ini peneliti akan menggunakan *full step* membuat step yang digunakan sama dengan 200 step. Akan berguna di perhitungan geometri untuk mencari Panjang kabel yang dibutuhkan untuk mencapai suatu koordinat.

4. Motor Stepper



Gambar 2.4 *Motor Stepper* NEMA17

Motor *stepper* adalah perangkat elektromekanis yang bekerja dengan mengubah tenaga listrik menjadi tenaga mekanis. Motor *stepper* bergerak berdasarkan urutan pulsa yang diberikan kepada motor. Motor *stepper* memiliki kelebihan seperti pemosisian dan pengulangan gerakan yang tepat, respon yang sangat baik untuk memulai, menghentikan, dan membalikkan, serta sangat andal karena tidak ada sikat kontak di motor. Oleh karena itu, umur motor hanya bergantung pada umur *bearing*.

Motor *stepper* beroperasi secara berbeda dari motor sikat DC, yang berputar ketika tegangan diberikan ke terminalnya. Motor *stepper*, di sisi lain, secara efektif memiliki banyak elektromagnet bergigi yang diatur di sekitar sepotong besi berbentuk roda gigi pusat. Elektromagnet diberi energi oleh rangkaian kontrol eksternal, misalnya, mikrokontroler. Pada dasarnya, cara kerja motor *stepper* berbeda beda, tergantung pada konstruksi rotor dan stator serta sistem lilitan pada statornya. Namun setiap *stepper* digerakkan oleh pulsa digital yang diubah setiap waktu.

Tabel 2.4 Spesifikasi Motor Stepper NEMA17*[7]*

NEMA17 *stepper motor*	
Step Angle	1.8 *degree*
Rated voltage	12 VDC
Number of Phases	4
Current	400 mA
DC Resistance	30 Ω
Phase Inductance	37 mH
Holding Torque	2600 g/cm

Kecepatan sinyal pulsa digital atau lebih tepatnya frekuensi sinyal mempengaruhi kecepatannya. Semakin cepat frekuensi sinyal, maka semakin cepat pula RPM pada *stepper*. Sinyal digital ini dapat pula digunakan untuk menentukan posisi pada motor *stepper*. Misalnya saja, *stepper* dengan sudut step 1.8° maka untuk satu putaran penuh diperlukan 200 step. Motor ini akan digunakan untuk menggerakkan bandul yang berada di tengah 4 buah tiang. Untuk kemudian menarik atau mengulur benang supaya bandul bergerak ke titik koordinat yang diinginkan.

5. Power Supply

Gambar 2.5 Power Supply 12V 30 A

Power Supply Switching adalah sistem catu daya yang menggunakan teknologi *switching* untuk mengatur tegangan dan arus listrik secara efisien. Tegangan AC dari sumber listrik diubah menjadi tegangan DC, yang kemudian diolah melalui proses *switching* untuk menghasilkan tegangan yang diinginkan. Proses ini melibatkan beberapa tahapan:

1. Penyatuan arah dan *Filtering*: Tegangan AC dari sumber listrik diubah menjadi DC melalui dioda penyearah dan filter kapasitor untuk meratakan tegangan.
2. Switching: Tegangan DC yang sudah dihasilkan dipecah menjadi sinyal pulsa DC berfrekuensi tinggi menggunakan transistor atau IC *switching*.
3. Transformasi: Sinyal pulsa ini melewati trafo switching yang berukuran lebih kecil, mengubah tegangan sesuai kebutuhan perangkat.
4. Penyatuan arah Akhir dan Regulasi: Tegangan AC dari trafo diubah kembali menjadi DC yang stabil melalui proses Penyatuan arah dan *filtering* tambahan.

Keunggulan *Power Supply Switching* dibandingkan dengan catu daya linear termasuk efisiensi yang lebih tinggi, ukuran dan berat yang lebih kecil, serta kemampuan untuk beroperasi dengan berbagai tegangan masukan sambil tetap memberikan keluaran yang stabil.

Efisiensi tinggi dicapai karena *Power Supply Switching* bekerja pada frekuensi tinggi, yang mengurangi kerugian daya dan panas yang dihasilkan [8].

Power Supply Switching juga dikenal karena keandalan dan kemampuannya untuk digunakan dalam berbagai aplikasi, mulai dari perangkat elektronik konsumen hingga peralatan industri. Studi dan penelitian terbaru telah menunjukkan bahwa metode seperti *Extended Kalman Filter* dan jaringan neural LSTM dapat digunakan untuk memantau kondisi dan memprediksi masa pakai *Power Supply Switching*, meningkatkan keandalan dan efisiensi dalam penggunaannya [9]. Pada penelitian ini catu daya akan digunakan untuk memberi daya pada masing-masing *driver motor stepper* dan juga ESP8266 yang terhubung. Dikarenakan perbedaan daya yang dibutuhkan jadi disini peneliti akan menggunakan dua buah *power supply*. Satu untuk *driver motor* dan satu lagi untuk ESP8266.

6. Protokol Komunikasi

Protokol komunikasi yang akan di gunakan pada penelitian ini adalah ESP-NOW. ESP-NOW adalah protokol komunikasi nirkabel yang dikembangkan oleh *espressif* untuk *chip* ESP mereka, seperti ESP8266 dan ESP32. Protokol ini memungkinkan komunikasi *peer-to-peer* yang cepat dan hemat daya tanpa memerlukan jaringan *Wi-Fi* tradisional atau *router*. ESP-NOW menggunakan struktur pesan sederhana yang memungkinkan pengiriman data langsung antar perangkat, menjadikannya ideal untuk aplikasi *IoT* yang membutuhkan komunikasi cepat dan andal dengan konsumsi daya rendah[10].

ESP-NOW menggunakan teknologi frekuensi radio yang sama dengan *Wi-Fi* (2.4 GHz), tetapi dengan *overhead* yang lebih rendah, sehingga lebih efisien. Protokol ini memungkinkan pengiriman pesan *broadcast* atau *unicast*, dan dapat mendukung hingga dua puluh perangkat secara bersamaan, termasuk hingga sepuluh perangkat yang dienkripsi. Implementasi ESP-NOW pada perangkat seperti ESP8266 dan ESP32 melibatkan penggunaan API yang disediakan oleh *Espressif*, yang memungkinkan pengembang untuk mengintegrasikan protokol ini dengan mudah dalam aplikasi mereka [10].

Dalam penelitian[11], penggunaan ESP-NOW dalam sistem pemantauan cerdas menunjukkan peningkatan yang signifikan dalam hal efisiensi dan keandalan pengiriman data, menjadikannya pilihan yang menarik untuk berbagai aplikasi *IoT* dan industri lainnya. Berdasarkan penjelasan sebelumnya dalam aplikasi pada *Greenhouse*, ESP-NOW dapat digunakan untuk pemantauan dan pengendalian proses dengan mengimplementasikan jaringan *peer to peer* yang dapat secara otomatis mengonfigurasi ulang jika terjadi kegagalan pada salah

satu titik jaringan, meningkatkan keandalan sistem secara keseluruhan. Faktor faktor tersebut membuat protokol komunikasi ini sangat cocok digunakan untuk di implementasikan pada perancangan sistem modular terdistribusi yang dibuat pada penelitian ini.

7. Platform Pengembangan

Arduino IDE adalah platform perangkat lunak yang memfasilitasi penulisan, penyusunan, dan pengunggahan kode ke papan mikrokontroler Arduino. Dikembangkan sebagai proyek *open-source*, Arduino IDE menyediakan antarmuka yang lengkap, termasuk editor kode, manajer pustaka, manajer papan, monitor serial, serta alat penyusunan dan pengunggah. Fitur utamanya adalah kemampuan untuk menambah pustaka tambahan yang mendukung integrasi mudah dengan berbagai sensor, aktuator, dan komponen lainnya. Dukungan dari komunitas pengguna yang besar dan aktif, bersama dengan dokumentasi, tutorial, dan forum *online*, membuat Arduino IDE menjadi alat yang esensial dalam pengembangan proyek elektronik dan pendidikan STEM[12].